Escola politècnica Superior
Universitat de Lleida
Enginyeria Informàtica

Criptografia lliure amb corbes el·líptiques

Author:
Sergi
Blanch i Torné

Director:
Ramiro
Moreno Chiral

Versió 1.0.0
Setembre 2008

Índex

Agraïments

Sempre agraït a en Ramiro per la llibertat donada per al desenvolupament i a l'esforç fet per fer-me entendre els conceptes necessaris per després poder escriure *la* lÃnia de codi i estalviar infinitats de temps en implementació. El considero un mentor i un gran amic amb qui compartir converses molt més enllà de la corba el·líptica. Realment és un referent amb qui es pot parlar absolutament de tot.

Merci també als companys Joan Valduvieco, Jordi Llonch i Albert Comerma. Dos d'ells per cedir-me infraestructura de la seva empresa `laigu` on allotjar el *subversion* la web i els pegats per al *GnuPG*. Peró molt agraït als tres per aguantar les més pesades explicacions sobre criptografia tot i que no en tinga ni idea.

Gràcies al Josep Rodrigo per enredar-me a fer-me pilot d'avionetes aquest últim any. Es un gran des-estressant aquesta activitat.

Merci també, als companys de feina del sincrotró, en especial a en Jordi Benach per l'esforç al llegir un *draft* i enviar correccions per que aquesta documentació llueixi millor.

Als pacients de *Cal Curcó* (la família) i la meva companya Helena, per l'eterna comprensió mostrada en front la meva dificultat de desconnexió després una llarga jornada combinant treball i estudi, quan el meu caràcter no és el més amable.

Capítol 1

Introducció i objectius

Aquest projecte és un pas més en l'evolució d'un projecte major de criptografia de corba el·líptica en programari lliure. El primer pas es va realitzar quan el curs 2003-2004 és va presentar [BM04] com a projecte final de carrera de l'enginyeria tècnica un pegat per al *GnuPG* que ampliava l'oferta criptogràfica d'aquest programa lliure cap a les corbes el·líptiques. Ja llavors no es volia que la feina s'acabes amb la publicació de la memòria i prou, i tant l'alumne com el director teníem tota l'intenció de seguir-hi.

Durant el temps que l'estudiant ha realitzat les assignatures corresponets al segon cicle d'informàtica, en paral·lel s'ha mantingut una senzilla web[1] com a repositori públic de la feina realitzada. Així com s'ha anat actualitzant les versions del pegat de corba el·líptica també s'ha afegit documentació sobre aquestes actualitzacions en forma d'articles que s'han presentat a conferències i congressos per tal de publicitar la feina i reportar que el projecte segueix viu.

Les principals correccions i millores són el que avui compendien aquest projecte en forma de treball final de carrera. S'ha treballat per buscar-hi errors i el públic escrutini de tenir el codi disponible va donar els seus fruits amb el descobriment per part d'en *Mikael Mylnikov* d'una errada en el procediment de firma digital. Posteriorment es converteix en col·laborador al aportar una solució a la debilitat de l'esquema de xifrat inicialment plantejat que culmina amb l'article [BM06] que justifica el perquè de l'error i el perquè de la solució proposada.

També posteriorment a aquest criptoanàlisi és va posar de moda una nova perspectiva per atacar els criptosistemes en general. Els anomenats atacs laterals del capítol 3 són formes de trencar una clau privada trencant una implementació degut a l'entorn on s'executa. Resulta de gran bellesa tècni-

[1] http://www.calcurco.cat/eccGnuPG/

ca ja que aprofita petits detalls d'enginyeria per resoldre d'una forma molt creativa i econòmica el que matemàticament resultava impossible i inviable econòmicament. Des d'aquest projecte també s'han tractat les contramesures des d'un punt de vista tant tècnic sobre la implementació, com recorrent a recursos matemàtics per invalidar alguns d'aquests intents laterals de criptoanàlisi.

Una avantatge molt important de les corbes el·líptiques que no es explotada per les implementacions en general s'està intentant resoldre des d'aquí. El setup dels criptosistemes el·líptics, d'ara endavant *ECC*s, sempre son preestablerts per a l'usuari i se li donen corbes el·líptiques base amb les que fer criptografia. Això fa que la gran avantatge de canviar de corba per un usuari el deixi obligat a la mateix solució que en el cas dels cossos finits amb l'augment de la longitud de la seva clau. En el capítol 4 es busca de treballar amb conjunts de corbes el·líptiques per buscar que l'usuari final tingui la possibilitat de canviar de corba el·líptica sense tenir que renunciar a una longitud que consideri adequada. Com s'explicarà amb detall en el corresponent capítol, encara no hi ha una implementació en el projecte que es dongui com a bona per a us públic.

Finalment, un punt d'inflexió en aquest projecte és la inclusió directa dins de la rama de desenvolupament de la llibreria matemàtica *libgcrypt* del *GnuPG* i la signatura d'una *assigment* amb la *Free Software fundation* (FSF) per la participació directa i activa en el projecte *Gnu*. Amb aquesta signatura es fa un pas de gegant per permetre aquest projecte existir dins del software lliure amb independència. Totes dues parts guanyem amb aquesta unió, facilitant l'accès i el desenvolupament; i també conseguint suport per a corbes el·líptiques en aquest software mundialment extès.

Capítol 2

Criptologia de corba el·líptica

Abans de poder entrar a l'algorísmica de les corbes el·líptiques s'han de prefilar una introducció a l'aritmètica d'aquestes. Com a convenció s'ha utilitzat una notació tipus a la apareguda en la normativa [P1363] que serà aprofundida en la secció 2.2.1 junt amb altres normatives que influeixen les implementacions de corba el·líptica.

2.1 Aritmètica de corba el·líptica

Tota la base necessària per comprendre i treballar amb les corbes el·líptiques és l'algebra. Començant des del principi es donen alguns conceptes per coneguts i es parteix de la definició de group o conjunt amb una llei de composició interna.

Definició 2.1. Donat un conjunt G i una llei de composició interna $\oplus$, anomenem grup $(G, \oplus)$ si compleix:

1. $\forall x, y, z \in G, \oplus$ és associativa: $(x \oplus y) \oplus z = x \oplus (y \oplus x)$.
2. $\exists e_0 \in G : \forall x \in G$ tal que $x \oplus e_0 = e_0 \oplus x = x$.
3. $\forall x \in G, \exists y \in G$ que és l'invers tal que $x \oplus y = y \oplus x = e_0$.

A més, nomenem a un grup *Abelià* quan a més de les propietats anteriors, l'operació es commutativa.

L'estructura sobre la que es realitza l'aritmètica de corbes el·líptiques és un grup. Ampliant la primera operació que pot tenir un grup, podem donar-li una segona llei de composició interna de manera que formi un anell.

Definició 2.2. Donat un conjunt A i dues lleis de composició interna $\oplus$ i $\otimes$, anomenem anell a $(A, \oplus, \otimes)$ si compleix:

1. $(A, \oplus)$ és un grup Abelià.
2. $\exists e_1 \in A,\ e_1 \neq e_0 : \forall x \in G$ tal que $x \otimes e_1 = e_1 \otimes x = x$.
3. $\forall x, y, z \in G$ l'operació $\otimes$ és distributiva respecte a $\oplus$: $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$.

L'exemple clàssic d'anell és el nombre enters, $(\mathbb{Z}, +, \cdot)$. Resulta evident es següent resultat:

Proposició 1. *Sí* $(\mathbb{A}, \oplus, \otimes)$ *és un anell, el conjunt* $\mathbb{A}^*$ *dels seus elements invertibles és un grup amb* $\otimes$.

A partir d'aquí la definició de *cos* és natural:

Definició 2.3. Definim un cos $\mathbb{K}$ com un anell on cada element no nul es invertible, es a dir $\mathbb{K}^* = \mathbb{K} \setminus \{e_0\}$.

L'estructura algebràica sobre la que definirem el grup de punts d'una corba el·líptica és un cos. Tenim molts cossos en la nostra aritmètica. Per exemple, amb dues operacions com la suma i el producte, tenim els nombres racionals, $\mathbb{Q}$, els reals, $\mathbb{R}$ o els complexes, $\mathbb{C}$. De manera general un cos amb dues operacions el denotem $(\mathbb{K}, \oplus, \otimes)$.

2.1.1 Cossos finits

Un cop tenim l'algebra bàsica definida podem acurar més el seu entorn i afegir restriccions per obtenir les propietats que necessitem. Entrarem així a l'aritmètica modular amb la definició dels *cossos finits* o *cossos de Galois*.

Definició 2.4. Un cos finit és un cos amb un nombre finit d'elements.

Al nombre d'aquests elements l'anomenarem *ordre* del cos. Denotarem els cossos finits com $\mathbb{F}$ o be $\mathbb{F}_q$, si es vol deixar palés el seu ordre, $|\mathbb{F}_q| = q$. S'obté el següent resultat fonamental,

Teorema 2. *Per tot primer* p *i qualsevol* $r \in \mathbb{Z}_{>0}$ *existeix un cos finit amb* $q = p^r$ *elements. Aquest cos és únic i es denota com* $\mathbb{F}_q$. *Reciprocament, si* $\mathbb{F}_p$ *és un cos finit es té que* $|\mathbb{F}| = q = p^r$, *on* p *és un enter primer, anomenat* característica, *i* $r \in \mathbb{Z}_{>0}$, *conegut com* extensió *o* índex de l'extenció.

En endavant usarem cossos finits en els que l'extensió es $r = 1$, per tant es tracta de cossos d'ordre primer, $q = p$: són els cossos finits *primers*. L'aritmètica en aquests cossos és l'aritmètica módul p, i.e., $\mathbb{F}_p$ es pot identificar

com $\mathbb{Z}/p\mathbb{Z}$. La criptografia de clau pública clàssica ha usat aquest cossos primers, més precisament el grup multiplicatiu $\mathbb{F}_p^*$, en els criptosistemes basats en el problema anomenat del *logaritme discret*. I la criptografia amb corbes el·líptiques torna a usar-los en el basats en el *logaritme discret el·líptic*: ara el grup es troba constituït pel conjunt de punts de la corba el·líptica *definida sobre* $\mathbb{F}_p$.

2.1.2 Plans afins i projectius

Abans d'entrar a definir les corbes el·líptiques es fa un repàs sobre geometria en el pla, en especial perquè servirà per entrar a definir uns plans més interessant per a la corba el·líptica. Clàssicament, quan parlem de plans pensem en dues dimensions on cada punt pot ser definit per un parell ordenat d'elements d'un conjunt. Més concretament per situar un punt sobre un pla real utilitzariem un parell ordenat de valors reals, $(x, y) \in \mathbb{R} \times \mathbb{R}$. Però hi ha altres tipus de plans com són els plans projectius que utilitzen una terna ordenada d'elements del conjunt per definir un punt bidimensional.

Definició 2.5. Donat un cos $\mathbb{F}$ el *pla projectiu* sobre $\mathbb{F}$, $\mathbb{P}_2(\mathbb{F})$, es defineix com el quocient

$$\mathbb{P}_2(\mathbb{F}) = \frac{\mathbb{F}^3 \setminus \{(0,0,0)\}}{\sim},$$

on $\sim$ és la relació d'equivalència definida en $\mathbb{F}^3 \setminus \{(0,0,0)\}$ com $(x, y, z) \sim (x', y', z')$ si existeix $\lambda \in \mathbb{F}^*$, tal que $x = \lambda x', y = \lambda y', z = \lambda z'$.

Un punt projectiu és denota com $[X : Y : Z] \in \mathbb{P}_2(\mathbb{F})$. La recta $Z = 0$ s'anomena *recta en l'infinit*. Es pot fer una aplicació entre els punts del pla afí, $\mathbb{A}_2(\mathbb{F}) = \mathbb{F}^2$, i els del pla projectiu $\mathbb{P}_2(\mathbb{F})$,

$$\begin{array}{rccl} f: & \mathbb{A}_2(\mathbb{F}) & \longrightarrow & \mathbb{P}_2(\mathbb{F}) \\ & (x, y) & \longmapsto & [x : y : 1]\,. \end{array}$$

Però la recíproca no és una aplicació ja que resulta evident que la recta projectiva en l'infinit to té representació en el pla afí,

$$\begin{array}{rccl} f_{-1}: & \mathbb{P}_2(\mathbb{F}) & \longrightarrow & \mathbb{A}_2(\mathbb{F}) \\ & [X : Y : Z] & \longmapsto & \begin{cases} \left(\frac{X}{Z}, \frac{Y}{Z}\right), & \text{si } Z \neq 0, \\ \nexists, & \text{si } Z = 0. \end{cases} \end{array}$$

2.1.3 Definint corbes el·líptiques

Ja tenim tot el necessari sobre cossos finits així com també sobre plans projectius i ara ens podem posar a *dibuixar* corbes per aquest pla com a subconjunts dels elements d'aquests plans que compleixen una regla particular:

Definició 2.6. Una corba el·líptica $E/\mathbb{F}_p$ està definida en $\mathbb{P}_2(\mathbb{F}_p)$, per una equació en la *forma normal de Weierstraß (Weierstraß Normal Form, WNF)*,

$$F(X,Y,Z) = \\ Y^2Z + a_1XYZ + a_3YZ^2 - X^3 - a_2X^2Z - a_4XZ^2 - a_6Z^3 = 0, \quad (2.1)$$

amb coeficients $a_i \in \mathbb{F}_p$ i sense punts singulars, que vol dir que no resól el següent sistema d'equacions:

$$\left.\begin{array}{c} \frac{\partial F(X,Y,Z)}{\partial X} = 0 \\ \frac{\partial F(X,Y,Z)}{\partial Y} = 0 \\ \frac{\partial F(X,Y,Z)}{\partial Z} = 0 \\ F(X,Y,Z) = 0 \end{array}\right\}$$

Hi ha reduccions de l'equació WNF, com per exemple la que hem utilitzat principalment en aquest projecte,

Definició 2.7. Per cos finit de característica diferent de 2 i 3, l'equació 2.1 pot substituir-se per la *forma reduïda de Weierstraß (WRF)*,

$$Y^2Z = X^3 + aXZ^2 + bZ^3, \quad (2.2)$$

i es pot traduïr a un pla afí,

$$\frac{Y^2}{Z^2} = \frac{X^3}{Z^3} + a\frac{X}{Z} + b, \quad \boxed{\begin{array}{ccc} \frac{Y}{Z} & \to & y \\ \frac{X}{Z} & \to & z \end{array}} \Rightarrow y^2 = x^3 + ax + b. \quad (2.3)$$

La condició de no–singularitat és equivalent a la de no anulació del discriminant, que en la WRF se pot escriure,

$$\Delta = -\left(4a^3 + 27b^2\right) \neq 0. \quad (2.4)$$

Donada l'equació de la corba el·líptica és moment de definir el conjunt de punts $E(\mathbb{F}_p)$, ja que serà amb aquest conjunt amb el que després voldrem fer criptografia.

Definició 2.8. El conjunt de punts de una corba el·líptica $E/\mathbb{F}_p$: $y^2 = x^3+ax+b$, que denotem $E(\mathbb{F}_p)$, és el conjunt de punts del pla que compleixen amb l'equació de la corba, juntament amb el punt en l'infinit, en el pla afí, $\mathbb{A}_2$,

$$E(\mathbb{F}_p) = \left\{(x,y) \in \mathbb{F}_p^2 :\ y^2 = x^3 + ax + b\right\} \cup \left\{\mathcal{O}_E\right\}. \quad (2.5)$$

I en el pla projectiu, $\mathbb{P}_2$,

$$E(\mathbb{F}_p) = \left\{[X:Y:Z] \in \mathbb{P}_2(\mathbb{F}_p) : Y^2Z = X^3 + aXZ^2 + bZ^3\right\}. \quad (2.6)$$

2.1.4 Grup de punts

Definides les corbes el·líptiques ja podem entrar a veure que podem fer amb elles. D'una forma genèrica primer, i entrant en l'us criptogràfic després. Per visualitzar les operacions entre punts d'una corba el·líptica, el més eficaç és veuren la seva representació gràfica en el pla real $\mathbb{R}^2$.

Definició 2.9. Donats dos punts racionals $P \neq Q$ d'una corba el·líptica E, la línia recta que passa per ambdos ha de intersecar la corba el·líptica per un tercer punt racional, l'oposat del qual es la suma $P + Q$.

En la figura 2.1.1 es pot veure un exemple gràfic de com seria una suma de dos punts d'una corba el·líptica.

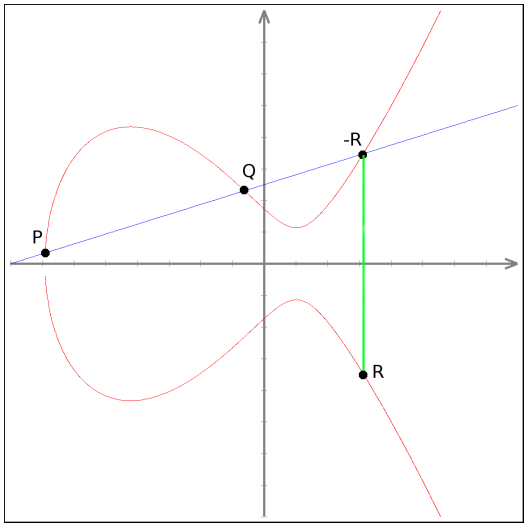


Figura 2.1.1: Suma de punts d'una corba el·líptica

Definició 2.10. Donat un punt racional P d'una corba el·líptica E, la línia recta tangent a aquell punt de la corba interseca la corba el·líptica per un segon punt racional, l'invers del qual es la suma del punt P amb si mateix, $[2]P = P + P$. Si la tangent resulta ser una recta vertical, la intersecció és amb el *punt en l'infinit*, $\mathcal{O}$, que també pertany a la corba però no te representació sobre el pla afí: es diu que P és d'ordre 2 ja que $[2]P = \mathcal{O}$.

En la figura 2.1.2 es pot veure un exemple gràfic de com seria un doblat d'un punt d'un corba el·líptica.

Existeix, per suposat, una formulació analíica d'aquesta suma en $E(\mathbb{F}_p)$, les equacions de la qual, que son les que s'implementen, és poden veure per exemple en [HANDECC, LMS265].

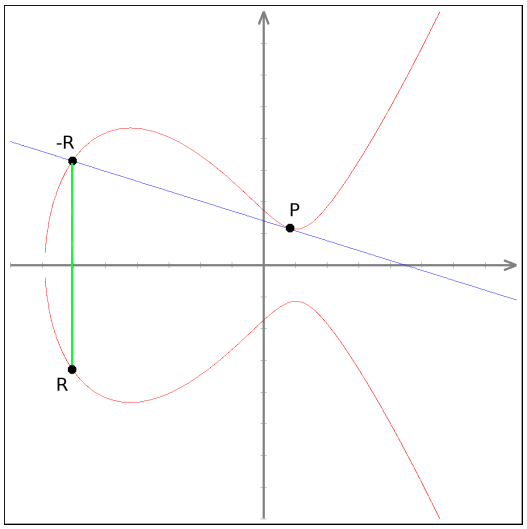


Figura 2.1.2: Duplicat d'un punt d'una corba el·líptica

Proposició 3. *El conjunt de punts d'una corba el·líptica amb aquesta suma,* $(E(\mathbb{F}_p), +)$*, és un grup abelià on el neutre és el punt en l'infinit,* $\mathcal{O}_E$*.*

L'*ordre* del grup de punts de $E(\mathbb{F}_p)$ o "cardinal de la corba el·líptica", que denotem com $|E(\mathbb{F}_p)|$, és una data que s'ha de saber per a la posada en funcionament, o *setup*, de un ECC. El seu càlcul és fa mitjançant l'algorisme SAE, però per la seva complexitat resulta freqüènt ([NIST186]) que sigui una dada que acompanyi als coeficients i la resta de dades necessàries per definir la corba el·líptica. De tota manera, el *teorema de Hasse* ens acota sempre aquest valor del cardinal de la corba.

Teorema 4 (Hasse)**.** *Donada una corba el·líptica* $E/\mathbb{F}_q$*, es compleix que*

$$q + 1 - 2\sqrt{q} \leq |E(\mathbb{F}_q)| \leq q + 1 + 2\sqrt{q}. \tag{2.7}$$

Aquest teorema permet un enunciat alternatiu usant el paràmetre t, *traça del endomorfisme de Forbenius*: $|E(\mathbb{F}_q)| = q + 1 - t$, on $|t| \leq 2\sqrt{q}$. Aquí es veu clar que $|E(\mathbb{F}_q)| \approx q$, es a dir, l'ordre del grup de punts de la corba és aproximadament l'ordre del cos finit sobre el que està definida.

Donades del definicions 2.9 i 2.10 podem deduir que al punt racional resultant d'un doblat li podem tornar a sumar el mateix. Ens anirà be tenir una operació *suma repetida*.

Definició 2.11. Donat $n \in \mathbb{Z}$ i un punt P d'una corba el·líptica E, anomenem *producte per un escalar* a la suma del punt P amb si mateix un nombre

n de vegades,

$$[n]P = \begin{cases} \overbrace{P + \cdots + P}^{n)}, & \text{si } n > 0, \\ \mathcal{O}, & \text{si } n = 0, \\ \overbrace{(-P) + \cdots + (-P)}^{n)}, & \text{si } n < 0. \end{cases}$$

És amb aquesta operació *producte d'un punt per un escalar* sobre la que tot gira. Més endavant, a partir de la definició 2.12, veurem com pren forma el criptosistema en base aquesta definició 2.11.

2.1.5 Logaritme discret el·líptic

Definides les operacions que podem realitzar amb les corbes el·líptiques anem a entrar en la criptografia. Per poder fer criptografia de corba el·líptica necessitem el mateix que es fa anar quan es planteja sobre cossos finits: allí s'usa el grup multiplicatiu del cos, $\left(\mathbb{F}_p^*, \cdot\right)$, que és sempre un grup cíclic d'ordre $p-1$; amb corbes el·líptiques tindrem també un subgrup cíclic del grup de punts de la corba el·líptica i sobre aquest es planteja el logaritme discret el·líptic.

Definició 2.12. Definim un *subgrup cíclic* d'una corba el·líptica com el conjunt de punts generat per un $G \in E(\mathbb{F}_p)$,

$$\langle G \rangle = \{G, [2]G, [3]G, \ldots, [n]G = \mathcal{O}_E\}, \tag{2.8}$$

on n és l'ordre d'aquest subgrup cíclic,

$$n = \operatorname{ord}(G) = \min\{r \in \mathbb{Z}_{>0} : \ [r]G = \mathcal{O}_E\}.$$

Per "aprofitar" els bits que es disposa és necessita un ordre n gran, $n \approx p$, degut a que, como s'ha dit més amunt, $|E(\mathbb{F}_p)| \approx p$. Pel *teorema de Lagrange* sabem que $\operatorname{ord}(G) \mid |E(\mathbb{F}_p)|$, es a dir, que l'orden del subgrup cíclic sobre el que es planteja el criptosistema divideix a l'orden de tot el grup de punts.

Definició 2.13. És defineix el *cofactor* com la relació entre el cardinal de la corba el·líptica i l'ordre que té aquest subgrup,

$$h = \frac{|E(\mathbb{F}_p)|}{n}. \tag{2.9}$$

Recordant l'operació *producte per un escalar* definida en 2.11 per qualsevol corba el·líptica, ara podem definir el logaritme discret el·líptic per una corba el·líptica definida sobre un cos finit primer 2.4 que contingui un subgrup cíclic de punts d'aquesta corba el·líptica.

Definició 2.14. Donat un punt Q, d'ordre n, del conjunt de punts d'una corba el·líptica $E(\mathbb{F}_p)$ i un segon punt R del conjunt, definim el logaritme discret el·líptic com la resolució del valor x tal que:

$$R = [x]\,Q \tag{2.10}$$

Ja tenim tots els elements necessaris per definir una estructura de dades per a la criptografía de corba el·líptica:

Definició 2.15. Definim *criptogràficament* un sistema de corba el·líptica com una sèxtupla composada per:

$$\{p, a, b, G, n, h\} \tag{2.11}$$

En aquesta definició, p és el nombre primer sobre el que es defineix el cos finit $\mathbb{F}_p$; a i b son els parametres de la corba el·líptica de la definició de l'equació reduïda de Weierstraß (2.2); G és el punt generador del subgrup cíclic segons s'ha vist en la definició 2.12; n és l'ordre d'aquest generador i el cofactor h definit en 2.13.

2.2 Criptografia amb corbes el·líptiques

Molts criptosistemes moderns utilitzen cossos finits per definir-hi un problema computacionalment molt dur del qual beneficiar-se però també es poden utilitzar altres estructures per aquest propòsit. Aquí és mostra com utilitzar aquestes corbes el·líptiques sobre cossos finits per fer criptografia.

2.2.1 Normatives

Hi ha diverses normes i estàndard que ens especifiquen com implementar les corbes el·líptiques de cara a l'interoperabilitat, així com també a evitar certes debilitats que podrien ser nefastes per al conjunt d'implementacions.

2.2.1.1 "P1363" del IEEE

La norma [P1363] posada ampliament en discusió pública que manté el seu espai dins l'IEEE comença la seva història l'any 1994 i es publica l'any 2000

la versió definitiva. El grup de treball no queda, però, parat i es manté actiu fins a data d'avui ampliant els estàndard proposats com la P1363a o les normes 1363.1, 1363.2 i 1363.3.

Principalment per al nostre propòsit, el que aquest estàndard conté son especificacions i implementacions d'algorismes en llenguatge formal per a ser seguides en les implementacions com a sistemes òptims i públicament cribats i corregits. De tota manera no està exempt d'haver-hi de treballar-hi amb compte ja que alguns dels seus algorismes estan subjectes a patents i deixen reduït l'ús que se'n pot fer.

La norma no neix per a les corbes el·líptiques, sinó que les inclou com un sistema més. Resulta especialment interessant l'apendix A que conté algorísmica de les funcions matemàtiques bàsiques que després són usades des dels mètodes criptogràfics. Inclús són especificades les funcions matemàtiques per realitzar les operacions sobre diferents representacions del pla, ja sigui l'afí ($\mathbb{A}_2$) o el projectiu ($\mathbb{P}_2$).

2.2.1.2 "FIPS 186" del NIST

L'última versió del document sobre la signatura digital del NIST que depén del Departament de Comerç dels Estats Units (EEUU) és un *draft* del març del 2006, sota el nom de FIPS PUB 186-3. Anteriorment es va publicar l'any 2000 una primera expansió de l'estàndard sobre signatura digital conegut sota el nom FIPS PUB 186-2. El seu periple comença l'any 1991 amb la publicació proposada pel NIST que és adoptada l'any 1993 i se li apliquen canvis menors l'any 1996.

El que és més interessant en el cas de les corbes el·líptiques es l'apendix *E* d'aquest document que detalla els mínims que ha de seguir qualsevol institució federal d'EEUU. Les corbes el·líptiques especificades en aquest document són les utilitzades per altres normes i les primeres que és van fer anar en el pegat lliure per al GnuPG [BM04].

2.2.1.3 "ECC in OpenPGP" del IETF

Recentment s'ha publicat un nou RFC, el [rfc4880], que substitueix el 2440. Tot i això, aquest nou estàndard d'interoperabilitat entre aplicacions criptogràfiques no ha inclòs, encara, l'ús de corbes el·líptiques. En l'òrgan mantenedor d'aquestes normatives, el IETF, s'ha començat a treballar en aquest sentit per tal de madurar el que futurament s'hi inclourà, amb la creació del document de típus *Internet Draft* que porta el títol "ECC in OpenPGP".

Aquest document especifica el que qualsevol aplicació típus *PGP* complirà i el *GnuPG* a més de complir-la n'és part implicada en l'elaboració. L'últim

draft d'aquest document és de l'abril d'aquest any 2008, i el seu següent pas es donarà a l'octubre, ja sigui tirant endavant cap a l'estandardització, revisant-se de nou, o caient en l'oblit (tot i que aquesta última opció avui per avui no sembla tenir cap punt).

En aquest estàndard les corbes el·líptiques triades són algunes de les especificades anteriorment en [NIST186], concretament les definides sobre cossos finits primers, anomenades *Curve P-256*, *Curve P-384* i *Curve P-521*, descartant així les de menor longitud. Resulta especialment interessant el sistema de xifrat emprat.

El sistema de xifrat *ElGamal* presenta debilitats quan el traduïm per usar-lo amb corbes el·líptiques, fet que va ser estudiat en [BM06] i on els autors vam proposar un sistema alternatiu basat en l'ús d'una operació més robusta que substituís el producte modular del ElGamal, com pot ser un xifrat *AES256* on s'utilitza com a clau un *hash* del resultat de l'intercanvi de claus *Diffie-Hellman* que es fa anteriorment.

Les diferències que presenta l'algorisme de la norma [ECPGP] amb el proposat en [BM06] son l'ús escalat dels algorismes *AES* incrementant la longitud en relació a l'augment de la longitud de la corba el·líptica, així com també un molt intel·ligent ús de un punt aleatori a més de l'escalar aleatori en el cos finit per tal de garantir que el xifrat sigui semànticament segur.

La publicació d'aquest estàndard suposa en gran avenç ja que permetrà realitzar implementacions interoperables entre elles. A la vegada i per tal de reforçar l'impacte d'aquest estàndard, el *GnuPG* inclourà suport per aquest estàndard implementat per l'alumne d'aquest projecte, que forma part del grup de desenvolupament com s'explicarà en la seva corresponent secció.

No es pot deixar passar l'oportunitat de posar per escrit aquí algunes de les peticions d'aplicació que aquest estàndard ja té. És un punt que ja està previst, com ho destaca el fet que en la codificació d'un punt d'una corba el·líptica s'indica per una banda l'us de coordenades afins així com també coordenades comprimides i deixa la porta oberta a l'inclusió d'altres possibilitats.

Presenta una limitació, per un futur ús d'isogènies de corba el·líptica, ja que pressuposa que el cofactor sempre és *1*. Això és cert per totes les corbes el·líptiques que s'utilitzen aquí ara, però no té perquè ser cert sempre i per a tota corba el·líptica usada.

2.2.2 Firma digital ECDSA

L'algorisme de firma digital *ECDSA* és equivalent a l'algorisme *DSA* que està definit amb operacions en un cos finit, mentre que el que aquí s'explica està definit per a punts de corba el·líptica. Es tracta del mateix problema

a resoldre, es vol enviar una informació amb garanties de mantenir-ne la integritat, la seva autenticitat i el no rebuig.

Així doncs com algorisme de firma que s'anirà emparellat amb la informació que es vol signar. Per fer la firma es comença per aplicar una funció de resum a la informació per garantir-ne la integritat per després fer ús de la clau privada per donar autenticitat i com tenim una criptografia al darrera tindrem el no rebuig a no ser que la clau es vegi compromesa.

ALGORISME 1 (Signatura ECDSA)

INPUT: Clau secreta $skey_M$ i hash del missatge $\#hash$.
OUTPUT: Parell (r, s)/*tal que $0 < r, s < skey_A.n$*/.

1: Generar aleatoriament una clau de sessió $k \in_R [1, (skey_M.p) - 2]$;
2: $I \leftarrow [k]\,(skey_M.G)$;
3: $i \leftarrow I_x$;
4: $r \leftarrow i\,(mod\, skey_M.n)$;
5: **si** $r = 0$ **llavors**
6: goto 1;
7: **fi si**
8: $s \leftarrow k^{-1} \cdot (\#hash + (skey_M.d) \cdot r)\,(mod\, skey_M.n)$;
9: **si** $s = 0$ **llavors**
10: goto 1;
11: **fi si**.
12: Return (r, s);

Després d'haver signat una informació el receptor, o nosaltres mateixos per revisar la integritat, es repetirà el hash i amb la clau pública es procedirà a comprovar que d'aquella signatura es pot recuperar també el resultat del hash, de manera que sols hi ha coincidència si la firma és correcta.

ALGORISME 2 (Verificació ECDSA)

INPUT: Clau pública $pkey_M$, hash del missatge $\#hash$ i el parell (r, s).
OUTPUT: Booleà d'acceptació o rebuig.

1: Verificar $(r, s) \in [1, (pkey_M.n) - 1]$;
2: $h \leftarrow s^{-1} \, (mod \, pkey_M.n)$;
3: $h_1 \leftarrow (\#hash) \cdot h \, (mod \, pkey_M.n)$;
4: $h_2 \leftarrow r \cdot h \, (mod \, pkey_M.n)$;
5: $Q \leftarrow [h_1] \cdot (pkey_M.G) + [h_2] \cdot (pkey_M.G)$;
6: **si** $Q = \mathcal{O}_E$ **llavors**
7: refuse;
8: **fi si**
9: $i \leftarrow Q_x \, (mod \, pkey_M.n)$.
10: **si** $i = r$ **llavors**
11: accept;
12: **si no**
13: refuse;
14: **fi si**

2.2.3 Xifrat

Així com veurem en els algorismes 3 i 4 un xifrat tipus *ElGamal* acostuma a començar per uns primers passos d'un intercanvi de *Diffie-Hellman* per després efectuar una operació amb aquest secret compartit i el missatge. Després, el receptor, procedirà primer a recuperar aquest secret per desfer aquesta operació que li permet recuperar el missatge. Així en aquest apartat també té cabuda un algorisme com el 5 que sols s'explicarà des del punt de vista de l'intercanvi de claus.

ALGORISME 3 (Xifrat ElGamal)

INPUT: Clau pública $pkey_M$ i l'enter a xifrar z.
OUTPUT: Xifrat format per un parell d'enters, (a, c).

1: Generar aleatoriament una clau de sessió $k \in_R [1, (pkey_M.p) - 2]$;
2: $a \leftarrow (pkey_M.g)^k \mod p$;
3: $b \leftarrow (pkey_M.y)^k \mod p$; /* $y = g^x \mod p$; $b = (g^x)^k \mod p$ */
4: $c \leftarrow z \cdot b \mod p$;
5: Return (a, c);

2.2.3.1 ECElGamal

Així com el sistema de firma digital es pot definir sobre cossos finits amb l'algorisme DSA i hem vist amb l'algorisme 1 que es pot portar per usar-lo sobre corbes el·líptiques, el mateix passar amb tot esquema. En aquest cas el que es tradueix és l'anterior algorisme 3, ElGamal, per a corbes el·líptiques:

ALGORISME 4 (Xifrat ElGamal el·lìptic)

INPUT: Clau pública $pkey_M$ i l'enter a xifrar z.
OUTPUT: Xifrat formar per un parell de punts, (R, C).

1: Generar aleatoriament una clau de sesió $k \in_R [1, (pkey_M.n) - 1]$;
2: $R \leftarrow [k] \cdot pkey_M.G$;
3: $Q \leftarrow [k] \cdot pkey_M.P$; /* $P = [d] \cdot G$; $Q = [k] \cdot ([d] \cdot G)$ */
4: Convertir el missatge a punt de la corba el·líptica $z \rightarrow Z$;
5: $C \leftarrow Z + Q$;
6: Return (R, C);

Aquest algorisme presenta alguns inconvenients que no ens havíem trobat amb l'algorisme 1 i és que no resulta trivial convertir la informació que volem xifrar al punt, que hem denotat com a Z en el pas 5 d'aquest algorisme 4.

Sovint el que arribarà com a informació a xifrar és una clau simètrica que s'ha utilitzat en un sistema híbrid per xifrar el gruix de la informació. Aquest valor convertit a punt no podria ser partit per fer dues coordenades, primer hauria de convertir-se a un enter llarg i utilitzat com a coordenada x per calcular-ne la y seguint el definit en la secció 2.1.3, en concret a la definició 2.8. No sempre l'equació de la corba el·líptica tindrà solució en y per un valor x qualsevol, i al mateix temps s'hauria de vigilar que de tenir solució, aquesta no fos el punt el l'infinit, $\mathcal{O}_E$.

2.2.3.2 ECMQV

Tot i no ser un protocol de xifrat, sinó d'intercanvi de claus, se li fa una menció aquí és per les patents[1] a les que està subjecte aquests algorisme. Les patents de software són una lacra que ha arribat fins a la criptografia i en aquest cas la complicació de l'estudi d'aquest algorisme recau en que no es fàcil trobar-lo. El que surt d'ell en les normes i especificacions no passen de ser indicacions i recomanacions. Per tot arreu, hi ha referències a publicacions exclusives de pagament, però si hi ha una descripció clara en

[1] http://en.wikipedia.org/wiki/ECC_patents

[LMS317]. Però, l'algorisme *ECMQV* és un sistema de comunicació *online* que requereix la presència d'emisor i receptor de la comunicació al mateix temps.

Com a sistema d'intercanvi, conté una primera part amb una comunicació entre emissor i receptor, que tothom pot escoltar, i després ambdós construeixen un secret compartit constituït per alguns dels elements públics i alguns que sols sap cadascun per separat.

$$\begin{array}{c|c|c} \texttt{Alice} & \{p, a, b, G, n, h\} & \texttt{Bob} \\ pub_{Bob} = [c]\,G & & pub_{Alice} = [a]\,G \\ priv_{Alice} = a & & priv_{Bob} = c \\ sess_{Alice} = b & & sess_{Bob} = d \\ \hline & \overset{[b]G}{\longrightarrow} & [b]\,G \\ [d]\,G & \overset{[d]G}{\longleftarrow} & \\ \hline [a]\,G, [b]\,G & & [c]\,G, [d]\,G \\ [c]\,G, [d]\,G & & [a]\,G, [b]\,G \end{array} \tag{2.12}$$

Arribats a aquest punt `Alice` ja és capaç de calcular un secret compartir i també serà capaç en `Bob` amb l'informació que ell té.

ALGORISME 5 (Derivació de clau ECMQV)

INPUT: Parametres inicials de domini $\{p, a, b, G, n, h\}$ i elements públics de la comunicació $a, b, [a]\,G, [b]\,G, [c]\,G, [d]\,G$.
OUTPUT: Secret compartir Q

1: $n \leftarrow \lceil log_2\,(|\mathbb{K}|)\rceil/2$;
2: $u \leftarrow (x\,([b]\,G)\pmod{2^n}) + 2^n$;
3: $s \leftarrow b + ua \pmod{q}$;
4: $v \leftarrow (x\,([d]\,G)\pmod{2^n}) + 2^n$;
5: $Q \leftarrow [s]\,([d]\,G + [v]\,([c]\,G))$;
6: **si** Q $= \mathcal{O}_E$ **llavors**
7: goto 1;
8: **fi si**
9: Return Q

2.2.3.3 ECDH+AES256

Donat que el sistema de xifrat de l'algorisme 4 no resulta útil i d'altres estan subjectes a patents, a l'article [BM06] es pot trobar un estudi i desenvolupa-

ment d'alternatives on es proposa l'ús d'un híbrid que utilitza un intercanvi de *Diffie-Hellman* al inici però substitueix l'operació *ElGamal* dels passos 5 dels algorismes 3 i 4 per una "operació" que és una crida a un mètode de xifrat simètric i una funció de resum.

ALGORISME 6 (Xifrat ECDH+AES256)

INPUT: Clau pública $pkey_M$ i text en clar numéric z.
OUTPUT: Parell punt resultant R i xifra c.

1: Generar aleatoriament una clau de sessió $k \in_R [1, (pkey_M.n) - 1]$;
2: $R \leftarrow [k] \cdot pkey_M.G$;
3: $Q \leftarrow [k] \cdot pkey_M.P$; /* $P = [d] \cdot G$; $Q = [k] \cdot ([d] \cdot G)$ */
4: $c \leftarrow \text{aes256}(z, \text{sha256}(Q_x))$;
5: Return (R, c);

Amb aquest sistema garantim la robustesa del pas que realment xifra la informació per al seu destinatari. En cas que aquesta es veiés compromesa, sempre pot ser actualitzada a més longitud o inclús canviada si la situació de compromis del algorisme simètric resultés extrema.

2.2.3.4 EccOpenPGP

El proposat estàndard del *IEEE* descrit en la secció de normatives del punt 2.2 és proposa un algorisme molt similar al 6 que havien presentat els autors en [BM06] però que té en compte més flancs. Cal comentar que aquí s'ha usat la doble barra vertical (||) com a simbol de concatenació de cadenes o strings.

ALGORISME 7 (Xifrat ECC_OpenPGP)

INPUT:: Clau pública $pkey_M$ i text en clar numéric z.
OUTPUT: Parell punt resultant R i xifra c.

1: Generar aleatoriament una clau de sessió $k \in_R [1, (pkey_M.n) - 1]$;
2: $R \leftarrow [k] \cdot pkey_M.G$
3: $S \leftarrow [k] \cdot pkey_M.P$; /* $S = [k] \cdot ([d] \cdot G) = [d] \cdot ([k] \cdot G) = [d] \cdot K$ */
4: Param $\leftarrow$ curveID||pubkeyID||01||KDF_hashID||
aesID||"AnonymousSender"||recipient_fingerprint;
5: $Z \leftarrow KDF\,(S, len\,(aesID)\,, Param)$;
6: $c \leftarrow$ AESkeyWrap (Z, m);
7: Return (R, c);

Els primers passos d'aquest algorisme consisteixen en un intercanvi *Diffie-Hellman* amb una preparació dels paràmetres per la crida al mètode *KDF*. Aquest mètode *KDF* produeix una clau de xifrat i no sols un *hash* com en el proposat en l'anterior algorisme 6. Amb aquest mètode ens protegim d'atacs de resposta escollida, poc aplicables en sistemes com el correu electrònic però més sensibles els sistemes d'intercanvi de claus. A més el mètode *AESKeyWrap* descrit en [rfc3394] ofereix protecció a l'integritat que el criptosistema *aes* per si sol no proporciona.

El punt més important que presenta com a millora, és el d'utilitzar les diferents longituds que pot admetre aquest algorisme segons la longitud de la corba el·líptica amb la que s'ha fet la clau.

Hi ha referida una operació en el pas 5 de l'algorisme 7 que trobem descrita a continuació:

ALGORISME 8 (Key Derivation Function)

INPUT:: Punt S, longitud de l'output *obits* i cadena $Param$.
OUTPUT: cadena de bits.

1: cntr $\leftarrow$ 1;
2: threshld $\leftarrow (obits + hbits - 1)/hbits)$;
3: **repetir**
4: C32 $\leftarrow$ (unit32) big_endian (cntr);
5: HB $\leftarrow$ hash $(S_x||C32||Param)$;
6: MB $\leftarrow$ MB||HB;
7: **fins** cntr $\leq$ threshold;
8: return leftmost obits of MB

El seu origen prové de la norma [NIST800] que proporciona mètodes generics per a cada cas d'implementació de l'embolcall per al xifrat *AESKeyWrapper*.

2.2.3.4.1 Desxifrat Els anteriors sistemes comentats per al xifrat tenen la seva implementació en el desxifrat en alguna de les versions del pegat per a corba el·líptica *eccGnuPG* però aquest sistema proposat com a estàndard per a l'*OpenPGP* encara no està completament implementat. És important deixar constància de com ha de ser el algorisme de desxifrat ja que serà el que es veurà implementat quan es faci públic per la *libgcrypt*.

ALGORISME 9 (Desxifrat ECC_OpenPGP)

INPUT:: Clau privada $skey_M$ i parell d'elements xifra (R, c).
OUTPUT: Text pla z.

1: $S \leftarrow [skey_M.d] \cdot R$;
2: Param $\leftarrow$ curveID||pubkeyID||01||KDF_hashID||
aesID||"AnonymousSender"||recipient_fingerprint;
3: $Z \leftarrow KDF\left(S, len\left(aesID\right), Param\right)$;
4: $m \leftarrow \text{AESkeyWrap}^{-1}\left(Z, \text{c}\right)$;
5: Return m;

És un algorisme senzill, similar al que es pot trobar escrit en [BM06] per l'anterior algorisme 6 per invertir l'operació de xifrat. No està clara però, la necessitat que imposa aquest estàndard amb l'ús l'estructura de concatenació *Param* que s'usa de la mateixa manera al xifrar que al desxifrar, així com la necessitat de usar el mètode *KDF* com a una forma de *sobre-hash*.

Al igual que amb el mètode embolcallat per al xifrat *aes* amb diferents longituds, la normativa [rfc3394] també especifica el procediment per realitzar l'operació inversa de desxifrar.

2.3 Criptoanàlisi el·líptic

Quan es planteja un criptosistema també es planteja com trencar-lo. La criptologia és el resultat de la pugna entre criptografia i criptoanàlisi. Molt bones idees són ideades per conseguir criptosistemes segurs, però idees molt bones i sovint partint de llocs que semblen descabellats alimenten el criptoanàlisi.

Des de la criptografia s'han de cobrir diversos flancs ja que tota debilitat és bona per fer metlla. Primer hi ha els atacs directes als algorismes que

busquen noves i creatives maneres de jugar amb la matemàtica per fer més fàcil allò que s'havia suposat molt difícil. Això és el que anomenem atacs directes i són els que es descriuen en la següent secció.

També hi ha però formes que no ataquen l'algorisme propiament, sinò el seu entorn i la implementació, són el que anomenem atacs laterals. També tenen per objectiu trencar d'una forma ill·lícita el secret guardat, però ho fan escoltant l'entorn on s'executa l'operació criptogràfica. A aquests atacs laterals els dedicarem al capítol 3.

2.3.1 Atacs directes

Amb els atacs directes es posa a proba les característiques del criptosistema a nivell matemàtic. Es tracta de posar en posició d'escac un algorisme. Depenent de la força del atac el criptosistema podrà sobreviure amb modificacions per evitar les situacions dèbils o pot arribar a comprometre'l sencer i que porti a l'abandó. En criptologia, la lluita entre criptografia i criptoanàlisi serà eterna, i això és bo. Tot i això no és fàcil constituir algun atac del segon tipus que s'ha anomenat abans. Sovint amb l'indici que pot existir un algorisme d'aquest tipus, la confiança amb el criptosistema decau ràpidament.

Les corbes el·líptiques porten un bon grapat d'anys aguantant les envestides dels criptoanalistes, i per sort al que hem arribat a un reforçament del criptosistema molt bo. Això no és garantia absoluta. El problema a resoldre per part d'un criptoanàlisi s'ha definit en 2.14 i podem veure-ho de forma particular com el punt d'ordre n com el generador G del subgrup cíclic sobre el que s'ha fet criptografia i P la clau pública, de forma que obtenir la clau privada d passaria per resoldre el logaritme discret el·líptic

$$P = [d]G. \tag{2.13}$$

A continuació recordem breument, per ordre de complexitat decreixent, els atacs clàssics al logaritme discret, alguns dels quals son aplicables a la criptografia de corba el·líptica y altres, com l'*Index Calculus* no ho son.

2.3.1.1 Força bruta

Els primer dels atacs per resoldre un logaritme discret el·líptic és calcular un a un els valor del subgrup cíclic definit en 2.12 ja que el valor generador del subgrup es públic (G) i resoldre l'equació 2.13 respont a l'algorisme 10.

ALGORISME 10 (Força bruta)

INPUT:: Una clau pública el·líptica.
OUTPUT: Valor privat d.

1: $m \leftarrow 1$
2: **bucle**
3: $P' = [m]G$;
4: **si** $P' = P$ **llavors**
5: Return m
6: **si no**
7: $m \leftarrow m + 1$;
8: **fi si**
9: **fi bucle**

Tot i la aparent senzillesa de l'algorisme es pràcticament infinit degut a l'ordre n del subgrup cíclic generat per G. El cost d'aquest algorisme és de l'ordre de $O(n)$, impracticable quan la longitud del valor cercat és superior a 80, i encara menys les implementacions actuals comencen amb 192 o inclús en 256.

2.3.1.2 Baby Step / Giant Step

Una altra forma d'atacar les corbes el·líptiques, que no es nova i constitueix una conversió de una també algorisme per a cossos finits és l'anomenat *pas de nen, pas de gegant*. Intenta buscar per dues vies (una de més senzilla, el pas de nen; i una de més complexa, el pas de gegant) un punt coincident que permeti donar el resultat del logaritme discret el·líptic. Donats dos punts $P, Q \in G$, on G és un subgrup cíclic com el definit en 2.12; aquest dos punts estan relacionats segons

$$Q = [m]\, P \tag{2.14}$$

Aquest valor m, per la divisió euclídea el podem escriure com

$$m = \lceil \sqrt{n} \rceil a + b,$$

on $0 \leq a, b < \lceil \sqrt{n} \rceil$ no els coneixem, però si n. Podem reescriure l'equació 2.14 com

$$\begin{aligned} (Q - [b]\, P) &= [a]\, ([\lceil \sqrt{n} \rceil]\, P)\,, \\ R_b &= Q - [b]\, P, \\ S_a &= [a]\, ([\lceil \sqrt{n} \rceil]\, P)\,. \end{aligned}$$

On anomenem *baby step* a la cerca de R_b i *giant step* a la cerca de S_a i el mètode té una complexitat $O(\sqrt{n})$.

baby step	
b	$R_b = Q - [b]\,P$
0	$Q - [0]\,P$
1	$Q - [1]\,P$
$\vdots$	$\vdots$
b'	$R_{b'} = Q - [b']\,P$

giant step	
a	$S_a = [a]\,([\lceil\sqrt{n}\rceil]\,P)$
0	$[0]\,([\lceil\sqrt{n}\rceil]\,P)$
1	$[1]\,([\lceil\sqrt{n}\rceil]\,P)$
$\vdots$	$\vdots$
a'	$S_{a'} = [a']\,([\lceil\sqrt{n}\rceil]\,P)$

I l'evolució d'aquest dos passos porta fins dos valors de a' i b' que fan $R_{b'} = S_{a'}$ que significa

$$m_0 \equiv a'\lceil\sqrt{n}\rceil + b' \pmod{n}.$$

El cost d'aquest algorisme és de l'ordre de $O\left(\sqrt[4]{n}\right)$ tant en memòria com computacionalment.

2.3.1.3 Rho (ρ) de Pollard

Un algorisme força ben explicat en articles com [H9DL] i [LMS317]. Es tracta de conseguir realitzar un cícle en el subgrup cíclic per resoldre el logaritme discret el·líptic. Donats dos punts $P, Q \in G$, on G és un subgrup cíclic de punts d'una corba el·líptica $E/\mathbb{F}_p$, i l'algorisme *Rho de Pollard* busca subcícles per resoldre un logaritme discret el·líptic 2.14 utilitzant dos punts aleatòris inicials obtinguts segons

$$\begin{aligned} X_0 &= [x_0]\,P + [x'_0]\,Q, \\ Y_0 &= [y_0]\,P + [y'_0]\,Q. \end{aligned}$$

Es recorrerà el subgrup cíclic segons:

$$\begin{aligned} X_k &= [x_k]\,P + [x'_k]\,Q, \\ Y_k &= [y_k]\,P + [y'_k]\,Q. \end{aligned}$$

I després de unes $O\left(\sqrt{\frac{n\pi}{2}}\right)$ iteracions els camins es creuaran segons

$$\left.\begin{aligned} [x_k]\,P + [x'_k]\,Q = X_k = Y_l = [y_l]\,P + [y'_l]\,Q, \\ [x_k - y_l]\,P = [y'_l - x'_k]\,Q = [y'_l - x'_k]\,[m]\,P, \end{aligned}\right\} \Longrightarrow m = \frac{x_k - y_l}{y'_l - x'_k}.$$

Tot i que és un molt bon atac a les corbes el·líptiques segueix tenint un cost molt elevat. Com el *BSGS* és computacionalment de l'ordre de $O\left(\sqrt[4]{n}\right)$ tot i que no necessita emmagatzemar tanta memòria.

2.3.1.4 Index Calculus i Xedni Calculus

El atac més efectiu per logaritme discret sobre grups multiplicatius de cossos finits s'anomena *Index Calculus*. Tot i tenir un cost elevat, aquest cost és subexponencial, que sense ser polinòmic ($O\left(log^{\alpha}n\right)$) és millor que un exponencial ($O\left(n^{\alpha}\right)$).

Una gran avantatge que ens dona la criptografia de corba el·líptica es justament que no es coneix cap algorisme de cost subexponencial com el *index calculus*. Es més, les aplicacións d'aquest atac per a les corbes el·líptiques acaben essent inclús més costoses que la força bruta i qualsevol atac ha de superar aquest cost per poder considerar-se un atac.

Hi ha però corbes el·líptiques sobre les que si que es pot aplicar i per això existeixen restricción a l'hora d'elegir una corba el·líptica criptogràficament bona. Es per aquest atac que una corba el·líptica no pot ser supersingular (quan el cardinal de la corba $n = |E(\mathbb{F}_p)|$ es de la forma $n = p - 1$) ja que el logaritme discret el·líptic es pot reduir a un logaritme discret sobre el extensió k del cos base de la corba $\mathbb{F}_p$, on k no es un valor excessivament gran com per que un *Index Calculus* sigui factible sobre $\mathbb{F}_{p^k}$. Pitjor es el cas d'una corba anòmala (amb $n = p$) on el logaritme discret el·líptic es veurà reduït a un logaritme discret sobre $\mathbb{F}_{p^2}$.

Totes aquestes situacions son fàcilment evitables a l'hora de elegir la corba.

Hi ha una publicació, [Silv99], anunciant l'existència d'un algoritme subexponencial basat en una forma de invertir l'*index calculus* que *Silverman* va donar a anomenar *Xedni calculus* invertint fins i tot la paraula per donar-li nom. Aquesta publicació va causar gran rebombori en el mon de la criptografia fins al punt que alguns catastrofistes les donaven per mortes.

Al poc temps de la publicació de l'article enunciant, conjuntament *Silverman*, *Koblitz* i *Jacobson* van publicar [Xedni99] explicant els defectes que contenia el procediment de l'atac *Xedni* i exposant que no existia tal atac. Després de la crispació inicial, va arribar la calma i la confiança amb les corbes el·líptiques va augmentar.

2.3.1.5 Atacs per deficiències en la ECC

Els anterior atacs son generals, es a dir, aplicables a qualsevol corba, en canvi els que presentem a continuació es poden dur a terme solsament sí el ECC ha estat "mal definit", en el sentit de que alguns parametres del mateix tenen característiques que els fan susceptibles a aquest atacs.

2.3.1.5.1 Pohlig-Hellman Un altre atac que és dona sota un cas particular de l'ordre del subgrup cíclic de punts de la corba el·líptica $E/\mathbb{F}_p$ sobre el que es planteja el logaritme discret el·líptic s'anomena *reducció de Pohlig-Hellman*. Consisteix en factoritzar l'ordre del subgrup i així procedir a reduïr el problema al cos finit $\mathbb{F}_p$. Com el cos finit primer sobre el que es defineix la corba el·líptica té una longitud molt menor, de poder-se fer aquesta reducció es podria atacar aquest $\mathbb{F}_p$ resultant amb algorismes ràpits sobre cossos finits sobre un cos especialment petit.

Donat un subgrup cíclic $\langle G \rangle$ de punts d'una corba el·líptica $E/\mathbb{F}_p$ amb un ordre

$$n = \text{ord}(G) = \prod_{i=1}^{k} {p_i}^{e_i},$$

la reducció de Pohlig-Hellman consisteix en dos passos:

1. Reduïr el subgrup cíclic a $\mathbb{F}_{{p_i}^{e_i}}$
2. Reduïr $\mathbb{F}_{{p_i}^{e_i}}$ a $\mathbb{F}_{p_i}$

Des d'on es pot atacar el cos $\mathbb{F}_{p_i}$ amb qualsevol criptoanàlisi eficient sobre cossos finits.

Aquest atac és pot evitar al agafar un ordre del subgrup cíclic que sigui primer. Segons les normes comentades en la secció 2.2.1 les corbes el·líptiques que es proposen com estàndards tenen un ordre del subgrup cíclic primer a més d'aconsellar que si el cardinal de la corba el·líptica no fos absolutament primer, sols tingui un factor 2 o 4. Aquest es el que s'ha donat a anomenar *cofactor* i la norma [P1363] be a denotar com h.

En cas que s'utilitzi un grup de punts de la corba el·líptica d'un ordre compost, que seria una negligència que permetria aplicar aquest algorisme, el cost d'aquest atac seria de l'ordre de $O\left(\sqrt{p_i}\right)$ on p_i és el mes gran dels factor amb que l'ordre n descomposa.

2.3.1.5.2 Condició MOV L'atac que porta per nom l'acrònim *MOV* es deu a tres investigadors: *Menezes*, *Okamoto* i *Vanstone*. No afecta a totes les corbes el·líptiques en general sinó a un subconjunt de corbes que mantenen una característica comú que ens permet reduïr un logaritme discret el·líptic, $E(\mathbb{F}_p)$, a una extensió d'un cos finit de la mateixa característica que el definit per la corba, $\mathbb{F}^*_{p^m}$.

Definició 2.16. Suposem un corba el·líptica $E(\mathbb{F}_p)$ i un punt P d'aquesta corba d'ordre n, per a un llindar b si el conjunt $p, p^2, p^3, \ldots, p^b$ no es congruent amb 1 mòdul n el problema del logaritme discret el·líptic es susceptible de ser reduït a un logaritme discret sobre un $\mathbb{F}^*_p$-vector.

En l'implementació del mètode de validació d'una corba el·líptica en el software lliure *Sage* s'agafa un valor $b = 30$ i resulta suficient i eficient.

2.3.2 Altres formes d'atac

Existeix una forma creativa d'atacar els problemes criptològics i es tracta de no atacar la matemàtica subjacent, sinó atacar la implementació el seu suport físic. Es tracta d'escoltar canals, espais de memòria en sistemes multiusuari, de buscar patrons en el soroll, en el consum elèctric o qualsevol detall que amb proutes mostres ens pugui portar a descobrir el preuat secret. En el fons s'ataca l'implementació i no el criptosistema propiament. Per sort, no sols és busca com atacar sinó també és busquen contramesures a aquests atacs.

Capítol 3

Atacs laterals

En sistemes de temps compartit, on hi poden estar diversos usuaris alternant les seves instruccions en el processador, on hi ha multiples fils d'execució, és comparteix la memòria cau i sense la possibilitat de una protecció per part del sistema operatiu en l'accès a aquesta memòria com passa amb la memòria ram, podem trobar-nos amb molestos companys de viatge que mirin que estem fent. Els sistemes monousuari (com podem veure les targetes intel·lgents o dispositius empotrats) tampoc queden alliberats, doncs es pot escoltar el seu consum elèctric durant la execució d'una operació per tal de deduir-ne informació. Sols es una qüestió del repertori d'exemples que en puguem arribar a aconseguir.

El que dona molta importància a aquests tipus d'atacs és el seu preu. Mentre que atacar un criptosistema, que pugui ser més o menys dèbil, des d'un punt de vista matemàtic pot costar molts diners en hardware, que molts cops és especialitzat, els instruments per perpetrar un atac lateral com els de canal sols costa del ordre de centenars de dòlars. Unes xifrar a l'abast de qualsevol organització.

3.1 Atacs actius

Extreure informació de la clau secreta per la via de mesurar el temps amb que el criptosistema ens dona la resposta, resulta viable si podem escollir el missatge. Si la nostra caixa negra, que computa un algorisme matemàticament robust i públic, ens xifra mitjançant una clau secreta els missatges que nosaltres li preparem i a més li cronometrem el temps de resposta, sols serà una qüestió de conseguir suficients exemples per extrapolar-ne informació.

En l'article [HAGAI] hi ha una descripció introductória a aquest atac per a sistemes criptogràfics basats en cossos finits com ElGamal o RSA. Per al

nostre cas, el plantejament és sobre un sistema basat en corba el·líptica. La operació que podem atacar és el producte d'un punt per un escalar $R = [x]\,P$ en $E(\mathbb{F}_p)$, on l'atacant coneix el setup del ECC que hem descrit en la tupla 2.15 i pot escollir quin punt P xifrar, mentre que x és la clau secreta.

3.2 Atacs passius

Extreure informació del consum elèctric d'una targeta intel·ligent que conté una clau secreta desconeguda per nosaltres resulta molt senzill i, en especial, barat. Simplement ens cal una petita resistència a l'entrada de corrent o en la presa de terra per observar les diferències en el voltatge. L'equip electrònic per a les freqüències a les que es treballa és pot trobar pràcticament en qualsevol laboratori d'electrònica i resulten ser proporcionalment barats.

Amb una visió global del consum, en una gràfica i amb un sol exemple, inclús un ull poc entrenat pot veure patrons com el moment en que es van executant les diferents rondes d'un sistema DES, o el procès de multiplicar un punt per un escalar que requereix recorre bit a bit el escalar i fer una operació o una altra segons si és 0 o 1, o be qualsevol altra operación que tingui algun pas iteratiu. En [KJJ] podem llegir-ne una descripció de com dur a terme un atac d'aquest tipus.

Davant de contramesures a aquest tipus d'atac, aquest s'han sofisticat inclús més que el descrit fins ara, doncs simplement em analitzat el consum de forma plana. Si el protegim amb l'introducció d'errors o soroll o fluctuacions en la freqüència de rellotge, sols incrementem el nombre de exemples que un atacant necessita per fer un anàlisi diferencial d'aquestes captures. L'esperança per la protecció l'haurem de posar en provocar-li que el nombre de exemples a obtenir per extreure alguna informació sigui tant elevat que se li faci impracticable l'atac.

3.3 Proteccions i contramesures

Com després s'explicarà amb més detall en la secció 5.1, aquest projecte de corbes el·líptiques ha transcendit i ha entrat a formar part d'un software major: el *GnuPG*. Això implica ampliar la conciència de responsabilitat i veure més entorns de treball. S'ha d'estar informat de l'evolució d'atacs com els descrits en la secció 2.3.1, els directes contra els algorismes; però també s'ha d'estar a sobre dels atacs descrits en aquest capítol i implementar les contramesures que es puguin aplicar. També quan s'escriu quelcom nou, s'ha de pensar amb no caure en debilitats conegudes.

Els entorns sobre els que el *GnuPG* s'executa pels usuaris és molt variat. Des de petits PCs personals on la integritat del software i la seva execució pot ser molt tranquil·la, però més s'ha de pensar en altres entorns que resulten més hostils. De manera que a l'hora d'implementar s'ha de tenir en compte que pot executar-se sobre un entorn multiusuari i podem tenir alguna debilitat per llegir de la cau, o pot executar-se sobre un entorn d'un sistema empotrat o una targeta intel·ligent de manera que el consum elèctric entre operacions podria estar comprom és.

Cadascuna de les evolucions que ha fet la implementació de corbes el·líptiques ha tingut en compte aquests factors.

3.3.1 La Protecció del sistema DH+AES

En l'article [BJN] es planteja una protecció a un esquema tipus ElGamal on el producte *no* és modular utilitzant una operació alternativa. Tot i que l'esquema clàssic ElGamal realitza aquesta operació de forma modular, la curta longitud de les corbes el·líptiques i que es la seva virtud, és converteix en un problema si s'intenta fer un producte modular en $\mathbb{F}_p$ i la p resulta petita.

Un cop constatada la debilitat que suposa utilitzar un algorisme ElGamal el·líptic híbrid (veure l'algorisme 4 és va treballar en una alternativa que utilitza un intercanvi de claus tipus Diffie-Hellman seguint el citat article i les reflexions d'en *Mikael Milnikov* per usar un sistema simètric com és l'AES, com s'ha descrit en l'algorisme 6. Aquest col·laborador rús va aportat un codi que utilitzava l'algorisme simètric AES per suplir la operació ElGamal amb cossos finits.

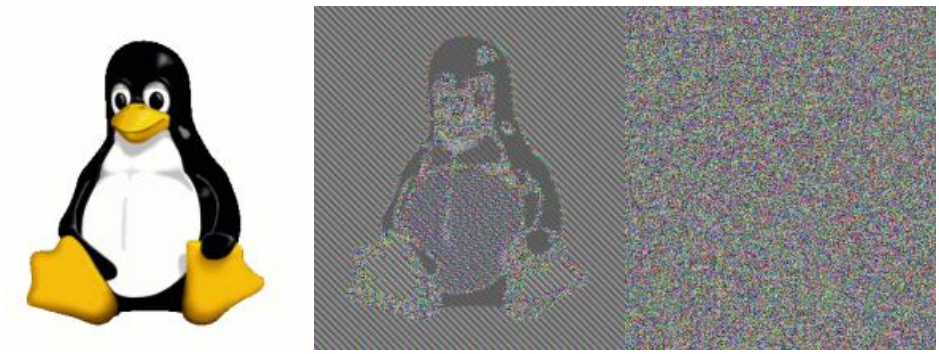
Utilitzem el secret compartit com a clau en el xifrat AES pel missatge. Necessitem que aquesta clau *sempre* sigui *màxima* i a la vegada reproductible per part del receptor, i per això també es fa us d'una funció de resum SHA-2 de la mateixa longitud que l'AES.

3.3.2 El sistema de xifrat de blocks CFB

Amb l'algorisme 6, proposat per solucionar el problema que suposa descriure l'esquema d'ElGamal el·líptic (veure també l'algorisme 4) s'ha de tenir en compte un detall d'implementació ja que es proposa d'utilitzar una longitud fixa per al xifrat AES, concretament 256 bits.

Si utilitzem blocs per xifrar allò que és major d'aquests 256 bits correm el risc de deixar entreveure una estructura. No sembla tant creu quan la longitud sols permetrà tenir un o dos bloc, però s'ha de deixar a l'atzar.

Per veure, i gràficament, un exemple de com afecta a la xifra final s'ha seguit l'explicació que hi ha en l'article de la Viquipèdia sota el títol "*Block cipher modes of operation*". Donada una figura inicial com la de la figura 3.3.1a li podem aplicar un xifrat molt robust però bloc a bloc (que s'anomena *Electronic CodeBook, ECB*), per separat i independent l'un de un altre. El resultat d'aquest xifrat no emmascarat completament com es pot comprobar en la figura 3.3.1b. Si es relaciona el text pla o la xifra amb els veïns surgeixen diferents modes, com el *Cipher FeedBack, CFB* i la imatge resultant del xifrat 3.3.1c no mostra cap patrò que la relacioni amb l'original.



(a) Original (b) Model ECB (c) Algun dels modes segurs

Figura 3.3.1: Exemple de com pot ser d'important el mode de xifrat ell·legit.

Aquest sistema concatena el procés de xifrar d'un bloc amb el següent de manera que s'afegeix dispersió i augmenta la entropia a la xifra final. També extret de la *Viquipèdia* tenim en les figures 3.3.2 i 3.3.3 amb el diagrama de xifrat i el de desxifrat respectivament. En aquest mode el que es fa es xifrar un vector inicial que s'emmascara amb una operació *xor* amb el text pla, per després reutilitzar la xifra com a següent entrada del pas següent. Per al desxifrat el procediment es molt similar, però ara la xifra s'utilita per desemmascarar el valor de l'operació *xor* i per realimentar els posteriors passos.

3.3.3 Protecció: coordenades projectives *vs* coordenades afins

En primera instància, l'ús de coordenades projectives era una el·lecció matemàtica, peró resulta una gran eina per protegir una implementació a atacs laterals passius com els descrits en [KJJ]. Es tracta d'introduir soroll a un

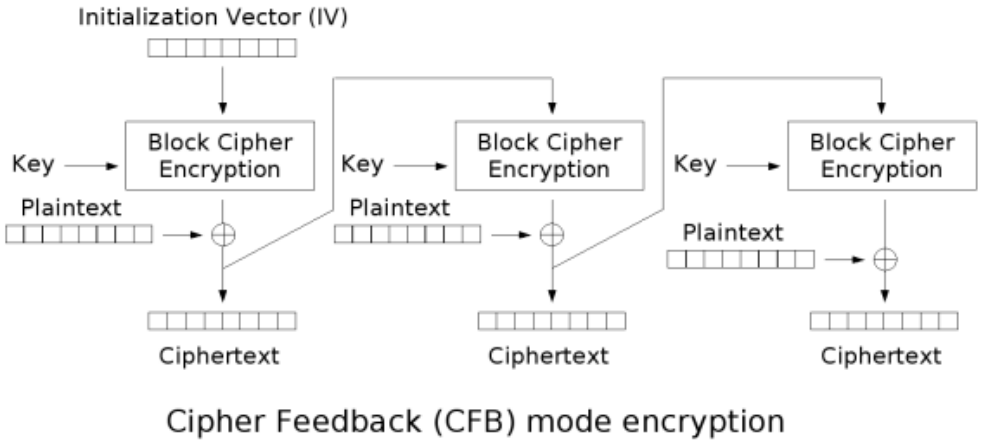


Figura 3.3.2: Diagrama de xifrat de blocs amb el sistema CFB, extret de la *Viquipèdia*.

possible atacant que pugui estar escoltant l'electrònica amb que estem treballant.

Com s'ha explicat a la secció 2.1.2, concretament amb la definició 2.5, un pla projectiu bidimensional és representa amb una terna de valors. Un punt d'un pla afí és representat per una recta en un pla projectiu i tots els punts d'aquesta recta formen una classe d'equivalència que es la representació d'un sol punt. Recordem l'equació:

$$(x, y, z) \sim (x\prime, y\prime, z\prime) \Rightarrow \exists \lambda \in \mathbb{F}^{\star} : x = \lambda x\prime, y = \lambda y\prime, z = \lambda z\prime \tag{3.1}$$

Una operació de *producte per un escalar* és realitza usant un algorisme molt eficient que recorre bit a bit l'escalar i realitza operacions *suma de punts* i/o de *duplicat d'un punt* segons si el bit del pas del bucle es un 0 o un 1. Aquestes dues operacions tenen costos diferents i per un atacant que n'analitzi el consum elèctric pot veure patrons de consum. El moment de realitzar l'operació es pot aïllar i sobreposant algunes mostres es pot extreure informació de quin és el secret guardat sota un logaritme discret el·líptic tipus $P = [d]\, G$ com l'enunciat en 2.14.

Com tenim una representació multiple d'un punt a través de qualsevol valor λ aleatori amb el que emmascarem el punt, tot i que el valor del recorregut de bits sigui el mateix (l'escalar no ha variat), el nombre de mostres necessàries per un atac d'anàlisi lineal augmenta.

En el capítol 5 és descriu amb més detall l'estat actual de la implementació d'aquesta protecció.

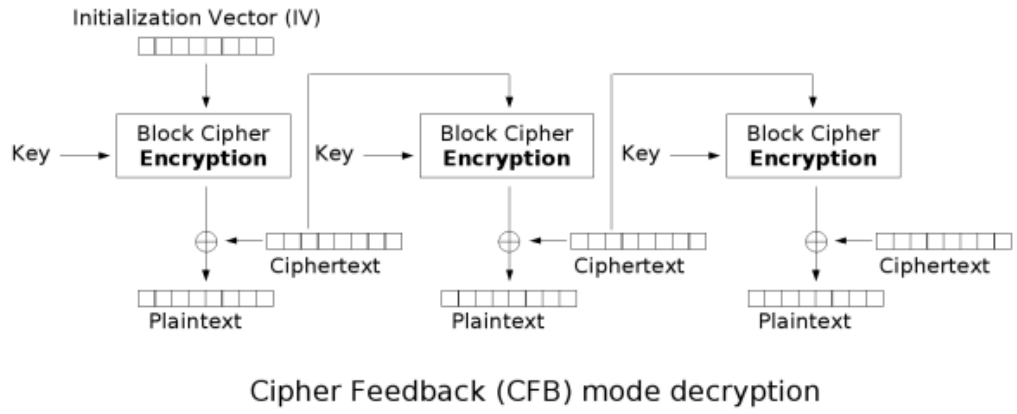


Figura 3.3.3: Diagrama de desxifrat de blocs amb el sistema CFB, extret de la *Viquipèdia*.

3.3.4 La protecció per isomorfismes del subgrup $\langle G \rangle$

Tot i que encara no em definit què és un isomorfisme, que és farà en la secció 4.1.1, concretament amb la definició 4.1. De moment, sols cal tenir present el seu significat semàntic dient que un isomorfisme d'un grup, és un altre grup diferent però que manté la mateixa "forma".

La protecció indicada en 3.3.3 a n l'utilitzar coordenades projectives resulta útil per donar un cert emmascarament, però no fa variar el valor escalar secret amb el que s'opera. Es poden aprofitar isomorfismes de grups per variar el generador del subgrup cíclic que no aporten una protecció matemàtica a l'algorisme però si a la firma el·lèctrica de les operacions. Així es pot canviar de generador

$$\langle G \rangle \equiv \langle G' \rangle \Longrightarrow \exists k \in \mathbb{F}_p^*, \text{ tal que } G = [k]G'.$$

D'aquesta manera tenim una igualtat

$$P = [d]\, G = [d]\, [k]\, G',$$

que en permet protegir l'implementació del ECC d'avant d'anàlisi diferencials de l'empremta elèctrica descrits en [KJJ].

També en aquest cas, aquesta protecció ja te una implementació sobre el mòdul escrit per la rama 1.4 del *GnuPG*.

Capítol 4

Reseteig d'un criptosistema: Estrelles d'isogènies

La criptografia de corba el·líptica presenta diverses avantatges respecte la criptografia sobre cossos finits. La longitud del cos sobre el que es treballa pot ser molt menor, a dia d'avui podem confrontat una clau *ElGamal* de 2048 bits amb una clau ell·líptica de 224 bits. Tot i que s'utilitzi una estructura de clau pública com la especificada en la norma [ECPGP], que pot fer que la clau rondi els 500 bits, segueix essent molt menor. També presenta avantatges de còmput ja que les operacions per un usuari lícit tenen menys cost. Però hi ha una avantatge molt major de les corbes el·líptiques que no s'aprofita: la seva facilitat a canviar de corba per resetejar el criptosistema i invalidar un atac.

Les dues primeres avantatges resulten molt bones per sistemes limitats. Ja sigui per la capacitat de comput, disponibilitat de memòria o ample de banda, les corbes el·líptiques són molt útils. Però els estàndards suggereixen l'us d'un nombre molt limitat de corbes, de manera que un atac sobre una pot afectar a totes les claus fetes amb ella. La solució d'elegir-ne una altra corba el·líptica major **no** és bona per aquests sistemes més limitats, ja que acostumen a estar molt dedicats a una longitud degut a la seva limitació. De tota manera, per als sistema que no sofreixen limitacions estrictes d'aquests tipus tampoc es bo ja que s'hi perd una gran agilitat.

Una solució passa per que en la generació del parell de claus *pública-privada* sigui generada una corba el·líptica nova, aleatòria i que, molt important, compleixi una bona llista de requisits, com per exemple que s'evitin situacions susceptibles de ser atacades de les formes explicades com atacs directes en la secció 2.3.1. Es una solució excessivament costosa per a l'usuari final.

Una segona aproximació passa per proporcionar una variabilitat major de

corbes en cada longitud que, tot i limitar les longituds elegibles si aportaria diferents nivells de protecció en base a les necessitats.

Però, i si es pogues oferir que cada usuari, en el moment de generar el seu parell de claus, aconseguis una corba el·líptica pròpia amb un cost de producció raonable? Aquí vam veure que les estrelles d'isogènies poden tenir joc. Cap la possibilitat que utilitzan un criptosistema basat en estrelles d'isogènies descrit en [RS06] un proposit diferent al inicialment plantejat pels autors, i és planteja aquí per al reseteig d'un criptosistema.

4.1 Isomorfismes i isogènies

Com passa amb la critografia amb cos finit en front l'amenaça l'un criptoanalista canviar la clau secreta utilitzant la mateixa longitud del grup no invalida l'atac. Cal per això canviar de corba que sovint implica augmentar la longitud a una de més bits. De ser així, una gran virtud de les corbes el·líptiques és perd en no res.

Cal poder iniciar, i també reiniciar, el ECC amb una corba el·líptica nova de manera que tot l'atac realitzat contra una corba no pugui ser portat a la nova que s'ha elegit. [RS06] i [isoTFC] són un article i un treball final de carrera on es planteja un criptosistema basat en estrelles d'isogènies. No coincideix amb el nostre propòsit ja que en el citat article els elements amb els que es descriu l'esquema criptogràfic són les corbes el·líptiques dins d'una estructura d'anells on els nodes representen corba el·líptica i les arestes transicions d'isogènia.

En canvi el que busquem nosaltres és fer un recorregut per aquesta estructura d'anell per conseguir una nova corba el·líptica criptogràficament útil. Principalment que, per a un atacant, portar aquest criptoanàlisi de la corba inicial a la nova sigui tant costos, i si pot ser més, que tornar a començar el criptoanàlisi sobre el nou subgrup cíclic.

L'escull principal d'aquest algorisme de reseteig és el propòsit de ser molt més eficient que crear una corba el·líptica criptogràficament útil des de zero. És més, la restricció ha de ser que el temps de creació d'un parell de claus el·líptiques ha de ser un temps raonable per a que un usuari ho reprodueixi en el seu computador; inclús en un de limitat com un empotrat o una targeta intel·ligent.

Hem estat dient que el ECC que es defineix ara és absolutament estàtic. Tot el procés de generar la tupla 2.15 i el seus valors ens arriba recomanat pels valor indicats en l'annèx E de l'article [NIST186]. També podem utilitzar altres corbes el·líptiques, com les proposades en [brainpool] o les definides en [sec2]. Peró totes tenen en comú que l'únic que difereix, al usar-les, entre dues

claus és el valor privat d; compartint la corba el·líptica també es comparteix el criptoanàlisi.

El reseteig de la corba el·líptica és un procés que haurà de ser possible de realitzar tant a la generació estàndard d'una clau com en qualsevol moment de la vida de la mateix. Amb claus sobre cossos finits la validesa d'una clau sempre està en funció de la seva longitud i de tant en tant es modifica el valor de la caducitat. Amb corba el·líptica podem ser capaços de generar una clau a l'inici que resulta diferent a la de la resta, i després d'un temps, on caducaria i la renovariem, procediríem a fer un reseteig, de manera que tindríem una nova corba amb una nova clau equivalent i tot criptoanàlisi quedaria obsolet.

4.1.1 Què és un isomorfisme?

Abans de definir una *isogènia* i l'estructura proposada en [RS06] em de començar per definir que és un isomorfisme. Un isomorfisme no aporta una major seguretat en front dels atacs directes als algorismes, peró hem vist que la seva utilitat amb els atacs laterals 2.3.2.

En general si $\mathcal{A}$ i $\mathcal{B}$ son estructures algebràiques del mateix tipus i $f : \mathcal{A} \to \mathcal{B}$ és una aplicació entre elles, es diu que f és un *morfisme* entre $\mathcal{A}$ i $\mathcal{B}$, quan l'aplicació "respecta" les operacions, es a dir, quan l'imatge per f d'una operació en $\mathcal{A}$ és el mateix que operar en $\mathcal{B}$ les imatges dels operands. Si $\mathcal{A} = \mathcal{B}$ a f se l'anomena un *endomorfisme*. Quan f és bijectiva, es parla d'*isomorfisme*. Per concretar vegem què és un morfisme de grups. Siguin $(\mathcal{G}, \circ)$ i $(\mathcal{G}', *)$ dos grups i $f : \mathcal{G} \to \mathcal{G}'$ una aplicació entre ells. Direm que f és un morfisme entre aquests dos grups quan per qualsevol element $a, b \in \mathcal{G}$ sigui $f(a \circ b) = f(a) * f(b)$. Formalment podem donar la definició següent.

Definició 4.1. Donats dos grups $(\mathcal{G}, \circ)$ i $(\mathcal{G}', *)$ direm que son isomorfs si existeix una aplicació bijectiva f entre $\mathcal{G}$ i $\mathcal{G}'$, tal que

$$\begin{array}{rcl} f : \mathcal{G} & \longrightarrow & \mathcal{G}', \\ f(a \circ b) & \longmapsto & f(a) * f(b), \end{array}$$

es a dir, f és un morfisme bijectiu entre ambdos grups.

Usarem més endavant el concepte de *núcli* d'uns morfismes entre corbes el·líptiques anomenats *isogènies*. Definim ara l'idea de núcli d'un morfisme de grups.

Definició 4.2. Si $\mathcal{G} \xrightarrow{f} \mathcal{G}'$ és un morfisme entre grups, el núcli (o *kernel*) d'aquest morfisme és el conjunt d'elements del primer grup $\mathcal{G}$ que s'apliquen

en l'element neutre del segon $\mathcal{G}'$, es a dir,

$$\ker f = \{a \in \mathcal{G} : \ f(a) = e'\}, \text{ essent } e' \text{ el neutre de } \mathcal{G}'.$$

Es demostra fàcilment que sí f és un morfisme de grups el seu núcli $\ker f$ és un subgrup del primer d'ells,

En el cas del subgrup cíclic usat per definir el logaritme discret el·líptic de la definició 2.12, tenim un punt $G \in E(\mathbb{F}_p)$ de $\text{ord}(G) = n$ que genera aquest subgrup cíclic del grup de punts, $\langle G\rangle \subseteq E(\mathbb{F}_p)$. Sí $G' \in E(\mathbb{F}_p)$ és també d'ordre n, és fàcil veure que la següent aplicació entre $\langle G\rangle$ i $\langle G'\rangle$,

$$\begin{array}{rcl} f : \langle G\rangle & \longrightarrow & \langle G'\rangle, \\ f([k]G) & \longmapsto & [k]G', \forall k \in \{0, \ldots, n-1\}, \end{array}$$

és un isomorfisme de grups. En la pràctica, es manté una corba precalculada com per exemple una del [NIST186], per la que $|E(\mathbb{F}_p)| = hn$, amb cofactor $h \lll n$ i n primer (per aquestes corba NIST és $h = 1$), i es pot generar el punt G' com un punt aleatori en $E(\mathbb{F}_p)$ vigilant que **no** tingui l'ordre del cofactor (definició 2.13), tindrem un isomorfisme, com l'explicat anteriorment, entre el subgrup cíclic recomanat $\langle G\rangle$ i el $\langle G'\rangle$.

4.1.2 Què és una isogènia?

Definició 4.3. Donades dos corbes el·líptiques $E/\mathbb{K}$ i $E'/\mathbb{K}$, son *isogènes sobre* $\mathbb{K}$ **sii** existeix una aplicación racional $E \xrightarrow{\mathcal{I}} E'$ que transforma $\mathcal{O}_E$ en $\mathcal{O}_{E'}$, es a dir,

$$\begin{array}{rcl} \mathcal{I} : E & \longrightarrow & E', \\ (x, y) & \longmapsto & (X, Y), \\ \mathcal{O}_E & \longmapsto & \mathcal{O}_{E'}, \end{array} \tag{4.1}$$

on X i Y responen a $X = f_1(x)$ i $Y = f_2(x, y)$, essent f_1 i f_2 funcions $\mathbb{K}$-racionals, i.e., quocients de polinomis de $\mathbb{K}[x, y]$. A l'aplicació $\mathcal{I}$ se l'anomena *isogènia sobre* $\mathbb{K}$ entre les corbes el·líptiques E i E'.

Les corbes isógenies tenen un parell de propietats que ens interessen especialment i que enunciem breument.

1. Sí $\mathcal{I} : E/\mathbb{F}_p \to E'/\mathbb{F}_p$ és una isogènia, llavors $\mathcal{I}$ també és un morfisme dels grups de punts, es a dir, per qualsevol parell $P, Q \in E(\mathbb{F}_p)$ es compleix que
$$\underbrace{\mathcal{I}(P + Q)}_{\text{en } E(\mathbb{F}_p)} = \underbrace{\mathcal{I}(P) + \mathcal{I}(Q)}_{\text{en } E'(\mathbb{F}_p)}.$$

2. Sí $E/\mathbb{F}_p$ i $E'/\mathbb{F}_p$ són isògenes llavors tenen el mateix cardinal,

$$|E(\mathbb{F}_p)| = |E'(\mathbb{F}_p)|,$$

i recíprocament, si dues corbes el·líptiques $E/\mathbb{F}_p$ i $E'/\mathbb{F}_p$ tenen el mateix cardinal llavors existeix una isogènia, sobre $\mathbb{F}_p$, entre elles.

Les isogènies entre corbes el·líptiques definides sobre cossos finits que nosaltres usarem permeten la següent definició de *grau de la isogènia*.

Definició 4.4 (Grau d'una isogènia)**.** Donada una isogènia $E/\mathbb{F}_p \xrightarrow{\mathcal{I}} E'/\mathbb{F}_p$ definirem el seu *grau* com el cardinal del seu núcli, $|\ker \mathcal{I}|$.

L'escenari en el que pensem que són d'utilitat les isogènies entre corbes el·líptiques és el d'un ECC que es suposa en risc i que s'ha de *resetejar*. Es pot reiniciar tot el ECC o es pot passar a una altra corba el·líptica isògena amb l'inicial del ECC. Aquesta manera, segons l'apartat 2 anterior, els paràmetres del ECC associats al cardinal de la corba, l'ordre n del grup criptogràfic, el cofactor h i, clar, el propi cardinal $|E(\mathbb{F}_p)|$ no varïen. Igualment es pot mantenir la clau sécreta $1 < d < n$, tot i que canviant la pública $P = [d]G$, al seu punt isogen, ja que per l'apartat 1 es tindrà

$$\mathcal{I}(P) = \mathcal{I}([d]G) = [d]\mathcal{I}(G). \tag{4.2}$$

Però en un reseteig per canvi de corba a una isógena, apareix un altre problema: sí es coneix l'isogènia l'atacant del ECC pot aplicar-la als punts que hagi anat calculant en el seu atac i així fer vàlid aquest atac per la nova corba. El reseteig no haurà servit de res. És, doncs, important que l'isogènia, concretament el seu grau, no sigui conegut per l'atacant. Per això poden servir les anomenades *estrelles d'isogènies*.

Així com s'ha dit, i es veurà més endavant que un *isomorfisme* no es efectiu contra atacs directes, però si ho es contra atacs laterals (2.3.2), una *isogènia* **si** resulta efectiva davant un atac directe contra el criptosistema descrit en la secció 2.3.1.

4.1.3 Volcans, serralades i estrelles de corba el·líptica

S'han comentat les cites [RS06] i [isoTFC], on es proposa utilitzar les *isogènies* per resetejar un ECC, utilitzant estructures composades per corbes el·líptiques isogènas. El camí seguit per arribar a una corba el·líptica en aquestes estructures és un camí fàcil de realitzar, però computacionalment molt dur de resseguir per un atacant. Concretament, en el citat article [RS06] és

considera el problema d'atacar una estructura de isogènies con un problema dur inclús per un hipotètic computador quàntic.

Les isogènies de corba el·líptica poden formar estructures i visualitzarles en forma de grafs. Diferenciant les característiques de com obtenir una transició d'isogènia ens permeten construïr formes que ens recordin objectes del nostre imaginari. En diem volcà d'isogènies per la forma que li fem agafar a la vegada que a un conjunt de volcans en diem serralada. Exemple clar d'això també son les estrelles d'isogènies.

Definició 4.5. Definim un *ℓ-volcà* com un graf dirigit on els nodes son corbes el·líptiques i les arestes representen isogènies de grau ℓ.

Aquests grafs –*volcans* tenen una estructura "per pisos" porque cada corba–node pot tenir fins $\ell + 1$ ℓ-isogènies. Cada una d'elles té una orientació, a saber, *horitzontal*, *descendent* o *ascendent*, que es pot determinar a partir dels *anells d'endomorfismes* de las corbas de partida i d'arribada. Degut a aquestes direccions possibles, podem construïr un graf que tindrà forma de volcà, format per un ciclo–*cràter*, on dels seus nodes o vèrtex pengen arbres ℓ-àris complets, excepte el cas que el volcà estigui reduït a sols el cràter. Totes les fulles dels subarbres del volcà estan en el mateix nivell per a tot el volcà. Excepte les fulles, tots el nodes del volcà tenen $\ell + 1$ arestes. Concretament, els nodes dels subarbres tenen una isogènia ascendent i ℓ descendents i les del cràter en tenen dues d'horitzontals i $\ell - 1$ descendents. Un exemple de volà el podem veure en la figura 4.1.1 que s'ha extret d'una figura de l'article [MSTTV07].

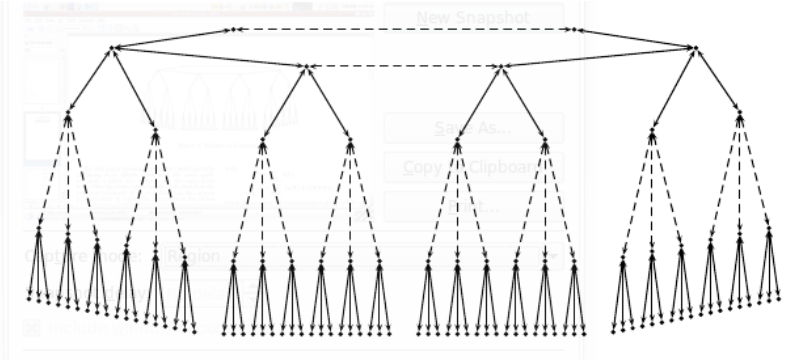


Figura 4.1.1: Exemple de volcà extret de [MSTTV07].

Sí aquests grafs construïts mitjançant ℓ-isogènies no son connexos, obtenim un conjunto de volcans que es poden denominar *serralades*: totes les corbes–node de tots els volcans tindran, doncs, el mateix cardinal. Però fi-

xada una sólo son accessibles, per composició repetida de ℓ-isogènies, les que estan en el mateix volcá.

Considerem ara ℓ_i-volcans d'isogènies per diferents primers ℓ_i, tots ells reduïts al cráter, i construïts a partir de una mateixa corba $E/\mathbb{F}_p$. En aquest cas, els multiples camins de isogènies es veuen superposats i ordenats de manera que prenen formes de polígons estrellats. D'aquí que a aquest graf se'l coneix com *estrella d'isogènies*. En la figura 4.1.2 en diferents colors estan representats els diferents cicles que són cada ℓ_i-cràter.

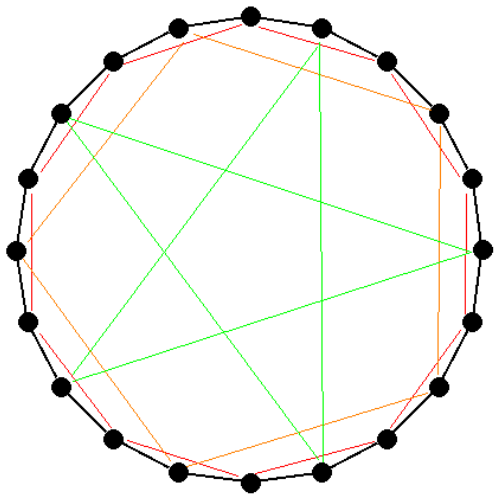


Figura 4.1.2: Exemple d'estrella d'isogènies

4.2 Possibilitats de l'ús d'estrelles de corbes el·líptiques

Al moment actual, les simulacions per tal d'obtenir una nova corba el·líptica a partir d'una inicial comú no han resultat satirfactories. Es tracta principalment que sigui un procés que un usuari pugui realitzar i que prengui un temps raonable pel temps de generar una claus nou. Per corbes relativament petites els temps son considerables i desanimen a intentar-ho per corbes majors. La intenció es aproximar el temps de creació d'un parell de claus el·líptiques utilitzant isogènies, al temps de creació d'un parell de claus en cossos finits d'un nivell de seguretat equivalent.

Hi ha un problema afegit a aquesta resticció, i és que les corbes que recomanen els estàndard no son les millors corbes per aplicar-hi algorismes

d'isogènia i compliquen l'ideal de trobar un procediment que calculi una nova corba el·líptica criptogràficament útil en un temps raonable.

S'està buscant que el sistema de *setting* o *resetting* de un parell de claus el·líptiques sigui computable per part de l'usuari final. Es podrien precalcular desenes de corbes que després fossin seleccionades per d'usuari, però l'obligariem a confiar en el nostre bon fer a l'hora d'*"oblidar"* el camí recorregut per trobar la seva corba. En criptografia sempre serà molt millor oferir un algorisme confiable i una bona implementació i que l'usuari pugui auditar tot el procediment. Així, com es vol que el procés l'executi l'usuari, ha de ser un procés que requereixi d'un temps sensat per ser computat en les plataformes on corri el binaris finals.

Resulta primordial, doncs, que el temps de creació d'un parell de claus el·líptiques noves sigui de l'ordre de la creació d'un parell de claus sobre cossos finits d'una longitud equivalent segons la taula donada en la secció 12 de [ECPGP]. Actualment podem dir que estem molt lluny d'aquest objectiu ja que el sistema proposat en articles i treballs utilitzen corbes el·líptiques sobre uns cossos finits minsos. Amb la implementació actual, la creació d'un parell de claus el·líptiques resulta molt ràpid doncs com a màxim es generarà una clau secreta d d'una longitud d'uns 521 bits; mentre la la creació d'una clau en cossos finits de 2048 bits requereix més temps, consumeix molta més entropia i sols és equivalent a una clau el·líptica de 224 bits.

L'us d'estrelles d'isogènies per (re)setejar el criptosistema no hauria de superar els temps sobre cossos finits i això no sembla trivial d'aconseguir amb les simulacions realitzades fins ara. L'algorisme es senzill però tenim que conseguir que els seus passos es puguin realitzar molt de presa.

ALGORISME 11 (Procediment de Setting d'una corba el·líptica)

INPUT:: Primer p sobre el que definir la corba el·líptica $E/\mathbb{F}_p$ i k longitud del recorregut a realitzar.
OUTPUT: Una corba el·líptica sobre un cos finit, d'una estrella d'isogènies.

1: Seleccionar una corba el·líptica inicial E_0 sobre $\mathbb{F}_p$;
2: Generar una ruta aleatòria L de longitud k d'elements $L = \{r_0, r_1, \ldots, r_k\}, t.q. \forall r \in_R \ldots$;
3: $E' \leftarrow L(E_0)$;
/*Calcular la corba E' resultant del recorregut L en l'estrella */
4: Retornar E';

Es un algorisme similar a un de xifrat, però del que podem i devem

oblidar els valors aleatòria de la llista per tal de mantenir aquest viatge per dins l'estrella en secret. Si ni tant sols es guarda, segur que no podrà deixar-se exposat. El procés de reseteig no seria molt diferent ja que tampoc resulta recomanable mantenir la clau secreta antiga podent escollir un valor nou de clau secreta d per obtenir una clau pública en la nova corba el·líptica.

Capítol 5

Gnu Privacy Guard

Quan el projecte [BM04] va començar no existien grans pretencions, però si la il·lusió de veure'l integrat en el *GnuPG*. Es important fer software lliure, però no s'ha de deixar que la feina realitzada quedi en l'oblit i es perdi.

5.1 Assignment - GNU GPG

Després de un llarg treball que culminar amb l'implementació d'un pegat per la rama *1.4* del programa lliure *GnuPG* es va iniciar un projecte de manteniment i millora amb la publicació d'una petita pàgina web on allotjar els fonts del projecte i la seva documentació de cara a l'escrutini públic.

Va funcionar. Poc més de 6 mesos després, sobre el novembre de 2004, un company rus, de nom Mikael Mylnikov, es va posar en contacte per indicar un mal funcionament o bug en l'implementació de la signatura digital. La implementació era capaç de verificar signatures que no eren. Era una errada greu que havia passat per alt als tests realitzats i uns bons ulls extern van poder veure. La trobada de l'error significa l'inici d'un bon treball que culmina amb l'aportació d'un algorisme de xifrat més robust. La publicació com a programàri lliure ha pres un sentit complert.

Molt posteriorment es va iniciar un *fil de discusió*[1] en la llista de desenvolupadors del *GnuPG* on es demanava suport per corba el·líptica en la llibreria *libgcrypt*. El codi del projecte [BM04] va ser portat per en *Werner Koch* del pegat preparat per la rama 1.4 cap a la mencionada llibreria matemàtica i criptogràfica de la branca 2 (en aquell moment branca 1.9) del *GnuPG*.

El fet de tenir ja la signatura digital portada anima a implicar-se amb aquest programa i a proposar l'aportació més directa i dinàmica de codi. Ai-

[1] El podem trobar a l'adreça http://lists.gnupg.org/pipermail/gnupg-devel/2007-March/023725.html

xò té dues opcions: agreements puntuals amb la *Free Software Fundation* i el projecte *GNU* per tal d'aportar algunes línies o arribar a un acord i signar un assignment amb la *FSF* i formar part dels desenvolupadors i mantenedors d'aquest software. A partir d'aquesta signatura, s'entra a formar part del grup de desenvolupadors que poden enviar codi per la seva avaluació i posterior publicació.

5.2 Dos branques, dos codis

S'ha comentat a la secció 5.1 l'aparició en la llibreria matemàtica *libgcrypt* de suport per a corba el·líptica escrit per en *Werner Koch* basat en el pegat [BM04]. Després s'ha comentat l'entrada al grup de desenvolupadors del *GnuPG*. I tot això junt porta a l'actual manteniment de dues branques de codi: una (amb unes primitives) per al *GnuPG* 1.4.x, i una altra per a la *libgcrypt* 1.4.y (usat pel *GnuPG* 2.0.z).

És important veure clar com es fan les coses en cada programa de cara a mantenir una sola línia sense divergències ni de disseny ni d'implementació i a la vegada esforçar-se a no cometre errors. Amb tota probabilitat, el desenvolupament es centrarà en el treball en la llibreria deixant obsolet el pegat del projecte [BM04]. Actualment s'està treballant per implementar l'estàndard [ECPGP] que s'ha comentat en la secció 2.2.1.3 i els algorismes s'han comentat en la secció 2.2.3.4.

A continuació es presenten unes taules on es divideixen en blocs les funcions, mètodes i primitives de les dues implementacions publicades; contraposant-les i oferint una visió del que cadascuna implementa i com ho defineix. No s'han inclòs aquelles funcions que estan essent incloses però que no apareixen actualment en els repositoris públics. Indicar també que s'ha seguit per a definició aquí el *coding standard* del projecte *Gnu*.

La primera de les taules és fonamental. A la taula 5.1 recopila les estructures de dades implementades per les dues branques en el referent a l'implementació de corba el·líptica. Resulta fonamental pel fet que serà continuament referenciada des de que es vulgui passar un parametre o referència a qualsevol de les funciones de la resta de taules. Entre les dues branques no hi ha diferències més que en la nomenclatura de les variables, el contingut de l'estructura segueix sent el mateix. Hi ha però si algunes estructures noves en la *libgcrypt* i això es deu a una millor implementació de l'assignació de valor preestablerts en els estàndards i normatives comentats en la secció 2.2.1.

La taula 5.2 recull d'Interfície entre el núcli en les diferents rames i el mòdul propiament. El *GnuPG* resulta exemplar en aplicar be patrons i el que fa es tractar cada algorisme o criptosistema com una unitat. D'aquí va

venir que el projecte [BM04] estiques completament codificat en un fitxer de codi i un de capçalera. Com veurem en posteriors taules, la llibreria *libgcrypt* dona un pas més i trasllada el codi més matemàtic cap a l'interior del seu nucli de càlcul criptogràfic.

Destacar que algunes funcions d'Interfície primordials, com son el xifrat i el desxifrat no tenen contrapartida en la columna de la *libgcrpt*, això es deu a que aquesta llibreria sols té publicada en aquest moment l'algorisme *ECDSA* i que s'ampliarà en breu cap al suport de l'estàndard [ECPGP] que s'ha esmentat en la secció 2.2.1.3.

Succeeix el mateix per parlar de la taula 5.3 on les funcions de xifrat i desxifrat no mostren especificació en la segona columna i es per la mateixa raó, ja que sols s'estan mostran aquelles funcions que es poden descarregar des del repositori públic. Si, però, queda patent la referència feta a la traducció directa d'estructures (taula 5.1) ja que tant la firma com la verificació l'esquema seguit es el mateix.

Entrant amb més detall cap a l'interior del mètodes criptogràfics tenim la taula 5.4 amb els mètodes que no son cridats directament per l'Interfície citada en la taula 5.2. Principalment son els mètodes que defineixen la generació d'un parell de claus i tot el seu setup. Per exemple, l'us de estrelles d'isogènies del que s'ha parlat en el capítol 4 modificaria el contingut de la funció `generate_curve()` i no afectaria a la resta que ja han estat pensats per no dependre de una llista estàtica de corbes.

Queden també en aquesta taula 5.4 uns mètodes al final que tenen relació directa amb el sistema de xifrat proposat en [BM06]. L'implementació del xifrat en la *libgcrypt* està feta seguint el proposat a [ECPGP] llavors els mètodes criptogràfics auxiliars seran uns altres i es veurà que la traducció no és possible. Els algorismes son incompatibles entre ells de manera que es descarta la seva convivència.

La taula 5.5 presenta forats a totes dues bandes i es deu a diferents motius. Els primers mètodes que sols existeixen, de moment en el pegat per la rama 1.4 del *GnuPG* es deuen a que en aquest s'implementa la protecció al atacs laterals de la secció 3.3.4 per utilitzar un isomorfisme del grup generador així com també l'us de diferents representacions projectives d'un punt com es deia en la secció 3.3.3.

També estan en la taula que tractem ara i estan en les dues rames les funcions fonamentals per les operacions de suma de punts que es deien en les definicions 2.9, duplicat d'un punt de la definició 2.10 i la fonamental per al logaritme discret el·líptic que es la `escalarMult()` que prové de la definició 2.11.

Al final d'aquesta taula 5.5 es fan referència a métodes per operar enters del cos finit base sobre el que viu la corba el·líptica. Això es deu a que

l'eficàcia dels mètodes genèrics de la llibreria *libgcrypt* no resulten eficients per als proporcionalment mes curts valors de les coordenades d'un punt de corba el·líptica.

Finalment ja sols ens queda una taula, la 5.6 que tanca el cicle al ser l'últim nivell que a més enllaça amb la primera taula 5.1, la de les estructures, i aquí hi ha els mètodes sobre com inicialitzar o alliberar la memòria d'aquestes, copiar-les, o resoldre l'equació de la corba el·líptica.

Patch 1.4	libgcrypt		
cipher/ecc.c	cipher/ecc.c	src/mpi.h	

`typedef struct{`
`  MPI x_;`
`  MPI y_;`
`  MPI z_;`
`} point;`
		`struct mpi_point_s;`
		`typedef struct mpi_point_s mpi_point_t;`
		`struct mpi_point_s {`
		`  gcrypt_mpi_t x;`
		`  gcrypt_mpi_t y;`
		`  gcrypt_mpi_t z;`
		`};`	

`typedef struct {`
`  MPI p_;`
`  MPI a_,b_;`
`  point G;`
`  MPI n;`
`} ellipticCurve;`
	`typedef struct {`
	`  gcrypt_mpi_t p;`
	`  gcrypt_mpi_t a,b;`
	`  mpi_point_t G;`
	`  gcrypt_mpi_t n;`
	`} elliptic_curve_t;`		

`typedef struct {`
`  ellipticCurve E;`
`  point Q;`
`} ECC_public_key;`
	`typedef struct {`
	`  elliptic_curve_t E;`
	`  mpi_point_t Q;`
	`} ECC_public_key;`		

`typedef struct {`
`  ellipticCurve E;`
`  point Q;`
`  MPI d;`
`} ECC_secret_key;`
	`typedef struct {`
	`  elliptic_curve_t E;`
	`  mpi_point_t Q;`
	`  gcrypt_mpi_t d;`
	`} ECC_secret_key;`		

		`static const struct {`
		`  const char *name;`
		`  const char *other;`
		`} curve_aliases[]`	

		`static const struct {`
		`  const char *desc;`
		`  unsigned int nbits;`
		`  const char *p;`
		`  const char *a,*b;`
		`  const char *n;`
		`  const char *g_x,*g_y;`
		`} domain_params[]`	

		`static const char *ecdsa_names[]`	
		`gcry_pk_spec_t` `_gcry_pubkey_spec_ecdsa`	

Taula 5.1: Taula d'estructures utilitzades en les rames del *GnuPG*

Patch 1.4	libgcrypt
cipher/ecc.c	cipher/ecc.c
`int ecc_generate(int algo, unsigned nbits, MPI *skey, MPI **refactors)`	`gcry_err_code_t _gcry_ecc_generate(int algo, unsigned int nbits, const char *curve, gcry_mpi_t *skey, gcry_mpi_t **retfactors)` `static gcry_err_code_t ecc_generate(int algo, unsigned int nbits, unsigned long dummy, gcrypt_mpi_t *skey, gcry_mpi_t **retfactors)`
`int ecc_check_secret_key(int algo, MPI *skey)`	`static gcry_err_code_t ecc_check_secret_key(int algo, gcry_mpi_t *skey)`
`int ecc_encrypt(int algo, MPI *resarr, MPI data, MPI *pkey)`	
`int ecc_decrypt(int algo, MPI *result, MPI *data, MPI *skey)`	
`int ecc_sign(int algo, MPI *resarr, MPI data, MPI *skey)`	`static gcry_err_code_t ecc_sign(int algo, gcry_mpi_t *resarr, gcry_mpi_t data, gcry_mpi_t *skey)`
`int ecc_verify(int algo, MPI hash, MPI *data, MPI *pkey)`	`static gcry_err_code_t ecc_verify(int algo, gcry_mpi_t hash, gcry_mpi_t *data, gcry_mpi_t *pkey, int (*cmp)(void *,gcry_mpi_t), void *opaquev)`
`unsigned int ecc_get_nbits(int algo, MPI *pkey)`	`static unsigned int ecc_get_nbits(int algo, gcry_mpi_t *pkey)`
`const char * ecc_get_info(int algo, int *npkey, int *nskey, int *nenc, int *nsig, int *use)`	
	`gcry_err_code_t _gcry_ecc_get_param(const char *name, gcry_mpi_t *pkey)`
	`static gcry_mpi_t ec2os(gcry_mpi_t x, gcry_mpi_t y, gcry_mpi_t p)`
	`static gcry_error_t os2ec(mpi_point_t *result, gcry_mpi_t value)`

Taula 5.2: Taula amb les funcions d'interficie d'un mòdul del *GnuPG*

Patch 1.4	libgcrypt
cipher/ecc.c	cipher/ecc.c

`static void`
`doEncrypt(MPI input,`
`ECC_public_key *pkey,`
`point *R, MPI c)`	

`static MPI`
`decrypt(MPI output,`
`ECC_secret_key *skey,`
`point *R, MPI c)`	

`static void`
`sign(MPI input,`
`ECC_secret_key *skey,`
`MPI *r, MPI *s)`
	`static gpg_err_code_t`
	`sign(gcry_mpi_t input,`
	`ECC_secret_key *skey,`
	`gcry_mpi_t r, gcry_mpi_t s)`

`static int`
`verify(MPI input,`
`ECC_public_key *pkey,`
`MPI r, MPI s)`
	`static gpg_error_code_t`
	`verify(gcry_mpi_t input,`
	`ECC_public_key *pkey,`
	`gcry_mpi_t r, gcry_mpi_t s)`

Taula 5.3: Taula amb els mètodes criptogràfics generals

Patch 1.4	libgcrypt
cipher/ecc.c	cipher/ecc.c
static MPI gen_k(MPI p, int secure)	static gcry_mpi_t gen_k(gcry_mpi_t p, int security_level)
static void generateCurve(unsigned nbits, ellipticCurve *ECC_curve)	static gpg_err_code_t generate_curve(unsigned int nbits, const char *name, elliptic_curve_t *curve, unsigned int *r_nbits)
static void generateKey(ECC_secret_key *sk, unsigned nbits)	static gpg_err_code_t generate_key(ECC_secret_key *sk, unsigned int nbits, const char *name, gcry_mpi_t g_x, gcry_mpi_t g_y, gcry_mpi_t q_x, gcry_mpi_t q_y)
static void testKeys(ECC_secret_key *sk, unsigned nbits)	static void test_keys(ECC_secret_key *sk, unsigned int nbits)
static int check_secret_key(ECC_secret_key *sk)	static int check_secret_key(ECC_secret_key *sk)
static void progress(int c)	void _gcry_register_pk_ecc_progress(void (*cb) (void *, const char *, int, int, int), void *cb_data)
static void sha256_hashing(MPI input, MPI *output)	
static void aes256_encrypting(MPI key, MPI input, MPI *output)	
static void aes256_decrypting(MPI key, MPI input, MPI *output)	

Taula 5.4: Taula amb mètode criptogràfics auxiliars

Patch 1.4	libgcrypt
cipher/ecc.c	mpi/ec.c
`static int genBigPoint(MPI prime, elliticCurve *base, point *G, unsigned nbits)`	
`static point genPoint(MPI prime, ellipticCurve *base)`	
`static MPI modularSquareRoot(MPI a, MPI modulus)`	
`static MPI symbolLegendre(MPI n, MPI modulus)`	
`static int PointAtInfinity(point Query)`	
`static void escalarMult(MPI escalar, point *P, point *R, ellipticCurve *base)`	`void _gcry_mpi_ec_mul_point(mpi_point_t *result, gcry_mpi_t scalar, mpi_point_t *point, mpi_ec_t ctx)`
`static void sumPoints(point *P0, point *P1, point *P2, ellipticCurve *base)`	`void _gcry_mpi_ec_add_points(mpi_point_t *result, mpi_point_t *p1, mpi_point_t *p2, mpi_ec_t ctx)`
`static void duplicatePoint(point *P, point *R, ellipticCurve *base)`	`void _gcry_mpi_ec_dub_point(mpi_point_t *result, mpi_point_t *point, mpi_ec_t ctx)`
`static void invertPoint(point *P, ellipticCurve *base)`	
	`static void ec_addm(gcry_mpi_t w, gcry_mpi_t u, gcry_mpi_t v, mpi_ec_t ctx)`
	`static void ec_subm(gcrt_mpi_t w, gcry_mpi_t u, gcry_mpi_t v, mpi_ec_t ctx)`
	`static void ec_mulm(gcrt_mpi_t w, gcry_mpi_t u, gcry_mpi_t v, mpi_ec_t ctx)`
	`static void ec_powm(gcrt_mpi_t w, const gcry_mpi_t b, const gcry_mpi_t e, mpi_ec_t ctx)`
	`static void ec_invm(gcrt_mpi_t x, gcry_mpi_t a, mpi_ec_t ctx)`

Taula 5.5: Taula amb les funcions matemàtiques

Patch 1.4	libgcrypt	
cipher/ecc.c	cipher/ecc.c	src/mpi.h
static point point_copy(point *P)	static void point_set(mpi_point_t *d, mpi_point_t *s)	static void point_set(mpi_point_t *d, mpi_point_t *s)
	#define point_init(a) _gcry_mpi_ec_point_init((a))	#define point_init(a) _gcry_mpi_ec_point_init((a)) void _gcry_mpi_ec_point_init(mpi_point_t *p)
static void point_free(point P)	#define point_free(a) _gcry_mpi_ec_point_free((a))	#define point_free(a) _gcry_mpi_ec_point_free((a)) void _gcry_mpi_ec_point_free(mpi_point_t *p)
static int point_affine(point P, MPI x, MPI y, ellipticCurve *base)		int _gcry_mpi_ec_get_affine(gcry_mpi_t x, gcry_mpi_t y, mpi_point_t *point, mpi_ec_t ctx)
		mpi_ec_t _gcry_mpi_ec_init(gcry_mpi_t p, gcry_mpi_t a)
static ellipticCurve curve_copy(ellipticCurve *E)	static elliptic_curve_t curve_copy(elliptic_curve_t E)	
static void curve_free(ellipticCurve E)	static void curve_free(elliptic_curve_t *E)	void _gcry_mpi_ec_free(mpi_ec_t ctx)
static MPI get_bit()		
static MPI gen_y_2(MPI x, ellipticCurve *base)	static gcry_mpi_t gen_y_2(gcry_mpi_t x, elliptic_curve_t *base)	
	static gcry_mpi_t scanval(const char *string)	

Taula 5.6: Taula amb mètodes per manupular estructures.

Capítol 6

Conclusions i treball futur

Recopilant la feina feta en aquest projecte queda desenvolupament pendent. El desenvolupament més extens que queda és l'implementació de les isogènies. Com s'ha comentat és vol tenir unes restriccions per a la creació del parell de clau que, actualment, no resulten trivials d'assolir. La longitud del viatge dins de l'estrella d'isogènies feta amb una ℓ-erralada definida en **??** que té la forma de la figura 4.1.2.

A més d'una trava matemàtica a resoldre per una implementació ràpida i eficient, hi ha una restricció des dels estàndards. L'estàndard per a la implementació [ECPGP], que actualment està en fase *internet draft* però que esdevindrà definitiu aviat, sols contempla una reduïda de corbes el·líptiques, una per a cada longitud, codificant-la amb un index contingut en la clau. D'aquesta manera la clau resulta molt menor, però dificulta usar-ne d'altres. En comunicació amb *Andrey Jivson*, autor de l'estàndard, la solució passaria per escriure un *extension draft* per permetre altres típus de corba i que aquestes puguin ser contingudes en la estructura de la clau.

En la branca 1.4 hi ha tenim les primitives necessàries per quasi totes les operacions que els càlculs de l'isogènia requereix però un cop completat el recull de primitives aquestes existents haurien de ser revisades per a que el seu procediment sigui òptim i no alenteixi el procès de la mateixa manera que en la secció 5.2 s'ha vist que que les operacions sobre cossos finits s'han d'optimitzar per la menor longitud i major repetibilitat amb que ens trobem quan usem corbes el·líptiques.

De tota manera el desenvolupament sobre la branca 1.4 toca a la seva fi, i el treball natural s'anirà decantant cap a directament la llibreria *libgcrypt* i la branca 2 del *GnuPG*. Per tant la ben pròxima publicació de l'implementació de l'estàndard [ECPGP], en especial l'apartat del xifrat i desxifrat, està previst de fer-se ja solsament sobre l'implementació de la *libgcrypt*, ampliant així el suport que ja té per l'algorisme de firma digital ECDSA.

Resumint, primordialment s'acabarà amb l'implementació del xifrat de corba el·líptica per a la rama 2, deixant-la a la mateixa posició que la rama 1.4 (tot i que amb algorismes diferents) i deixant aquesta última en desús. Després es reforçarà les implementacions, sota la cobertura dels estàndards, proteccions als atacs laterals tractats en el capítol 3 i en paral·lel es vol posar en pràctica la viabilitat de les estrelles d'isogènies del capítol 4. Es preveu la publicació d'articles per part dels autors a cada pas que es vagi donant.

Bibliografia

[1] Llibres i revistes

[HANDECC] Henri Cohen, Gerhard Frey *Handbook of Elliptic and Hyperelliptic Curve Cryptography* Ed. Chapman & Hall

[LMS265] Ian Blake, Gadiel Seroussi, Nigel Smart, *Elliptic Curve in Cryptography* Ed. Cambridge University Press

[LMS317] Ian Blake, Gadiel Seroussi, Nigel Smart, *Advances in Elliptic Curve Cryptography* Ed. Cambridge University Press

[ECDSA] D. Johnson, A. Menezes, S. Vanstone, *The elliptic Curve Digital Signature Algorithm (ECDSA)*. Dept. of Combinatorics & Optimization, University of Waterloo, Certicom, Canada.

[Men93] Alfred Menezes, *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers, 1993

[BS96] Bruce Schneier, *Appiled cryptography*.

[H9DL] Dineal Lerch, *Criptografía de Curva Elíptica: Ataque de Rho de Pollard*, article en *Hakin9*, hakin9.org/upload/hakin9/PDFVersion/curvas_elipticas.pdf

[2] Collita pròpia

[BM04] Sergi Blanch, Ramiro Moreno, *GnuPG Implementation with Elliptic Curves* Treball final de carrera, Enginyeria tècnica en Informàtica de Sistemes, Universitat de Lleida. Març 2004.

[BM04-1] Sergi Blanch, Ramiro Moreno, *Implementació GnuPG con Curvas Elípticas* Recsi'04, Universidad Carlos III, 2004.

[BM06] Sergi Blanch, Ramiro Moreno, *Análisis del cifrado ElGamal de un módulo con curvas elípticas propuesto para el GnuPG* Recsi06, Universitat Autónoma de Barcelona i Universitat Oberta de Catalunya, 2006.

[3] Estàndards

[P1363] IEEE P1363/D13 (Draft Version 13) Standard Specifications for Public key Cryptography. 1999 November 12.

[NIST197] FIPS PUB 197, Advanced Encryption Standard (AES), U.S.Department of Commerce/National Institute of Standards and Technology. 2001.

[NIST180] FIPS PUB 180-2, Secure Hash Standard (SHS), U.S.Department of Commerce/National Institute of Standards and Technology. 2002.

[NIST186] FIPS PUB 186-3, Digital Signature Standard (DSS), U.S.Department of Commerce/National Institute of Standards and Technology. March 2006.

[NIST800] Nist Special Publication 800-56A *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography* March 2007.

[ECPGP] A. Jivsov, *ECC in OpenPGP*, IETF internet Draft, April 2008

[GNUSTD] R. Stallman, *Gnu Coding Standards*, July 2005.

[rfc2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*. 1997 March.

[rfc3278] S. Blake-Wilson, D. Brown, P. Lambert, *Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)*. 2002 April.

[rfc3394] J. Schaad, R. Housley, *Advanced Encryption Standard (AES) Key Wrap Algorithm* September 2002.

[rfc4492] S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, B. Moeller, *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*. 2006 May.

[rfc4880] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, R. Thayer, *OpenPGP Message Format*. 2007 November

[brainpool] ECC Brainpool Standard Curves and Curve Generation. 2005 October 19.

[sec1] SEC 1. Standards for Efficient Cryptography Group: Elliptic Curve Cryptography.

[sec2] SEC 2. Standards for Efficient Cryptography Group: Recommended Elliptic Curve Domain Parameters.

[4] Atacs directes

[Silv99] J. Silverman, *The Xedni calculus and the elliptic curve discrete logarithm problem*, Designs, codes and Cyptography, n20 (2000), 5-40.

[Xedni99] M. j. jacobson, N. Koblitz, J. H. Silverman, A. Stein, E. Teske, *Analysis of the Xedni Calculus Attack*, 1999.

[mov97] Alfred J, Menezes, Paul C.van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, first edition, 1997. http://cacr.math.uwaterloo.ca/hac/

[5] Atacs laterals

[HAGAI] Hagai Bar-El, *Introduction to Side Channel Attacks* White paper, Discretix Technologies Ltd.

[KJJ] Paul Kocher, Joshua Jaffe, Benjamin Jun, *Introduction to Differential Power Analysis and Related Attacks* Cryptography Resarch Inc, San Fancisco.

[BJN] Dan Boneh, Antonie Joux and Phong Q. Nguyen, *Why textbook ElGamal and RSA encryption are Insecure.*

[MZ] Serge Mister and Robert Zuccherato, *An Attack on CFB Mode Encryption As Used By OpenPGP*

[6] Isogènies

[MSTTV07] J, Miret, D. Sadornil, J. Tena, R. Tomàs, M. Valls, *Isogeny cordillera algorithm to obtain cryptographically good elliptic curves.*

[MMSTV06] J. Miret, R. Moreno, D. Sadornil, J. Tena, M. Valls, *An algorithm to compute volcanoes of 2-isogenies of elliptic curves over finite fields.*

[Galbr98] S. Galbraith, *Constructing isogenies between elliptic curves over finite fields.*

[LM98] R. Lercier, F. Morain, *Algorithms for computing isogenies between elliptic curves.*

[FM05] M. Fouquet, F. Morain, *Isogeny volcanoes and the SEA algorithm.*

[BMSS06] A. Bostan, F. Morain, B. Salvy, E. Schost, *Fast algorithms for computing isogenies between elliptic curves.*

[RS06] A. Rostovtsev, A. Stolbunov, *Public-key cryptosystem based on isogenies.*

[MTRVM] S. Martinez, R. Tomàs, C. Roig, M. Valls, R. Moreno, *Parallel Calculation of Volcanoes for Cryptographyc Uses.*

[isoTFC] R. Arias, J. Miret, *Implementacón de un criptosistema de clave pública basado en estrellas de isogenias* TFC Ingeniería Informática UdL.