ATAC AL PROBLEMA DEL LOGARITME DISCRET MITJANÇANT L'ALGORISME DE POHLIG-HELLMAN

Autor: David Fàbrega Sabaté
Director: Josep M. Miret Biosca

Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica d'Informàtica de Sistemes
Treball de Final de Carrera 2007

Índex

Introducció

La Criptografia (del Grec *kryptós*, "ocult", i *gráphein*, "escriure") és, tradicionalment, l'estudi de formes de convertir informació des de la seva forma original cap a un codi incomprensible, de forma que sigui il·legible pels que no coneguin aquesta tècnica. En el passat la criptografia va ajudar a assegurar el secret en les comunicacions importants, agents secrets o documents militars o diplomàtics.

En l'actualitat, l'ús de la criptografia s'ha estès tant en els sectors empresarials i polítics, per exemple els bancs per fer comerç electrònic, o als usuaris particulars que gràcies a iniciatives com el PGP (Pretty Good Privacy o privacitat bastant bona) poden disposar de nivells de seguretat usats pels serveis secrets per enviar els seus correus electrònics.

De la informació original en diem el text pla (encara que no necessàriament treballem amb textos). Llavors passa per un procés de xifrat que fent servir algorismes converteix la informació original en un codi il·legible per tothom que no tingui els mitjans per desxifrar (un altre algorisme), i la clau.

Actualment els algorismes, o tècniques criptogràfiques, consisteixen en programes d'ordinador que aprofiten propietats numèriques que fan que sense la clau sigui molt difícil d'obtenir la informació. Per exemple estem fent servir la criptografia quan ens connectem al nostre banc a través de Internet, de manera que encara que algú intercepti la informació que intercanviem amb aquest, no podrà descodificar la informació interceptada.

Un algorisme criptogràfic es diu de clau simètrica quan es fa servir la mateixa clau per a xifrar i per a desxifrar. Cal doncs que aquesta clau es faci arribar al destinatari del missatge per algun mitjà alternatiu. Els algorismes de clau pública generen un parell de claus, una d'elles es fa servir per a xifrar el missatge i l'altra pot desxifrar-lo. El receptor del missatge pot fer pública una de les claus i mantenir secreta l'altra, d'aquesta manera no és necessari disposar d'un canal segur per on enviar claus. Els algorismes de clau pública es fan servir per a construir signatures digitals.

En Criptografia de clau pública la seguretat està basada en la intractabilitat computacional de certs problemes matemàtics d'inversió. Per exemple, és senzill multiplicar dos nombres p,q de 200 xifres decimals cadascun i fabricar un nombre $N{=}p{\cdot}q$ de 400 xifres decimals, però és molt difícil factoritzar el nombre N per recuperar els factors originals. Qualsevol ordinador pot efectuar el primer procés en milèsimes de segon, però tardarà més de mil anys en aplicar un algorisme que factoritzi N.

L'avenç de la ciència i la tecnologia comporta un creixement del tamany de les claus criptogràfiques. Per exemple, el dia que un ordinador i/o un algorisme factoritzin el nombre N d'abans en poques hores, caldrà passar a treballar amb claus N de 800 xifres decimals en comptes de 400. Aquest fenomen dificulta l'aplicació de protocols criptogràfics en dispositius de poca memòria com és el cas de les targetes intel·ligents.

Un altre problema en que es basen bona part dels criptosistemes de clau pública és el problema del logaritme discret (DLP). En l'actualitat no existeixen algorismes eficients que siguin capaços de calcular en un temps raonable aquest tipus de logaritmes, i molts esquemes criptogràfics basen la seva resistència en aquesta circumstància.

En aquest treball de final de carrera, s'ha implementat l'algorisme de Pohlig Hellman, el cual permet atacar al problema del logaritme discret en determinats casos que més tard explicarem. La primera part d'aquesta memòria es tracta de les bases i conceptes matemàtics necessaris per a entendre els següents capítols. També s'explicará quin és el problema del logaritme discret, i també quins son alguns dels seus atacs. Després es donarà una descripció detallada de l'algorisme de Pohlig Hellman, i també alguns esquemes sobre la seva implementació i els resultats obtinguts, tant sobre el grup multiplicatiu $\mathbb{F}_p^*$ com sobre el grup de punts d'una corba el·líptica. Finalment s'analitzaran ambdós algorismes, i s'escriuran les conclusions on s'hauran arribat.

1 Preliminars matemàtics

En aquest capítol explicarem els fonaments matemàtics que creiem que són bàsics per introduir les corbes el·líptiques i els posteriors capítols del treball.

Les corbes el·líptiques són especialment importants en la teoria de nombres, i constitueixen una àrea de recerca actual molt important; per exemple, foren usades per Andrew Wiles en la demostració del darrer teorema de Fermat [1]. També tenen múltiples aplicacions en la criptografia i en la factorització d'enters. Són estudiades des de fa més de 150 anys, i presenten una sèrie de propietats que donen lloc a problemes difícils, cosa que les fa vàlides per aplicarles a alguns dels algorismes asimètrics més coneguts. La seva estructura algebraica i geomètrica és complexa, i la implementació de la seva operació de grup és eficient. Les primeres propostes d'ús de corbes el·líptiques en criptografia van ser fetes per Neal Koblitz i Victor Miller al 1985 [2]. Precisament el principal argument que fan servir els detractors d'aquestes tècniques són que, si les corbes el·líptiques han estat objecte d'estudi i anàlisi durant més d'un segle, les propietats que poden estar directament relacionades amb la seva qualitat com a base d'un sistema criptogràfic, solament porten 17-22 anys éssent considerades. Tot i això en corbes el·líptiques les claus que utilitzen són molt més curtes i tenen el mateix nivell de seguretat que amb altres tècniques.

1.1 Teoria de Grups

En aquest apartat explicarem els fonaments matemàtics per tal de comprendre la resta de conceptes que s'exposaran en aquest capítol.

Definició 1.1.1

Donat un conjunt G i una llei de composició interna $*$ sobre G, direm que $(G, *)$té una estructura de grup si satisfà:

1. L'operació $*$ és associativa, és a dir, si $a*(b*c) = (a*b)*c$, $\forall a,\ b,\ c \in G$.

2. Existeix un únic element *neutre* $e \in G$ tal que $e*a = a*e = a$, $\forall a \in G$.

3. Tot element és invertible, és a dir, per a cada $a \in G$ existeix un element $b \in G$ tal que $b*a = a*b = e$. L'element b és únic i s'anomena *invers*

de a. El grup $(G, *)$ s'anomena grup abelià o commutatiu si l'operació $*$ és commutativa, és a dir, si $a * b = b * a$ per a qualsevol a, $b \in G$.

Definició 1.1.2

Es diu que un grup G és finit si té un nombre finit d'elements. L'ordre o cardinal d'un grup finit G, denotat per $\#G$ o $|G|$ és el nombre d'elements que té el grup. L'ordre d'un element a de G, denotat per $\#a$, és el menor $n \in N$ tal que $a^n = a * \ldots * a = e$, on e és l'element neutre de $(G, *)$.

Definició 1.1.3

Un subconjunt H, no buit, d'un grup $(G, *)$, és un subgrup de $(G, *)$ si el subconjunt H amb la llei de composició interna $*$ és també un grup. Per a que el subconjunt H del grup $(G, *)$ sigui subgrup és condició necessària i suficient que es verifiqui la condició següent:

$$\forall a,\ b \in H \Rightarrow a * b^{-1} \in H.$$

Definició 1.1.4

Si $(G, *)$ és un grup finit i a un element de G, denotarem per $< a >$ el conjunt format per totes les potències diferents de a, és a dir:

$$< a >= a, a * a, a * a * a, ..., a * ... * a = e,$$

aquest conjunt és un subgrup de $(G, *)$, anomenat subgrup cíclic generat per a.

Teorema 1.1.1 ***(Lagrange)***

Donat un grup finit G i un subgrup H de G, llavors $\#H | \#G$. En particular, donat un element a de G, es verica que $\#a | \#G$, és a dir, l'ordre d'un element del grup sempre divideix l'ordre del grup.

Definició 1.1.5

Donats dos grups $(G, *)$ i $(G', \cdot)$, una aplicació:

$$f : G \longrightarrow G'$$

$$x \longmapsto f(x),$$

es diu que és un morfisme de grups si satisfà:

$$f(x * y) = f(x) \cdot f(y), \forall x, y \in G.$$

Donats dos grups $(G, *)$ i $(G', \cdot)$, direm que són isomorfs, i ho denotarem per $(G, *) \simeq (G, \cdot)$, si existeix un morfisme de grups$f : G \longrightarrow G'$ bijectiu.

1.2 Anells

Definició 1.2.1

Donat un conjunt A amb dues operacions internes $+$ i $\cdot$, direm que $(A, +, \cdot)$ és un anell si se satisfà:

1. $(A, +)$ és un grup abelià.
2. L'operació $\cdot$ és associativa.
3. L'operació $\cdot$ és distributiva respecte la $+$.

Un anell $(A, +, \cdot)$ que té element neutre respecte a l'operació interna $\cdot$ es diu que és unitari. Si l'operació interna $\cdot$ satisfà la propietat commutativa es diu que l'anell $(A, +, \cdot)$ és commutatiu.

Es diu que $a \in A$ - $\{0\}$ és un divisor de zero d'un anell $(A, +, \cdot)$, on 0 és l'element neutre de $(A, +)$, si existeix un element $b \in A - \{0\}$ tal que $a \cdot b = 0$ o bé $b \cdot a = 0$. Un anell commutatiu unitari sense divisors de zero s'anomena *domini d'integritat*.

L'anell $\mathbb{Z}/n\mathbb{Z}$ de classes de restes mòdul n

Anomenarem $\mathbb{Z}/n\mathbb{Z}$ o $\mathbb{Z}_n$ al conjunt quocient de $\mathbb{Z}$ per la relació de la congruència mòdul n, que estarà format per totes les classes de congruència diferents mòdul n, és a dir:

$$\mathbb{Z}/n\mathbb{Z} = \{\overline{0}, \overline{1}, ..., \overline{n-1}\}, \text{ on } \overline{a} = \{a + k \cdot n \mid k \in \mathbb{Z}\}.$$

Si n és un enter positiu, les operacions internes $+$ i $\cdot$ a $\mathbb{Z}/n\mathbb{Z}$, definides a partir de $+$ i $\cdot$ del conjunt $\mathbb{Z}$ doten a $(\mathbb{Z}/n\mathbb{Z}, +, \cdot)$ d'estructura d'anell unitari commutatiu. Denotem per $(\mathbb{Z}/n\mathbb{Z})*$ al conjunt d'elements invertibles de $\mathbb{Z}/n\mathbb{Z}$. Aquest conjunt amb l'operació $\cdot$ té estructura de grup abelià.

Proposició 1.2.1

Siguin n un enter positiu i a un enter.

Si $m.c.d.(a,\ n) = 1$, llavors $\overline{a}$ és invertible a $\mathbb{Z}/n\mathbb{Z}$, és a dir existeix un $x \in \mathbb{Z}$ tal que $\overline{a} \cdot \overline{x} = \overline{1}$.

Si $m.c.d.(a,\ n) = n$, llavors $\overline{a} = \overline{0}$ en $\mathbb{Z}/n\mathbb{Z}$.

Si $m.c.d.(a,\ n) = d$, éssent $1 < d < n$, llavors $\overline{a}$ és un divisor de zero a $\mathbb{Z}/n\mathbb{Z}$.

1.3 Cossos

Definició 1.3.1

Donat un conjunt $\mathbb{K}$ amb dues operacions internes $+$ i $\cdot$, direm que $(\mathbb{K},\ +,\ \cdot)$ té estructura de cos si:

1. $(\mathbb{K},\ +,\ \cdot)$ és un anell unitari.
2. $\forall a \in \mathbb{K} - \{0\}$és invertible per a la segona operació, on 0 és el neutre de l'operació $+$.

A més a més si, l'operació interna $\cdot$ satisfà la propietat commutativa, es diu que $(\mathbb{K}\ ,\ +,\ \cdot)$ és un cos commutatiu. El conjunt $\mathbb{K}^*$ té estructura de grup respecte l'operació interna $\cdot$.

Teorema 1.3.2

Si p és primer, l'anell $(\mathbb{Z}_p,\ +,\ \cdot)$ té estructura de cos (aquest cos es denota amb $\mathbb{F}_p^*$).

Definició 1.3.3

Sigui p un primer senar i $a \in \mathbb{Z}$, $1 \leq a \leq + - 1$. Llavors a s'anomena residu quadràtic mòdul p si $x^2 \equiv a \pmod{p}$ té una solució $x \in \mathbb{F}_p$. Sinó x és no residu quadràtic mòdul p.

Definició 1.3.4 ***Símbol de Legendre sobre*** $\mathbb{F}_p$

Donat un element $aa \in \mathbb{F}_p$, es defineix el símbol de Legendre $\left(\frac{a}{p}\right)$ de a sobre p de la forma:

- 1, si $\exists x \in \mathbb{F}_p$ tal que $x^2 = a$.

- 0, si $a = 0$.
- -1, si $\nexists x \in \mathbb{F}_p$ tal que $x^2 = a$.

Si $n = p_1^{e1}...p_k^{ek}$, on p_i són primers, es defineix el símbol de Jacobi Kronecker de a sobre n:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_i}\right)^{e_1} ... \left(\frac{a}{p_k}\right)^{e_k},$$

on els $\left(\frac{a}{p_i}\right)$són els símbols de Legendre de a sobre els primers p_i.

1.4 Teorema Xinès del Residu

La raó del nom d'aquest teorema es troba al llibre *Sun Tzu Suan Ching* [3], on el problema 26 del tercer volum diu:

"*Tenim un cert nombre de coses, però no sabem exactament quantes. Si les contem de 3 en 3 ens en sobren 2. Si les contem de 5 en 5, ens en sobren tres. Si les contem de 7 en 7, ens en sobren 2. Quantes coses hi ha?*"

El que diu aquest teorema es que si tenim una sèrie d'equacions:

$x = a_1 \ (mod \ m_1)$

$x = a_2 \ (mod \ m_2)$

...

$x = a_k \ (mod \ m_k)$

on els m_i (i=1...k) són primers entre ells dos a dos, llavors el sistema té una única solució módul $M = m_1 \cdot m_2 ... m_k$.

La forma de trobar la solució ens dóna la demostració del teorema. Es calculen els nombres $M_i = (m_1 \cdot m_2 ... m_k)/m_i$(o equivalentment el producte de tots els móduls excepte l'i-èsim i els enters Y_i que són els inversos de M_i módul m_i. A continuació es demostra que el número:

$x = a_1 \cdot M_1 \cdot Y_1 + a_2 \cdot M_2 \cdot Y_2 + ... + a_k \cdot M_k \cdot Y_k$

és la solució buscada.

El Teorema Xinès del Residu s'ha empleat en algunes implementacions de RSA per a simplificar els càlculs (si $n = p \cdot q$ en comptes de treballar en $\mathbb{Z}_n$ es treballa en $\mathbb{Z}_p$ x $\mathbb{Z}_q$) així com per a l'aritmètica amb enters llargs amb l'ordenador.

1.5 Corbes el·líptiques

En aquesta secció dins dels preliminars matemàtics, explicarem els conceptes bàsics sobre corbes el·líptiques necessaris pel nostre treball.

1.5.1 Introducció a les corbes el·líptiques

Una corba el·líptica sobre un cos $\mathbb{K}$ és una corba plana definida per una equació de la forma:

$$E_{A,B}/\ \mathbb{K}\colon\ y^2 = x^3 + Ax + B,$$

que no té punts singulars. Aquesta equació és anomenada equació reduïda de Weierstrass. Per a evitar que la corba tingui punts singulars, el discriminant de la corba no pot ser 0, és a dir, la seva gràfica no té cúspides o punts d'intersecció amb ella mateixa.

El conjunt de punts d'una corba el·líptica $E_{A,B}/\ \mathbb{K}$ ve definit com:

$$E_{A,B}/\ \mathbb{K} = (x, y) \in \mathbb{K} \times \mathbb{K} \mid y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\},$$

on $\mathcal{O}$ és l'anomenat punt de l'infinit de la corba, que permetrà dotar al conjunt d'estructura de grup abelià.

1.5.2 Suma de punts en una corba el·líptica

Al conjunt de punts d'una corba el·líptica$E_{A,B}$ ($\mathbb{K}$) hi podem definir una operació suma. Aquesta operació es pot expressar gràficament pel mètode de la corda i la tangent. A continuació mostrarem aquest mètode:

Mètode de la corda i la tangent

Consisteix en traçar la recta que uneixi els dos punts P i Q que volem sumar, si P i Q són diferents. Aquesta recta talla en un tercer punt de la corba que anomenarem R. Per aquest tercer punt trobem el seu simètric respecte l'eix de les abscisses, traçant una recta vertical sobre R i agafant el punt per on torna a tallar la corba. Aquest punt és el punt suma, $S = P + Q$. Tal i com podem veure a la figura 1.1.

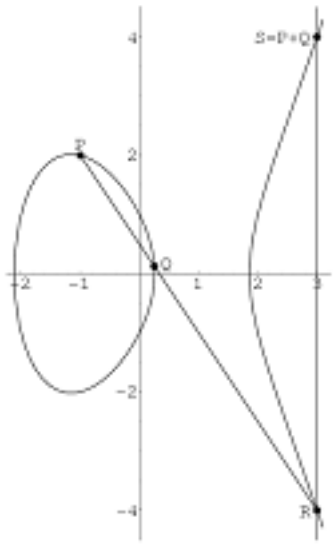

Figura1.1: Suma de dos punts pel mètode de la corda i la tangent.

En el cas que $P = Q$, es traça una recta tangent respecte el punt $P = Q$ que, al igual que en el cas anterior, tallarà a la corba en un tercer punt, que anomenem R. Després trobem el simètric de R respecte l'eix de les abscisses, i el punt per on talla aquesta recta és el punt suma, $S = 2P$, tal i com podem veure a la figura 1.2.

També podem expressar analíticament el mètode de la corda i la tangent. Sigui $E_{A,B}$ una corba el·líptica d'equació $y^2 = x^3 + Ax + B$ definida sobre un cos $\mathbb{K}$. Siguin $P = (x_1 , y_1)$ i $Q = (x_2 , y_2) \in E_{A,B}$ $(\mathbb{K})$ els dos punts a sumar i sigui $S = (x_3 , y_3) \in E_{A,B}/ \mathbb{K}$ el resultat de la seva suma. Llavors distingirem entre els casos següents:

- Si $x_1 \neq x_2$ i $y_1 \neq y_2$, és a dir, si $P \neq Q$ i $P \neq -Q$, les dades de S seran:

$$x_3 = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2,$$

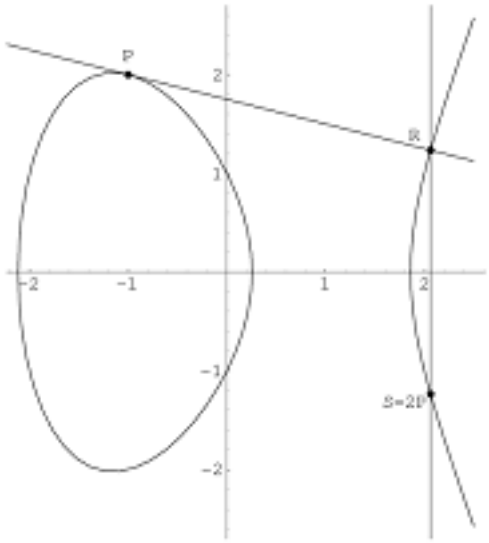

Figura 2.2: Doblat d'un punt pel mètode de la corda i la tangent.

$$y_3 = -y_1 + \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 (x_1 - x_2),$$

- Si $x_1 = x_2$ i $y_1 = y_2$, és a dir, si $P = Q$, les dades de S seran:

$$x_3 = \left(\frac{3x_1^2 + A}{2y_1}\right)^2 - x_1 - x_2,$$

$$y_3 = -y_1 + \frac{3x_1^2 + A}{2y_1}(x_1 - x_3),$$

- Si $x_1 = x_2$ i $y_1 = -y_2$, és a dir, si Q = -P, les dades de S seran:

$$S = P + Q = P - P = \mathcal{O}.$$

1.5.3 Múltiples d'un punt

Utilitzant la suma el·líptica podem definir el múltiple d'un punt d'una corba el·líptica per un número enter. Si $P \in E_{A,B}(\mathbb{K})$, i n és un enter, el punt nP es defineix com:

- si $n > 0$: $(P + P + ... + P)n$ vegades

- si $n < 0$: $((-P) + (-P) + ... + (-P))n$ vegades
- si $n = 0$: $\mathcal{O}$

Aquest procediment per calcular nP quan la n és molt gran és poc eficient. Una manera més eficient seria utilitzant l'algorisme del camperol rus, ja que el cost d'aquest càlcul és de $O(log_2\ n)$, un cost bastant millor que el que tenim si utilitzem sumes successives, $O(n)$.

Algorisme del camperol rus

L'algorisme del camperol rus és un algorisme d'exponenciació ràpida, que ens permet calcular nP amb un cost reduït. Aquest consisteix en sumar i multiplicar P amb o per potències de dos, però, segurament, la millor forma d'explicar-lo, serà a partir d'un exemple, tal com es mostra a continuació:

Sigui $n = 11$, llavors podem dir

$$nP = 11P = (2^3 + 2^1 + 2^0)P = (2(2^2 + 1) + 1)P$$

$$= (2(2^2P + P)) + P = (2(2(2P) + P)) + P.$$

Per tant, en el pitjor dels casos, el número màxim d'operacions que farem és:

$$2([log_2 n - 1]) + 1 = 2[log_2 n] + 1,$$

que té un cost de $O(log_2 n)$.

1.5.4 Corbes el·líptiques sobre cossos finits $\mathbb{F}_p$

Una corba el·líptica sobre un cos finit $\mathbb{F}_p$, on p és un primer, ve definida per una equació de la forma:

$E_{A,B}/\ \mathbb{F}_p : y^2 \equiv x^3 + Ax + B \pmod p$,

on A i B són elements de $\mathbb{F}_p$ i $4A^3 + 27B^2/ \equiv 0 \pmod p$. El conjunt dels punts de la corba, que denotats per $E_{A,B}(\mathbb{F}_p\ )$ són els punts $(x,\ y)$ tals que x i $y \in \mathbb{F}_p$ i que satisfan l'equació de la corba, junt amb el punt de l'infinit totes les propietats vistes anteriorment sobre cossos també son aplicables a aquestes corbes el·líptiques, ja que estan definides sobre el mateix cos $\mathbb{F}_p$, però aquestes tenen una sèrie de propietats i característiques pròpies.

Cardinal

El cardinal d'una corba el·líptica $E_{A,B}/\ \mathbb{F}_p$, que denotem per $\#E_{A,B}/\ \mathbb{F}_p$, és el número de punts que conté la corba amb coordenades a $\mathbb{F}_p$, més el punt de l'infinit $\mathcal{O}$. El següent resultat, anomenat teorema de *Hasse*, acota el cardinal d'una corba el·líptica.

Teorema 1.5.4.1 *(Hasse)*

Sigui $E_{A,B}$ una corba el·líptica sobre un cos finit $\mathbb{F}_p$ i sigui $m = \#E_{A,B}(\mathbb{F}_p)$, aleshores se satisfà:

$$p + 1 - 2\sqrt{p} \leq m \leq p + 1 + 2\sqrt{p}.$$

Teorema 1.5.4.2 *(Waterhouse)*

Si $m \in [p + 1 - 2\sqrt{p},\ p + 1 + 2\sqrt{p}]$, llavors existeix una corba el·líptica$E_{A,B}/\ \mathbb{F}_p$ tal que

$$\#E_{A,B}/\ \mathbb{F}_p = m.$$

Cal remarcar que el càlcul del cardinal d'una corba el·líptica és un problema computacionalment difícil. Tot i que existeixen algorismes polinòmics que el resolen, aquests a la pràctica resulten ineficients.

2 El Problema del Logaritme Discret

En aquest capítol comentarem el protocol de claus Diffie-Hellman, però sobretot parlarem del *problema del logaritme discret*, tant en el cos multiplicatiu $\mathbb{F}_p^*$, com sobre el grup de punts d'una corba el·líptica.

2.1 Protocol de claus Diffie-Hellman

Al 1976 W. Diffie i M.E. Hellman [4] van establir les bases teòriques dels algorismes de clau pública. Fins llavors no s'havia pogut concebre un sistema de xifrat que no fos de clau secreta. Suggereixen utilitzar problemes computacionalment irresolubles pel disseny de criptosistemes segurs. La idea és trobar un sistema de xifrat computacionalment senzill, de manera que el desxifrat sigui computacionalment irresoluble a menys que es conegui una informació addicional anomenada clau privada. En els xifrats asimètrics, o de clau pública, la clau de desxifrat no es pot calcular a partir de la de xifrat. Qualsevol usuari pot xifrar usant la clau pública, però només aquells que coneguin la clau privada poden desxifrar correctament. Aquest protocol va proporcionar la primera solució pràctica al problema de distribució de claus, permetent que dos parts estableixin un secret compartit intercanviant els missatges sobre un canal insegur.

Com acabem de dir el protocol de claus Diffie-Hellman dóna un mètode per a que dos usuaris A i B puguin compartir una informació confidencial (clau privada) sense necessitat de trobar-se. Per a això, A escolleix un número primer p i un generador $\alpha \in \mathbb{F}_p^*$, genera un número aleatori secret x, $1 \leq x \leq p-2$, calcula α^x i envia a B el missatge $(p,\ \alpha,\ \alpha^x)$. Un cop B ha rebut el missatge, també triarà un número aleatori y, $1 \leq y \leq p-2$, calcularà x^y, i enviarà el missatge x^y cap a A. Més tard B calcula el secret compartit fent el següent: $K = (\alpha^x)^y$. Per altra banda, A rebrà el missatge de B i trobarà el mateix valor mitjançant: $K = (\alpha^y)^x$.

Encara que el problema real que troba el atacant és obtenir α^{xy}, coneixent p, el generador α de $\mathbb{F}_p^*$, i els elements α^x i α^y, es creu que la resolució d'aquest problema té la mateixa complexitat que la del *logaritme discret*.

2.2 Logaritme Discret

Sigui $(G, *)$ un grup cíclic finit d'ordre n, sigui α un generador de $(G, *)$, i sigui $\beta \in G$. S'anomena *logaritme discret* de β en base α, i es denota $log_\alpha \beta$,

a l'únic enter x, $0 \leq x \leq n-1$, tal que:

$$x^y = \beta.$$

Aleshores el *problema del logaritme discret generalitzat* GDLP és el següent: donat un grup cíclic finit $(G, *)$ d'ordre n, un generador $\alpha \in (G, *)$ i un element $\beta \in G$, trobar l'enter x, $0 \leq x \leq n-1$, tal que $x^y = \beta$.

En el GDLP, l'únic requeriment que es fa al grup $(G, *)$ perquè tingui un ús útil en criptografia és que sigui molt difícil calcular el logaritme discret, mentres que l'operació $*$ del grup sigui ràpida d'executar. Cal dir que es pot generalitzar la definició del GDLP per usar-lo amb qualsevol grup G no necessàriament cíclic. Els grups més habituals són el grup multiplicatiu d'un cos $\mathbb{F}_p^*$ o més en general d'un cos $\mathbb{F}_{p^m}$, i el grup de punts d'una corba el·líptica.

La dificultat de resoldre el GDLP és independent del generador, ja que donats α i γ generadors de grup cíclic G d'ordre n, i donat un $\beta \in G$, tindrem $x = log_\alpha \beta$, $y = log_\gamma \beta$ i $z = log_\alpha \gamma$. Llavors $\alpha^x = \beta = \gamma^y = (\alpha^z)^y$. Com a conseqüència $x = zy \pmod{n}$, i

$$log_\gamma \beta = (log_\alpha \beta)(log_\alpha \gamma)^{-1} \pmod{n}.$$

Això signica que cada algorisme que calcula logaritmes de base α, podrà ser utilitzat per calcular logaritmes en una altre base γ que també és generadora de G.

2.3 Logaritme discret sobre el grup $\mathbb{F}_p^*$

El problema del logaritme discret sobre el grup multiplicatiu $\mathbb{F}_p^*$, (DLP) és un cas particular del GLDP: $G = \mathbb{F}_p^*$, on p és un primer, i l'operació de $\mathbb{F}_p^*$ és el producte i el cardinal és p-1, de manera que tenint un generador $\alpha \in \mathbb{F}_p^*$ i un element $\beta \in \mathbb{F}_p^*$, el problema serà trobar un possible enter x, $0 \leq x \leq p-2$, tal que :

$$\alpha^x = \beta,$$

és a dir, calcular

$$x = log_\alpha \beta$$

El problema del logaritme discret en el grup multiplicatiu $\mathbb{F}_p^*$ amb p primer és un problema d'interès amb criptografia. En un cos finit és fàcil elevar un número a una potència degut a l'algorisme de la potenciació modular, dels quadrats repetits o el camperol rus. En canvi l'operació inversa és molt costosa. Per això aquest problema és utilitzat en sistemes criptogràfics. Si ens pregunten a quina potència s'ha d'elevar 7 per obtenir 4 a $\mathbb{F}_2 3^*$, no és tan fàcil donar la resposta. Hauriem de provar d'una forma exhaustiva 7^1, 7^2, ... fins obtenir com a resultat el 4. Com més gran sigui el tamany del cos finit, més difícil serà el problema.

L'operació del grup $\mathbb{F}_p^*$ és la multiplicació mòdul p. A més a més, el GDLP es pot escriure utilitzant com a grup el grup multiplicatiu d'un cos finit $\mathbb{F}_{p^m}$, amb p primer i $m \geq 1$.

Hi ha diversos criptosistemes que la seva seguretat es basa en el DLP, entre ells estan:

1. Els esquemes de claus derivats del Diffie-Hellman, tals com el de ElGamal, La família de protocols MTI (Matsumoto, Takashima and Mai [5]) i els protocols STS (Station-to-Station [6]).

2. L'esquema de firma digital ElGamal i les seves variants, com el DSA (Digital Signature Algorithm [7]), l'esquema de firma de Schnorr [7] i l'esquema ElGamal amb recuperació de missatge de Nyberg-Rueppel [7].

2.4 Logaritme discret sobre corbes el·líptiques

El *problema del logaritme discret* sobre el grup de punts d'una corba el·líptica (ECDLP) es pot plantejar de la manera següent: Donada una corba el·líptica E definida sobre un cos finit F_p, un punt $P \in E(F_p)$ d'ordre n i un punt $Q \in < P >$ trobar l'enter l, $0 \leq l \leq n-1$, tal que $lP = Q$, és a dir, $l = log_p Q$.

La similitud de la Definició amb la del DLP fa que tots els criptosistemes basats en aquest puguin ser adaptats a corbes el·líptiques. Així tindrem variants dels protocols i esquemes anteriors convertits a corbes el·líptiques, i llavors tindrem ECDSA, EC Diffie-Hellman, EC ElGamal, ... Algunes modicacions s'hauran de fer per adaptarlos al grup de corbes, però els principis són els mateixos que per els altres sistemes basats en el DLP.

Donat que el ECDLP és un cas particular del GDLP, la pregunta que tothom és fa es que si el problema del logaritme discret és definit idènticament sobre $\mathbb{F}_p^*$, $\mathbb{F}_{p^m}^*$, $E(\mathbb{F}_p$) o $E(\mathbb{F}_{p^m}$), quin serà el guany de seguretat que s'obté al utilitzar corbes el·líptiques, en comptes d'altres esquemes i protocols sobre grups $\mathbb{F}_p^*$ i $\mathbb{F}_{p^m}^*$. La diferència està en els tipus d'atacs que es poden

implementar per a resoldre el problema del logaritme discret sobre aquests diferents grups.

2.5 Atacs coneguts al GDLP

En aquesta secció parlarem sobre els possibles atacs al problema del logaritme discret. Per començar introduirem l'atac més bàsic, la cerca exhaustiva, el qual es pot utilitzar sobre un grup qualsevol. Més tard parlarem d'atacs especícs sobre DLP.

2.5.1 Cerca exhaustiva

Es tracta de l'algorisme més senzill de tots: És un simple algorisme que opera a base de força bruta. Tan sols va calculant tots els possibles valors de $\alpha^t : \{(\alpha^1, \alpha^2, \alpha^3, ..., \alpha^n)\}$, des de $t = 1$ fins a n, on n és l'ordre de $G =< \alpha >$. El que vol dir un temps esperat d'ordre $O(n)$ operacions a G.

L'objectiu és obtindre una t tal que α^t doni β com a resultat. Podríem calcular cada element α^t , però seria una pèrdua de temps molt gran. Per tant solament és necessari agafar el valor anterior obtingut α^{t-1}, i a partir d'aquest calcular el següent.

$$\alpha^t = \alpha^{t-1} \cdot \alpha.$$

Gràcies a aquesta forma de càlcul, el procés serà una mica més ràpid, tot i això, per calcular números grans continuarà éssent molt lent, per tant haurem de buscar uns altres algorismes més efectius.

2.5.2 Algorismes per resoldre el GDLP

Essencialment hi ha tres categories d'algorismes per calcular *logaritmes discrets*:

Algorismes que treballen amb grups arbitraris, és a dir, els que no exploten cap característica específica del grup. Com per exemple el mètode Baby-Step Giant-Step [8], Rho de Pollard [8] i Pollard Lambda [9].

Algorismes que treballen bé dins dels grups, els quals l'ordre del grup no té cap factor primer gran. Més específicament, algorismes que treballen amb grups d'ordre smooth (un número n és diu que és B-*smooth* si tots els seus factors primers són$\leq B$). El principal algorisme d'aquest apartat és

el de Pohlig-Hellman [8], sobre el cual es centra aquest treball i el què més endavant s'analitzarà a fons la seva implementació..

Algorismes que exploten mètodes per representar els elements del grup com productes d'elements d'un sistema relativament petit. Els típics algorismes en aquesta categoria són l'Índex Calculus [9] i el Number Field Sieve [7]. Aquest algorismes són els més eficients per calcular el DLP, però no és poden usar en corbes el·líptiques.

2.5.3 Atacs coneguts al DLP

Com acabem de dir l'algorisme més eficient que no explota cap característica dels elements del grup és l'Índex Calculus. Aquest mètode comença amb una base de dades de primers petits i els seus logaritmes corresponents són calculats, això serveix per calcular eficientment logaritmes d'elements arbitraris del grup. L'algorisme Índex Calculus, així com el Number Field Sieve (adaptat per logaritmes) tenen un temps d'execució sub-exponencial i són fàcilment paral·lelitzables.

3 Atac de Pohlig Hellman

Sigui $(G, *)$ un grup cíclic finit d'ordre n, sigui α un generador de $(G, *)$, i sigui $\beta \in G$. L'algorisme de Pohlig-Hellman redueix d'una manera eficient el càlcul de $x = log_\alpha \beta$ al càlcul de logaritmes discrets per als subgrups d'ordre primer de $< \alpha >$. Es basa en que GDLP en $< \alpha >$ no és més costós que GDLP amb els seus subgrups d'ordre primer. Per tal de maximitzar la resistència a l'atac de Pohlig Hellman, el grup ha de ser sel·leccionat de manera que l'ordre n de α sigui divisible per un primer gran. Ara passarem a explicar l'algorisme de Pohlig Hellman sobre el grup multiplicatiu $\mathbb{F}_p^*$, i més tard ho farem sobre el grup de punts d'una corba el·líptica.

3.1 Algorisme de Pohlig-Hellman per al grup multiplicatiu $\mathbb{F}_p^*$

Sigui p un primer i sigui X un generador del grup multiplicatiu $\mathbb{F}_p^*$. Suposem que la factorització de $n = p-1$ sigui $p_1^{e_1} p_2^{e_2} ... p_r^{e_r}$. Sigui $Y \in \mathbb{F}_p^*$. L'estratègia de Pohlig Hellman per a calcular el logaritme discret $\ell = log_x Y$ és calcular $\ell_i = \ell \bmod p_i^{e_i}$ per a cada $1 \leq i \leq r$, i després solucionar el sistema de congruències:

$$\ell = \ell_1 \pmod{p_1^{e_1}}$$

$$\ell = \ell_2 \pmod{p_2^{e_2}}$$

$$\vdots$$

$$\ell = \ell_r \pmod{p_r^{e_r}}$$

per $\ell \in [0, n-1]$ el Teorema Xinès del Residu garanteix una única sol·lució. El càlcul de cada ℓ_i pot ser reduïda al càlcul de e_i logaritmes discrets, en el subgrup d'ordre p_i de $< X >$. Per a simplificar l'expressió, escrivim p per denominar p_i i e per a e_i. Llavors calcularem la representació en base p de ℓ_i:

$$\ell_i = z_0 + z_1 p + z_2 p^2 + \cdots + z_{e-1} p^{e-1}$$

on cada $z_i \in [0, n-1]$. Els dígits $z_0, z_1, \ldots, z_{e-1}$ són calculats tal i com ara explicarem. Primer calculem $X_0 = X^{\left(\frac{n}{p}\right)}$ i $Y_0 = Y^{\left(\frac{n}{p}\right)}$. Tenint en compte que l'ordre de X_0 és p, tenim:

$$Y_0 = Y^{\frac{n}{p}} = \left(X^{\frac{n}{p}}\right)^{\ell} = X_0^{\ell} = X_0^{z_0}.$$

Per tant $z_0 = log_{X_0} Y_0$ pot ser obtingut sol·lucionant el DLP en el cas $< X_0 >$. Després, calculem $Y_1 = (Y \Delta X^{-z_0})^{\frac{n}{p^2}}$. Així aconseguim:

$$Y_1 = (Y \Delta X^{-z_0})^{\frac{n}{p^2}} = X^{\frac{n}{p^2}(\ell - z_0)} = \left(X^{\frac{n}{p^2}}\right)^{(\ell - z_0)}$$

$$= \left(X^{\frac{n}{p^2}}\right)^{z_0 + z_1 p - z_0} = \left(X^{\frac{n}{p}}\right)^{z_1} = X_0^{z_1}.$$

Per tant $z_1 = log_{X_0} Y_1$ pot ser aconseguit solucionant el DLP en el cas $< X_0 >$. En general, si els dígits $z_0, z_1, \ldots, z_{t-1}$ han sigut calculats, llavors $z_t = log_{X_0} Y_t$, on: $Y_t = (Y \Delta X^{-z_0} \Delta X^{-z_1 x} \Delta X^{z_2 x^2} \Delta \cdots \Delta X^{z_{t-1} x^{t-1}})^{\frac{n}{p^{t+1}}}$.

3.2 Algorisme de Pohlig-Hellman per a corbes el·líptiques

Sigui E una corba el·líptica sobre $\mathbb{F}_p$ i P un punt de $E(\mathbb{F}_p)$ d'ordre n. Suposem que la factorització de n, sigui $n = p_1^{e_1} p_2^{e_2} ... p_r^{e_r}$. Com abans, l'estratègia de Pohlig Hellman per a calcular el logaritme discret $\ell = log_p Q$ és calcular $\ell_i = \ell \bmod p_i^{e_i}$ per a cada $1 \leq i \leq r$, i després solucionar el sistema de congruències:

$$\ell = \ell_1 \pmod{p_1^{e_1}}$$

$$\ell = \ell_2 \pmod{p_2^{e_2}}$$

$$\vdots$$

$$\ell = \ell_r \pmod{p_r^{e_r}}$$

per $\ell \in [0, n-1]$ tal i com hem explicat en el cas anterior, el Teorema Xinès del Residu garanteix una única sol·lució. El càlcul de cada ℓ_i pot ser reduïda al càlcul de e_i logaritmes discrets, en el subgrup d'ordre p_i de $< P >$. Per a simplificar l'expressió, escrivim p per denominar p_i i e per a e_i. Llavors calcularem la representació en base p, de ℓ_i de la següent manera::

$$\ell_i = z_0 + z_1 p + z_2 p^2 + \cdots + z_{e-1} p^{e-1}$$

on cada $z_i \in [0, n-1]$. Els dígits $z_0, z_1, \ldots, z_{e-1}$ són calculats tal i com ara explicarem. Primer calculem $P_0 = \left(\frac{n}{p}\right) P$ i $Q_0 = \left(\frac{n}{p}\right) Q$. Tenint en compte que l'ordre de P_0 és p, tenim:

$$Q_0 = \tfrac{n}{p}Q = \ell \left(\tfrac{n}{p}P\right) = \ell P_0 = \ell_i P_0 = z_0 P_0.$$

Per tant $z_0 = log_{P_0} Q_0$ pot ser obtingut sol·lucionant l'ECDLP sobre el subgrup $< P_0 >$. Després, calculem $Q_1 = \left(\frac{n}{p^2}\right)(Q - z_0 P)$. Així aconseguim:

$$Q_1 = \tfrac{n}{p^2}(Q - z_0 P) = \tfrac{n}{p^2}(\ell - z_0)P = (\ell - z_0)\left(\tfrac{n}{p^2}P\right)$$

$$= (z_0 + z_1 p - z_0)\left(\tfrac{n}{p^2}P\right) = z_1 \left(\tfrac{n}{p}P\right) = z_1 P_0.$$

Tal i com passava abans, $z_1 = log_{P_0} Q_1$ pot ser aconseguit solucionant l'ECDLP en el cas $< P_0 >$. En general, si els dígits $z_0, z_1, \ldots, z_{t-1}$ han sigut calculats, llavors $z_t = log_{P_0} Q_t$, on:

$$Q_t = \tfrac{n}{p^{t+1}}(Q - z_0 P - z_1 pP - z_2 p^2 P - \cdots - z_{t-1} p^{t-1} P).$$

Tal i com hem fet abans, una vegada obtinguts tots els ℓ_i, aplicant el Teorema Xinès del Residu, obtindrem el valor de ℓ.

3.3 Esquemes d'implementació

En aquest apartat, no aportarem la implementació exacta que s'ha fet de l'algorisme, sinò que en donarem la idea general, amb un parell d'esquemes mitjançant pseudo-codi, un per a la implementació de l'algorisme per a l'atac amb el grup multiplicatiu $\mathbb{F}_p^*$ i l'altre per al cas on tenim corbes el·líptiques.

3.3.1 Pohlig-Hellman sobre grup multiplicatiu $\mathbb{F}_p^*$

El següent algorisme en pseudo-codi correspon al mètode de Pohlig-Hellman sobre el grup multiplicatiu $\mathbb{F}_p^*$. Abans de començar amb l'algorisme haurem de complir una condició. Necessitarem trobar un punt p aleatori i primer, l'ordre del qual no sigui divisible per un primer gran, l'anterior condició sobre la semblança dels seus ordres no farà falta ja que aquí necessitarem un generador X d'aquest grup, i per tant els seus ordres seran els mateixos.

Entrada: Un primer p, un generador del grup X i un punt aleatori $(Y) \in (\mathbb{F}_p^*)$.

Sortida: El logaritme discret $\ell = log_X Y$.

Pohlig-Hellman (p, X, Y)
Calcular l'ordre de p, (n)
Calcular els factors de n, $(p_1, p_2, ..., p_r)$
per **a** $(p_1, p_2, ..., p_r)$ **fer**:
  Guardem p_i (la necessitarem per al càlcul final del Teorema Xinès del Residu)
  Calculem X_0
  mentres $(e_1, e_2, ..., e_r)$ **fer**:
    X_0^z
    Calculem $Y_e = R^{\frac{n}{p^{t+1}}}$
    si $((X_0^z) == Y_e)$ **fer**:
      Calculem ℓ_r (parcial o final)
      Calculem $R = (Y \cdot X^{-z_0} \cdot X^{-z_1 x} \cdot X^{z_2 x^2} \cdot \cdots \cdot X^{z_{t-1} x^{t-1}})$
    fsi
    $z++$
  fmentres
  Guardem ℓ_r
fper
Calcular Teorema Xinès del Residu
retorna ℓ

3.3.2 Pohlig-Hellman sobre el grup de punts d'una corba el·líptica.

El següent altorisme en pseudo-codi correspon al mètode de Pohlig-Hellman sobre el grup de punts d'una corba el·líptica definida sobre un cos $\mathbb{F}_p$. Abans de començar amb l'algorisme haurem de complir un parell de condicions. Necessitarem trobar un punt p aleatori i primer, que l'ordre de la corba definida sobre el cos $\mathbb{F}_p$ no sigui divisible per un primer gran, i que els ordres d'aquesta corba i d'un punt seu aleatori P siguin semblants.

Entrada: Un primer p, una corba el·líptica E $(\mathbb{F}_p)$, i dos punts P, Q de E $(\mathbb{F}_p)$.

Sortida: El logaritme discret $\ell = log_p Q$.

Pohlig-Hellman (p, P, Q)
Calcular l'ordre de p, (n)
Calcular els factors de n, $(p_1, p_2, ..., p_r)$
per **a** $(p_1, p_2, ..., p_r)$ **fer**:
Guardem p_i (la necessitarem per al càlcul final del Teorema Xinès del Residu)
Calculem P_0
mentres $(e_1, e_2, ..., e_r)$ **fer**:
$P_0 \Delta z$
Calculem $Q_e = \frac{n}{p^{t+1}} R$
si $((P_0 \Delta z) == Q_e)$ **fer**:
Calculem ℓ_r (parcial o final)
Calculem $R = (Q - z_0 P - z_1 pP - z_2 p^2 P - \cdots - z_{t-1} p^{t-1} P)$
fsi
$z++$
fmentres
Guardem ℓ_r
fper
Calcular Teorema Xinès del Residu
retorna ℓ

3.4 Dificultats trobades durant la implementació

En aquest apartat, comentarem les dificultats que s'han trobat a mesura que s'anava implementant l'algorisme de Pohlig-Hellman, amb la llibreria LiDIA:

- **void a.factor ( )** Si el nombre que volem que factoritzi és massa gran, és possible que dóni un missatge d'error. Ens hem trobat amb aquest cas, a partir de 19 xifres, encara que no acostuma a ser el més normal. A partir de 23, 24 xifres mentres va buscant un primer apropiat, és més probable que acabi donant el missatge d'error.

- **bigint C.group_order ( )** Si intentem calcular l'ordre per a grups de punts de corbes el·líptiques definides sobre cossos més grans de 20 dígits, el temps de càlcul s'incrementa de manera exagerada.

- **bigint order_point (const point< T > & P)** El mateix ens passa quant intentem calcular l'ordre d'un punt per a corbes sobre cossos finits grans.

4 Implementació

En aquest capítol veurem quines són les eines de software i hardware que hem utilitzat per a la implementació dels diferents algorismes comentats al capítol anterior. A més a més farem una introducció al seu codi font i parlarem del motiu de la utilització de la llibreria matemàtica LiDIA.

4.1 El software base utilitzat

La implementació de l'algorisme de Pohlig Hellman s'ha realitzat amb el llenguatge de programació C, a través de l'editor de text gedit versió 2.18.2. A l'hora de compilar s'ha utilitzat el compilador de lliure distribució GNU g++, inclòs a la distribució Red Hat Linux 7.2 (Enigma) Kernel 2.4.7-20 smp.

Per al tractament de números grans s'ha fet servir la llibreria matemàtica LiDIA i el sistema d'edició de documents s'ha realitzat amb LyX 1.5.1 el qual treballa sobre LaTeX.

4.1.1 Compilador GNU g++

GNU g++ és un compilador de lliure distribució (versió 2.96) que proporciona per defecte LINUX, i que ens permet la utilització d'una infinitat de llibreries, A més a més gràcies als comentaris anteriors sabem que tindrà un alt grau de portabilitat a qualsevol entorn o plataforma.

4.1.2 Llibreria matemàtica LiDIA

En aquest treball hem utilitzat la llibreria matemàtica LiDIA [10] per implementar el codi, ja que aquesta té implementats de forma eficient paquets per treballar amb enters grans, cossos finits, corbes el·líptiques, etc... LiDIA va ser desenvolupat pel Departament de Ciències de la Computació de la Universitat de Saarlandes, a Alemanya, l'any 1994.

La seva utilització és molt important en la realització d'aquest treball, ja que en C++ el número més gran que podem representar és de l'ordre de 2^{32} , degut a que els enters tenen una longitud de 32 bits. Aquest límit és totalment insucient quan es tracten problemes amb números molt grans.

Avui en dia LiDIA consta de molts paquets i cada paquet de diverses subllibreries. A continuació comentarem les que s'han utilitzat majoritàriament:

- **bigint.h:** És una aplicació del tipus predefinit *int* de C++, per tant totes les operacions disponibles pels enters també estaran pels elements de la classe *bigint*. Ens proporcionarà tota l'aritmètica necessària per treballar amb enters grans. Una variable *bigint* pot contindre un enter sense límit de dígits.

- **galos_fiend.h:** Ens permet definir i treballar amb cossos finits. Una variable del tipus galois_field representa un cos finit GF_p de p elements, així es poden realitzar sobre ell les operacions que habitualment es realitzen sobre un cos d'aquestes característiques.

- **gf_element.h:** Si el cos finit és definit mitjançant un polinomi $f(x) \pmod p$, llavors la variable gf_element conté la representació polinòmica dels elements del cos.

- **elliptic_curve.h:** Ens permet representar corbes el·líptiques definides sobre un cos $< T >$. Es poden definir de diferents formes, el model projectiu inclòs.

4.1.3 LyX

LyX és un programa gràfic multiplataforma creat per Matthias Ettrich que permet l'edició de text utilitzant LaTeX, pel qual «hereta» totes les seves capacittas (notació científica, edició d'equacions, creació d'índex, etc).

Es tracta d'un processador de textes en el qual l'usuari no necessita pensar en el format final del seu treball, sinó només en el contingut i la seva estructura (WYSIWYM) (Allò Que Veus És El Que Vols Dir, per les sigles amb anglès), per tant pot ser utilitzat per editar documents grans (llibres) o amb format rigurós (tesis, articles per a revistes científiques), amb facilitat.

4.1.4 LaTeX

TeX (implementat per Donald Knuth) és un sofisticat programa per preparar documents científics tals com articles, reportatges , llibres, treballs, ... LaTeX és un conjunt adequat de comandes TeX preparat per Leslie Lamport.

LaTeX no és un processador de text, és un programa que ens permet preparar automàticament un document d'aparença estàndard i d'alta qual·litat. En general, solament necessitem editar el text i algunes comandes, i LaTeX

s'encarrega de composar automàticament les formules del document. A diferencia d'un processador de text, tenim un control més sobre qualsevol aspecte tipogràc del document.

5 Resultats i Comentaris

Quan es finalitza la implementació d'un algorisme, el següent pas consisteix en avaluar si el treball realitzat compleix les espectatives inicials. Observant això, intentarem interpretar els diferents resultats obtinguts de les diverses proves realitzades amb els nostres algorismes, a més de donar les conclusions a que hem arribat a l'observar els resultats. Finalment, parlarem també sobre les conclusions més generals del TFC i de les futures línies de treball.

5.1 Resultats

Els números utilitzats per realitzar els càlculs s'han generat aleatòriament. Cada prova s'ha fet amb diferents nombres aleatoris i primers p per a crear diferents grups F_p.

5.1.1 Execució sobre $\mathbb{F}_p^*$

A la taula 5.2 mostrem els temps calculats de l'algorisme de Pohlig Hellman per diferents tamanys de p i dels factors del seu ordre n. La primera columna es refereix al nombre de digits del nombre primer p, i les altres columnes, al tamany del factor més gran de la descomposició de l'ordre n en primers.

dígits p	major factor de l'ordre n					
	3	4	5	6	7	8
6	0.01	0.01	0.03	1.38		
7	0.01	0.02	0.2	2.27	43.12	
8		0.01	0.07	1.12	22.53	438.09
9		0.01	0.08	1.08	35.89	159.45
10		0.06	0.87	1.62	10.51	325.75
11		0.14	0.45	1.01	41.34	265.51
12			0.27	9.14	91.15	86.14
13			0.48	2.06	25.58	45.86
14				11.45	32.3	584.55
15				2.05	8.66	858.56
16				5.54	18.95	112.48
17				9.15	21.48	1314.34
18				2.38	81.12	456.56
19				13.17	27.5	981.45
20				16.63	7.17	371.32

Taula 5.1: Temps mitjans de l'algorisme de Pohlig-Hellman per al grup $\mathbb{F}_p^*$.

5.1.2 Execució amb corbes el·líptiques

A la taula 5.1 mostrem els temps calculats de l'algorisme de Pohlig Hellman per diferents tamanys de p i dels factors del seu ordre n. La primera columna es refereix al nombre de digits del nombre primer p, i les altres columnes, al tamany del factor més gran de la descomposició de l'ordre n en primers.

dígits p	major factor de l'ordre n					
	2	3	4	5	6	7
6	0.01	0.07	0.33	4.68	25.36	
7	0.02	0.035	0.4	2.47	28.83	379.23
8		0.15	0.44	0.81	60.37	577.82
9		0.22	1.47	6.5	80.79	297.76
10		0.26	1.17	7.08	52.4	336.26
11			0.98	10.85	114.96	486.62
12			4.93	27.94	92.85	565.23
13			2.25	33.18	133.84	487.25
14			7.23	15.24	126.45	687.15
15			12.3	12.1	86.14	798.26

Taula 5.1: Temps mitjans(en segons) de l'algorisme de Pohlig-Hellman amb corbes el·líptiques

Pel que es pot comprobar a la taula, el temps de l'algorisme no depèn del tamany de p, sinó més exactament del tamany dels factors de n. El fet de que hi hagi temps desequilibrats i que per a un factor més gran el temps sigui menor, és degut a que la cerca de cada z és una búsqueda exhaustiva, així que depenen de en quin moment trobi la z correcta, tardarà més o menys en acabar d'executar el programa. Si féssim centenars de proves per cada dígit, segur que ens surtiria una taula més equilibrada. Això també ens passava en l'anterior execució sobre el grup multiplicatiu $\mathbb{F}_p^*$.

5.2 Conclusions

En aquest treball de fi de carrera s'ha implementat l'algorisme de Pohlig-Hellman en el llenguatge C, tant en el grup multiplicatiu $\mathbb{F}_p^*$ com sobre el grup de punts d'una corba el·líptica sobre $\mathbb{F}_p$. Tot seguit, després d'analitzar els resultats que es presenten a l'apartat anterior, sembla oportú comentar quines són les conclusions que s'en poden extraure d'aquest treball de final de carrera.

Quan vam iniciar aquest treball no sabiem quins serien els resultats obtinguts. Doncs bé, una vegada acabat l'anàlisi dels resultats podem dir que la nostra implementació de l'algorisme de Pohlig-Hellman permet calcular el *logaritme discret* sobre un grup $\mathbb{F}_p^*$ del qual el factor més gran del seu

ordre $p-1$ no superi els 8 dígits. Per al cas del grup de punts d'una corba el·líptica el factor més gran de l'ordre d'un generador ha estat de 7 dígits. Si superem aquestes xifres els temps de càlcul es disparen. Una altra cosa que volem remarcar, és que resulta difícil trobar primers p grans tal que $p-1$ o l'ordre d'una corba sobre $\mathbb{F}_p$ factoritzi en factors petits.

5.3 Futures línies de treball

A continuació, com a futures línies de treball, presentem algunes idees per millorar i ampliar aquest TFC.

- Editar les llibreries de LiDIA, creant noves funcions o modificant les existents, per millorar aquelles amb les hem sofert algun tipus de problema durant la implementació d'aquest treball.
- Implementar altres algorismes per atacar al DLP, com per exemple l'Índex Calculus el qual, tal i com hem comentat abans, explota mètodes per representar els elements del grup $\mathbb{F}_p^*$ com productes d'elements d'un sistema relativament petit.

Bibliografia

[1] Viquipèdia, l'Enciclopedia Lliure. *Corba el·líptica.*

[2] Sun zi. *Master Sun's Mathematical Manual*, any 300 aproximadament.

[3] W. Diffie and M.E Hellman. *New directions in cryptography.* IEEEE Trans. Inform. Theory, 1976.

[4] R. Delicata and S. Schneider. *A formar approach for reasoning about a class of Diffie-Hellman protocols.* Dep. of Computing, University of Surrey.

[5] W. Diffie, P.C. van Osrschot and M.J. Wiener. *Authentication and Authenticated Key Exchanges.* Sun Microsystems, 1992.

[6] A.J.Menezes, P.C. van Osrschot and S.A.Vanstone. *Handbook of Applied Cryptography*, 1996.

[7] J.A.Gras. *Ataques al problema del logaritmo discreto y estudio de su seguridad en protocolos criptográficos modernos.* TFC, Universitat de Lleida,2003.

[8] J. Nakahara Jr. *A Short overview of the Discrete Logarihm Problem.* Katholieke Universiteit Leuven. Departament of Electrical Engineering - ESAT SISTA/COSIC Research Group.

[9] LiDIA-Group. *LiDIA Manual A library for computational number theory.* Tech. Univ. Darmstadt. 2001.

[10] T. Bigordà. *Atac al problema del logaritme discret mitjançant la Rho de Pollard.* Treball de Fi de Carrera, Universitat de Lleida, Setembre de 2005.