Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica en Informàtica de Sistemes

Treball Final de Carrera

Disseny d'una plataforma de votació electrònica: Mòdul criptogràfic

Autora:
Núria Busom Figueres

Directors:
Josep Maria Miret Biosca
Francesc Sebé Feixas

Juliol del 2010

Índex

Capítol 1

Introducció

En aquest capítol es presenta el concepte de votació electrònica tant en el context actual com en el marc històric. També es defineix el propòsit d'aquest treball.

1.1 Antecedents i motivacions

Els aparells electrònics i els sistemes de telecomunicacions ens faciliten cada cop més el dia a dia. Els ordinadors són cada cop més potents, més econòmics i més accessibles per a tothom i les comunicacions són cada cop més segures.

Avui en dia podem realitzar infinitat de tràmits a través d'Internet, no només coses senzilles com fer la compra al supermercat del costat de casa (o a un que es troba a milers de kilòmetres) o comprovar l'estat d'un paquet que ens han d'enviar, sinó signar correus electrònics, demanar documents oficials, pagar tributs o sancions,... tot identificant-nos amb el DNI electrònic. Tenint en compte totes aquestes facilitats, haver de desplaçar-se fins a un col·legi electoral per poder emetre el nostre vot, que els membres de la mesa electoral hagin de romandre allí per validar la nostra identitat i garantir que el procés s'ha dut a terme sense irregularitats,... sembla que hauria de formar part de la història i no pas de la nostra actualitat.

Per a les persones que no han format part del disseny d'un sistema de votació electrònica, aquest no és més que una caixa negra. Han de realitzar l'acte de fe de creure que la votació es duu a terme de forma íntegra i confidencial. Els votants són encara poc "avessats" a l'ús de la votació electrònica tot i ser molt més segura que el vot per correu tradicional. Si s'assegura el compliment de les propietats que s'enuncien en l'apartat 2.1 del següent capítol, el vot electrònic és fins i tot més segur que el vot tradicional.

La votació tradicional té tota una sèrie de desavantatges que es podrien superar fàcilment amb la votació electrònica. Com ja s'ha esmentat anteriorment, el desplaçament fins als col·legis i haver de supervisar el procés de votació en són, potser, els més evidents. Però quan es tanca el procés, els vots s'han de recomptar i normalment els encarregats d'aquesta tasca són els mateixos que han supervisat tot el procés anterior, a més del temps que han d'invertir aquests, el fet que el recompte el facin persones fa que sigui possible que hi hagi un error en l'escrutini o fins i tot també hi podria haver errors durant el procés de votació. A més a més, el temps que transcorre des de que es tanca el procés fins que es fan públics els resultats també es podria disminuir amb l'ús de la votació electrònica.

Amb el sistema actual de votació d'introduir el vot dins d'un sobre i aquest dins d'una urna tancada es garanteix la privadesa, tot i que algun dels votants podria patir coacció.

1.2 Propòsit d'aquest treball

Aprofitant els avantatges que ens ofereixen avui en dia les noves tecnologies, en aquest treball, s'intenta dissenyar una plataforma de votació que resolgui tots aquests problemes que es presenten en la votació tradicional i que la seva implantació sigui factible.

Aquest treball forma part d'un projecte de votació electrònica que ha desenvolupat la Universitat de Lleida en col·laboració amb l'empresa *Scytl* dins del programa *Avanza*. Mitjançant beques, en aquest projecte, s'ha col·laborat conjuntament amb Teodoro Andrés Lairla Morlans (estudiant d'*Enginyeria Informàtica*) i Ivan Radigales Creus (estudiant d'*Enginyeria Tècnica en Informàtica de Sistemes*).

La memòria d'aquest projecte està estructurada en 6 capítols. Els dos primers són una petita introducció a la votació electrònica i al perquè d'aquest treball. En el tercer s'expliquen alguns conceptes bàsics necessaris per a poder entendre el treball realitzat en cas de no estar familiaritzat amb la criptografia emprada. Els dos següents expliquen el disseny de la plataforma i la metodologia emprada per a la seva implementació. Al final, s'ha afegit un capítol on es llisten les possibles millores a implementar.

Capítol 2

Votació electrònica

Un sistema de votació electrònica permet als usuaris emetre còmodament un vot a través d'Internet, evitant així les múltiples incomoditats que representa el sistema de votació tradicional, com els desplaçaments fins als col·legis electorals per a poder emetre el vot, el fet que algunes persones han de romandre allí durant tota la jornada electoral per tal de mantenir la correctesa del procediment... Anem a veure els requisits de seguretat que ha de satisfer:

2.1 Propietats d'una votació electrònica segura

Tota votació electrònica ha de garantir una sèrie de propietats de seguretat:

1. Autenticitat: no tothom pot votar, només aquells que consten en el cens.
2. Unicitat: cada votant només pot votar un sol cop.
3. Integritat: ni el contingut dels vots ni el seu recompte total no es poden manipular.
4. Privadesa: cap vot es podrà relacionar amb la persona que l'ha emès.
5. Verificabilitat: tothom qui ho desitgi podrà verificar la validesa dels resultats.
6. Impossibilitat de coacció: el votant no té cap prova del contingut del seu vot.
7. No es rebel·len resultats parcials durant el període de votació.

Mantenir aquests requeriments no és trivial. Generar una interfície web per a que els usuaris puguin enviar els seus vots no és suficient, cal que es compleixin totes i cada una de les propietats anteriors.

La tecnologia del vot electrònic pot incloure targetes perforades, sistemes òptics d'escaneig, sistemes de registre directe (DRE), així com també la transmissió dels vots per telèfon, xarxes locals o bé Internet. Per això els sistemes de votació electrònica es poden classificar en dos grups:

1. Sistemes que substitueixen alguns dels components tradicionals amb processos electrònics.

2. Sistemes que permeten votar a distància mitjançant xarxes de telecomunicacions. (El treball dut a terme en aquest projecte és d'aquest tipus).

2.2 Paradigmes que es tracten actualment

Actualment, els treballs en seguretat criptogràfica centren la seva investigació en els paradigmes següents:

1. **Firma digital cega**: permet a l'usuari obtenir un missatge signat per una entitat sense que aquesta en conegui el contingut. El votant envia el seu vot a una Autoritat. Després que el votant s'identifiqui, l'Autoritat signa el vot i l'envia per un canal segur. Aquest mètode presenta l'avantatge que preserva l'anonimat del votant, tot i que també dóna a l'Autoritat la possibilitat d'enviar vots en nom de persones que s'han abstingut.
   Per entendre fàcilment la idea principal aquest sistema: el votant escriu el seu vot en un paper i a sobre hi grapa paper de calcar i li ho envia a l'Autoritat, aquesta signa sobre el paper de calcar sense veure que hi ha a sota.

2. **Recomptes homomòrfics de vots**: ja es tracti d'homomorfismes additius o bé multiplicatius, s'empren per a recomptar els vots desxifrant un sol cop. Cada votant empra un xifratge homomòrfic per xifrar el seu vot, el servidor suma o multiplica homomòrficament tots els vots xifrats de forma que la puntuació total de cada candidat es pot recuperar amb una única operació de desxifratge.

   - Avantatges:
     - Privadesa
     - Eficiència en el recompte
   - Desavantatges:
     - Inseguretat davant de vots corruptes. Aquest fet obliga a utilitzar proves molt complexes per demostrar que el vot està ben generat. Per la qual cosa només són adients en els casos en que hi hagi pocs candidats o opcions.

3. **Mixing**: el recompte es realitza vot a vot, però prèviament s'ha efectuat una mescla de vots. Cada un dels votants s'autentica i envia el seu vot xifrat. Un cop el servidor ha rebut tots els vots, els rexifra i els barreja de forma que no es pugui saber qui ha emés cada vot. Finalment, es desxifren els vots i es comptabilitzen.

    - Avantatges:
        - Privadesa
        - Seguretat davant de vots corruptes
    - Desavantatges:
        - Proves de correctesa costoses però molt fiables
        - Recompte de vots ineficient

Donat que els avantatges i desavantatges d'aquests dos últims sistemes són complementaris, s'estan realitzant estudis en votació híbrida, combinant ambdós sistemes per tal d'obtenir una major eficiència sense haver de renunciar a la seguretat, tal i com es proposa en [SMPP10].

Capítol 3

Criptografia emprada

Les propietats de seguretat d'una votació electrònica s'aconsegueixen mitjançant l'ús de protocols criptogràfics. En aquest capítol s'expliquen els conceptes teòrics necessaris per tal de poder entendre la criptografia emprada per a desenvolupar aquesta plataforma de votacióó electrònica. Es pot trobar més informació de caire general en [Cab02] i [HPS00]

3.1 Tipus de criptosistemes

Un criptosistema està format per:

- Un conjunt de missatges en clar M
- Un conjunt de missatges xifrats C
- Dos conjunts de claus K_1 i K_2
- Dues famílies de funcions:
  - Funcions de xifratge $\{E_{k_1}\}_{k_1 \in K_1}$ tal que E_{k_1}: $M \longrightarrow C$
  - Funcions de desxifratge $\{D_{k_2}\}_{k_2 \in K_2}$ tal que D_{k_2}: $C \longrightarrow M$

  Si D_{k_2} és la funció de desxifratge corresponent a E_{k_1}, aleshores s'ha de complir que $\forall m \in M$, $D_{k_2}(E_{k_1}(m)) = m$

Depenent de la relació que existeix entre la clau per xifrar k_1 i la clau per desxifrar k_2, els criptosistemes es poden classificar en dos grans grups:

1. Criptosistemes de clau compartida (o de xifrat simètric): a partir de k_1 es pot obtenir fàcilment k_2, i a l'inrevés.
2. Criptosistemes de clau pública (o de xifrat asimètric): és impossible obtenir k_2 a partir de k_1.

Els criptosistemes de clau pública es basen en la idea de l'intercanvi de claus de Diffie i Hellman [DH76] i compleixen les següents quatre propietats:

1. Desxifrar el text xifrat m ens retorna m. Formalment,

$$D(E(m)) = m,$$

   on E és la funció de xifratge i D la de desxifratge.

2. Tant E com D són fàcils de calcular.

3. Revelar públicament E, no revela una forma fàcil de calcular D.

4. Si un missatge m és desxifrat primer i després xifrat, el resultat és m. Formalment,

$$E(D(m)) = m$$

3.2 Criptosistema ElGamal

3.2.1 Funcionament

ElGamal [Elg85] és un criptosistema de clau pública. La seva seguretat rau en l'elevada complexitat de resoldre el problema del logaritme discret (DLP) aplicat a alguns grups (GDLP), és a dir, donat un grup finit cíclic $(G, *)$ d'ordre n, g un generador de G i x un element de G, determinar l'enter a tal que $g^a = g * \cdots * g = x$.

- Paràmetres del criptosistema
  - Es defineix un grup cíclic G d'ordre n
  - Es defineix un generador $g \in G$.
- Generació de claus
  - S'elegeix una clau privada a tal que $a \in [1, n-1]$
  - Es calcula la clau pública $y = g^a$
- Algorisme de xifratge

$$m \xrightarrow{encrypt} c$$

  - S'escull un nombre r a l'atzar, $r \in [1, n-1]$
  - Es calcula g^r
  - S'envia el missatge xifrat c

$$c = (g^r, m \cdot y^r) = (c_1, c_2)$$

- Algorisme de desxifratge

$$c \xrightarrow{decrypt} m$$

  - Es calcula $c_1^a = (g^r)^a$
  - S'obté el missatge en clar m

$$m = \frac{c_2}{c_1} = \frac{c_2}{(g^r)^a} = \frac{m \cdot y^r}{g^{r \cdot a}} = \frac{m \cdot g^{r \cdot a}}{g^{r \cdot a}}$$

3.2.2 Rexifratge ElGamal

Transforma el criptograma de ElGamal en un de totalment diferent, però amb la particularitat de que es pot desxifrar de la mateixa manera que si no s'hagués rexifrat. Amb el mateix missatge original s'obtenen dos valors diferents que no tenen res a veure l'un amb l'altre. Podem rexifrar perquè utilitzem les propietats de l'homomorfisme multiplicatiu de ElGamal.

- Algorisme de rexifratge
  - Partim del missatge xifrat m_1 de la forma tradicional

$$m_1 = m \longrightarrow E(m_1) = c = (g^{r_1}, m \cdot y^{r_1}) = (c_1, c_2)$$

  - Xifrem el missatge m_2=1

$$m_2 = 1 \longrightarrow E(m_2) = c' = (g^{r_2}, y^{r_2}) = (c_1', c_2')$$

  - Es multipliquen tots dos missatges

$$c = E(m_1) \cdot E(m_2) = (g^{r_1} \cdot g^{r_2}, m \cdot y^{r_1} \cdot y^{r_2}) = (g^{r_1+r_2}, m \cdot y^{r_1+r_2}) = (c_1'', c_2'')$$

- Algorisme de desxifratge
  - Es calcula $(g^{r_1+r_2})^a = (c_1'')^a$
  - S'obté el missatge en clar m

$$m = \frac{c_2''}{(c_1'')^a} = \frac{m \cdot (g^a)^{r_1+r_2}}{(g^{r_1+r_2})^a} = \frac{m \cdot g^{a \cdot (r_1+r_2)}}{g^{a \cdot (r_1+r_2)}}$$

3.3 Corbes el·líptiques

Una corba el·líptica E [MMPV07], [Was08] definida sobre un cos finit $\mathbb{F}_p$ ve definida per una equació:

$$y^2 = x^3 + ax + b \tag{3.1}$$

on a, $b \in \mathbb{F}_p$ amb discriminant $4a^3 + 27b^2 \neq 0 \pmod{p}$. Aleshores, els punts $(x,\ y) \in \mathbb{F}_p \times \mathbb{F}_p$ que formen part d'aquesta corba són aquells que satisfan l'equació més el punt de l'infinit $\mathcal{O}$,

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p^2 \mid y^2 = x^3 + ax + b\} \cup \mathcal{O}$$

Amb el mètode de la corda i la tangent es pot definir una operació additiva sobre els punts de la corba, obtenint aleshores que $(E(\mathbb{F}_p), +)$ té estructura de grup abelià. En aquest cas es tracta d'un grup abelià finit ja que es treballa sobre un cos finit. Donat que el problema del logaritme discret és més difícil de resoldre sobre aquest conjunt de punts que sobre el mateix cos finit, la longitud de les claus en criptografia de corbes el·líptiques pot ser més curta mantenint el mateix nivell de seguretat.

3.3.1 Criptosistema ECIES

Per xifrar amb corbes el·líptiques s'ha emprat l'algorisme ECIES (*Elliptic Curve Integrated Encryption Scheme*) [HMV03], que és una variant de l'algorisme de xifratge de clau pública ElGamal.

- Paràmetres del criptosistema: les corbes venen definides per un conjunt $t = \{p, a, b, P, n\}$
  - p (primer) característica del cos $\mathbb{F}_p$
  - a, b coeficients de la corba el·líptica $y^2 = x^3 + ax + b$
  - P punt de la corba
  - n ordre del punt P

  A més, també cal:
  - Elegir la clau privada $d \in \mathbb{Z}$ tal que $d \in [1, n-1]$
  - Calcular la clau pública Q, punt pertanyent a la corba tal que $Q = d \cdot P$
- Algorisme de xifratge

$$m \xrightarrow{encrypt} (R, c)$$

  - Es tria $r \in \mathbb{Z}$ tal que $r \in [1, n-1]$
  - Es calculen els punts $R = r \cdot P$ i $Z = r \cdot Q$.
    Si Z fos $\mathcal{O}$ llavors es recalcularia r.
  - Es calcula la clau compartida k_S, mitjançant una funció de derivació KDF(x_Z, R), on x_Z representa la coordenada X del punt Z.

Existeixen múltiples funcions de derivació, en aquest cas se n'empra una de molt senzilla:

$$k_S = KDF(x_Z, R) = (x_R + y_R + x_Z) \pmod{2^{128}},$$

on x_R i y_R representen les coordenades X i Y, respectivament, del punt R.
D'aquesta forma s'obté una clau k_S de 128 bits.

– Per a xifrar m (text en clar) s'empra el xifratge en flux RC4, que consisteix, bàsicament, en dos algorismes:

1. KSA (Key Scheduling Algorithm): inicialitza un array S de 256 bytes amb valors seqüencials del 0 al 255 i a continuació el permuta amb un algorisme que depèn de k_S creant un nou array K.
2. PRGA (Pseudo-Random Generation Algorithm): genera bytes pseudo-aleatoris a partir de K.

Es realitza l'operació lògica OR byte a byte entre m i els bytes pseudo-aleatoris generats per l'algorisme PRGA i s'obté el text xifrat c.

– Finalment, s'envia el missatge (R, c)

- Algorisme de desxifratge

$$(R, c) \xrightarrow{decrypt} m$$

– Amb la clau privada d es calcula el punt Z

$$d \cdot R = d \cdot r \cdot P = r \cdot d \cdot P = r \cdot Q = Z$$

– Es calcula la clau compartida k_S amb KDF(x_Z, R).

– Per a recuperar m s'empra el mateix algorisme RC4 que per a xifrar.

3.4 RSA

3.4.1 Xifratge i desxifratge

RSA [RSA78] és un criptosistema de clau pública dissenyat el 1978 per Ronald Rivest, Adi Shamir i Leonhard Adleman. La seva seguretat rau en la dificultat de la factorització de nombres enters.

- Paràmetres del criptosistema

– S'escullen dos primers p i q grans.

  - Es considera el grup multiplicatiu $G = \mathbb{Z}_N^*$, on $N = p \cdot q$
  - Es calcula $n = \phi(N)$, on ϕ és la funció d'Euler. Per les propietats de ϕ es té:

$$n = \phi(N) = \phi(p) \cdot \phi(q) = (p-1)(q-1)$$

- Generació de claus
  - S'escull e, tal que mcd(e, n)=1
  - Es calcula d, tal que $e \cdot d \equiv 1 \pmod{n}$

Clau privada	n, d, p, q
Clau pública	$\mathbb{Z}_N^*$, e

Taula 3.1: Clau pública i privada del RSA

- Algorisme de xifratge

$$m \xrightarrow{encrypt} c$$

  - Es pren un missatge $m \in \mathbb{Z}_N^*$
  - Es calcula $c = m^e \pmod{N}$
  - S'envia el missatge xifrat c

$$c = m^e \pmod{N}$$

- Algorisme de desxifratge

$$c \xrightarrow{encrypt} m$$

  - Es calcula $m = c^d \pmod{N}$
  - Pel Teorema d'Euler ($m^{\phi(N)} \equiv 1 \pmod{N}$) és fàcil comprovar que:

$$c^d \equiv (m^e)^d \equiv m^{e \cdot d} \equiv m^{1+\phi(N) \cdot k} \equiv m \cdot (m^{\phi(N)})^k \equiv m \pmod{N}$$

3.4.2 Signatura digital amb RSA

La signatura digital és una eina que permet a un emissor enviar un missatge tal que tothom podrà verificar-ne l'origen. El DNI electrònic empra la signatura de RSA amb claus de 2048 bits, és a dir, el paràmetre N és de 2048 bits.

Per a signar i verificar la signatura s'empren els mateixos paràmetres que per a xifrar i desxifrar amb RSA, però aplicant la propietat 4 dels criptosistemes 3.1, és a dir, en comptes d'elevar el missatge a la clau pública i a continuació a la privada es fa a l'inrevés.

- Generació de la signatura

$$m \xrightarrow{sign} (m, s)$$

  - Es calcula el *hash* del missatge m, $\mathcal{H}(m)$
  - Es calcula la signatura s del *hash*

$$s = \mathcal{H}(m)^d \pmod{N}$$

  - S'envia el parell (m, s)

- Verificació de la signatura
  - Es calcula

$$m' \equiv s^e \pmod{N}$$

  - Es comprova que m'=m

Capítol 4

Visió general de la plataforma

En aquest capítol s'explica en quines parts s'estructura la plataforma i quin és el disseny i funcions de cada una d'aquestes.

4.1 Disseny general

La plataforma de votació electrònica consta de tres mòduls ben diferenciats però que es combinen entre sí tal i com es pot veure en la figura 4.1:

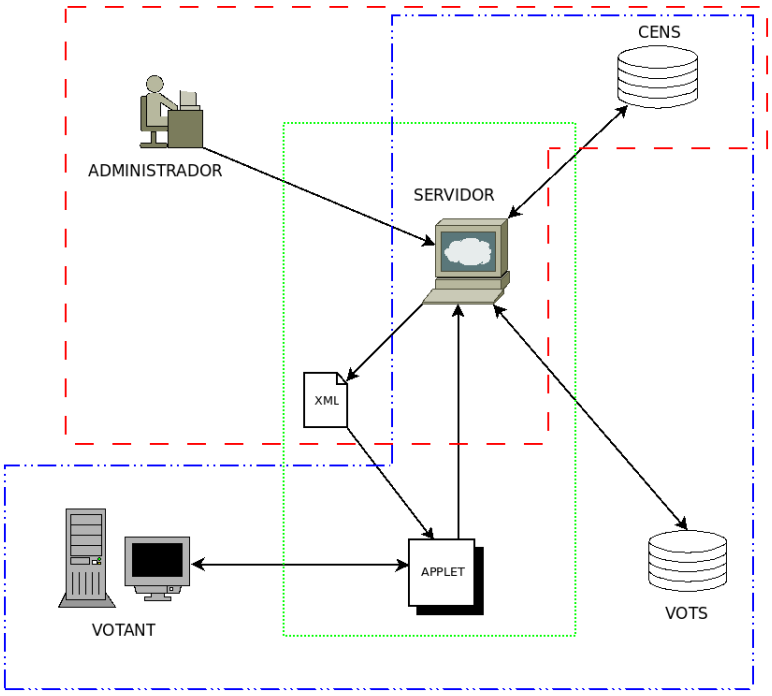


Figura 4.1: Disseny general de la plataforma

1. Mòdul d'administració (- - - -): és el mòdul que empra l'administrador

del sistema per a generar el fitxer XML amb els paràmetres de la votació, és a dir, nom dels candidats o opcions de vot, el nombre de vots i el valor que poden prendre,...

2. Mòdul d'interfície (·····): s'encarrega de llegir el fitxer XML i mitjançant l'*applet* generar la interfície de votació.

3. Mòdul criptogràfic (- ·· - ·· -): el client, a través de l'*applet*, xifra i signa el vot. El servidor, comprova la signatura, desxifra i recompta els vots.

4.2 Funcionalitats

La funció principal és la de poder emetre un vot electrònic que compleixi la major quantitat possible de les característiques abans esmentades 2.1. A continuació es comenten tant les que s'han implementat (en negreta) com les que no:

1. **Autenticitat**: Pot votar tothom ja que no hi ha cap comprovació abans de l'emissió del vot, però només es comptabilitzen els vots d'aquelles persones que es troben en el cens. El servidor s'assegura de la seva identitat mitjançant el certificat digital que s'adjunta al vot emès.

2. Unicitat: Cada votant només pot votar un sol cop ja que a més de comprovar que el votant es troba al cens també es comprova si ha votat anteriorment. Aquesta propietat no s'ha implementat en aquest treball, tot i que sí s'ha afegit a la llista de funcionalitats a implementar en un treball futur.

3. **Integritat**: En el servidor es realitzen dues comprovacions. La primera es realitza abans d'emmagatzemar el vot en el servidor, consistent en comprovar que la signatura sigui correcta i la segona es realitza durant l'obertura de vots, ja que junt amb el vot xifrat amb corbes el·líptiques també s'envia el seu hash.

4. **Privadesa**: cap vot es podrà relacionar amb la persona que l'ha emés ja que abans de desxifrar els vots i mitjançant el rexifratge ElGamal es trenca el vincle que existeix entre el vot i la persona que l'ha emés.

5. **Verificabilitat**: el procediment emprat és públic i tothom que ho vulgui pot comprovar el seu funcionament.

6. Impossibilitat de coacció: la implementació d'aquesta propietat també s'ha deixat per un treball futur. Un sistema que ja està implantat és donar a cada usuari dos *passwords*, un de "bo" per a que el vot es comptabilitzi i un altre de "dolent" per a que no, en cas de que el

votant estigui essent coaccionat per emetre un vot no desitjat s'identifica amb el *password* "dolent" i quan el servidor rep el vot el descarta directament. Més tard, pot emetre el seu vot amb el *password* "bo".

7. **Imparcialitat**: no es revela cap resultat parcial durant el període de votació. Durant el període de votació el servidor comprova la signatura i si és correcta, emmagatzema el vot xifrat.

4.3 Estructura dels mòduls

4.3.1 Mòdul d'administració

Per tal que el client pugui efectuar la seva votació cal una interfície. De la seva gestió se n'encarrega l'administrador. Per tal que l'administrador no hagi de programar una interfície diferent per cada tipus de votació, s'ha buscat un disseny que permeti abastar-los tots. Per això s'han definit els següents paràmetres:

- Nom i descripció de la votació: cadenes de caràcters que defineixen un nom i una descripció i que donen informació al client del què ha de fer per emetre el seu vot.
- Dates d'obertura i de tancament de la votació: atributs del tipus AAAAMMDDhhmm, on:
  - AAAA: representa l'any en quatre xifres.
  - MM: representa el mes en dues xifres, desde Gener 01 fins al Desembre 12.
  - DD: representa el dia en dues xifres, desde 01 fins a 31.
  - hh: representa l'hora entre 00 i 23. Sempre ha d'estar format per dues xifres.
  - mm: representa els minuts entre 00 i 59. Sempre ha d'estar format per dues xifres.
- Vots en blanc permesos?: valor booleà que pren els valors de cert o fals.
- Llista de candidats: llista amb un candidat o més on, per cada candidat, es defineix:
  - Identificador del candidat: valor numèric emprat únicament en el moment d'"empaquetar" el vot.
  - Nom del candidat: cadena de caràcters que representa el candidat. Pot ser el nom de la persona, el partit al que representa,...

  - Descripció del candidat: cadena de caràcters que fa una breu descripció del candidat

- Nombre màxim i mínim de vots que es poden atorgar: es defineixen aquests dos enters per tal de poder abastar un gran nombre de votacions diferents, així per exemple en un concurs en que s'ha d'elegir un únic disseny el nombre màxim de vots i el mínim serien el mateix, 1; en canvi, en una votació en que es pugui votar els dissenys que més agraden, el nombre mínim pot ser 0 (si es permet vots en blanc) o 1 (en cas contrari) i el màxim pot ser donat per una cota.

- Llista de possibles vots per candidat: es defineix un rang o llista que indica els valors possibles que pot prendre un vot. Per exemple, en una votació del tipus Eurovisió es pot votar de l'1 al 10 i el 12. Per això es defineixen quatre variables per cada un dels possibles valors:

  - Identificador: enter que identifica el vot.
  - Nom: cadena de caràcters que descriu el vot.
  - Ús mínim: nombre mínim de cops que es pot emprar aquest valor en un vot.
  - Ús màxim: nombre màxim de cops que es pot emprar aquest valor en un vot.

  Així en la votació del tipus Eurovisió es pot votar 1-12 prenent 1 com a valor màxim i mínim en tots, excepte el valor 11 que no es pot assignar a ningú.

- URL del cens: fitxer XML que conté l'identificador de totes les persones que poden votar, per exemple: el DNI, el *login* de la UdL,...

Tota aquesta informació es desa en un fitxer XML de manera que tant el client (mitjançant un Applet) com el servidor (mitjançant Struts per la configuració web de les eleccions i un programa de Java per al recompte) pugui carregar aquesta informació i entendre de quin tipus de votació es tracta.

4.3.2 Mòdul d'interfície

Aquest mòdul és amb el que interactua directament l'usuari per tal de poder emetre el seu vot. Essencialment, es tracta d'una pàgina HTML l'única funció de la qual és executar un *applet*.

L'*applet* és un programa que s'executa a l'ordinador del client i li permet votar. També s'encarrega de xifrar i signar el vot. Això aporta dos avantatges molt importants:

1. El vot viatja de forma segura. En el moment en que el vot surt de l'ordinador de l'usuari està xifrat i signat, per tant, no viatja mai en clar per la xarxa.

2. La càrrega computacional del xifratge i signatura dels vots té lloc en l'ordinador del votant.

L'*applet* opera de la següent forma:

- L'*applet* llegeix el fitxer XML del servidor per saber quin tipus d'eleccions ha de mostrar i generar així la interfície per a l'usuari.
- Carrega una interfície gràfica amb els candidats i un quadre per seleccionar el vot possible per a cada un dels candidats.
- Quan l'usuari prem el botó d'enviar el vot, comprova que es compleixin les condicions de vots màxims i mínims.
- Es genera un valor numèric que codifica l'elecció feta pel votant.
- Si tot és correcte, crida a les funcions de xifratge i signatura.
- Si tot ha anat bé, envia el vot cap al servidor emprant una connexió TCP via *sockets*.

4.3.3 Mòdul criptogràfic

Tant el servidor com el client empren criptografia per tal que la votació esdevingui segura. Donat que la implementació de la criptografia en la plataforma és l'objecte d'aquest treball, s'explica a continuació i amb tot detall en el capítol 5.

Capítol 5

Mòdul criptogràfic

En aquest capítol s'explicarà amb més detall el disseny i les funcionalitats del mòdul criptogràfic, així com la seva estructura diferenciant les operacions que tenen lloc en l'ordinador del votant i les que executa el servidor.

5.1 Visió general

1. El servidor genera les seves claus públiques i privades necessàries per a que es pugui dur a terme el procés de votació.

2. Quan el votant es connecta a la pàgina on es troba la interfície per votar es carrega en el seu ordinador l'*applet* que s'encarregarà de que el vot viatgi per la xarxa de forma segura.

3. L'*applet* xifra el vot v amb el criptosistema ECIES, en calcula el seu *hash* i ho concatena tot.

$$m = E(v)||h(E(v))$$

4. L'*applet* xifra el vot m amb la clau pública del servidor i el criptosistema ElGamal.

$$c = (g^r, m \cdot y^r)$$

5. L'*applet* signa el vot amb la clau privada del DNI electrònic del votant i envia al servidor el vot, la signatura i el certificat d'autenticació del votant.

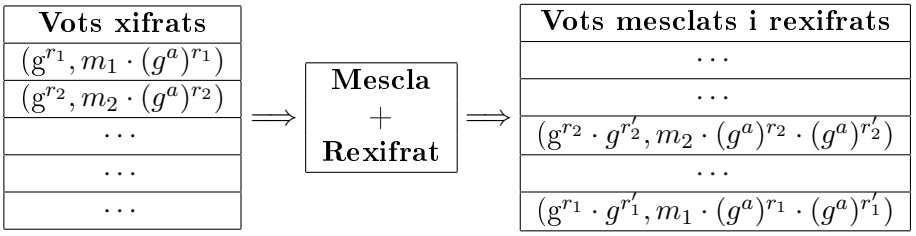
$$vot = (c, sign(c), cert)$$

6. El servidor comprova si la signatura del vot és correcta. Si no ho és, el vot es descarta.

7. El servidor guarda cadascun dels vots correctes (sense desxifra-los), així com la signatura del vot i el certificat del votant.

Vots rebuts		
$Vot_1 = (g^{r_1}, m_1 \cdot (g^a)^{r_1})$	$Sign(Vot_1)$	$Cert_1$
$Vot_2 = (g^{r_2}, m_2 \cdot (g^a)^{r_2})$	$Sign(Vot_2)$	$Cert_2$
$\cdots$	$\cdots$	$\cdots$

8. Per a no saber qui ha votat què, es barregen els vots. Si ho deixem així, es podria relacionar el vot amb el seu votant, així que al mateix temps que es barregen els vots també s'emmascaren. Per tant, barregem i rexifrem.



9. Es desxifren els vots i es comptabilitzen.

Així, doncs, tot el procés criptogràfic es podria resumir en la taula 5.1, on es mostra qui executa l'acció (el votant o el servidor), quin programa s'executa, quines accions tenen lloc i en quin apartat d'aquest treball es detallen:

Servidor	*Initial_Server*	Generació de les claus	Generació de les claus 5.2
Votant	*Applet_Election*	Xifrar amb corbes el·líptiques Xifrar amb ElGamal	Generació del vot 5.3
		Signar	Signatura 5.4
Servidor	*Server_Election*	Rebre i emmagatzemar el vots Comprovar la signatura	Emmagatzemar vots 5.5
	Final_Server	Mescla + Rexifratge	Mescla 5.6
		Desxifrar Recompte de vots	Obertura dels vots 5.7

Taula 5.1: Estructuració del mòdul criptogràfic

5.2 Generació de les claus

Abans de començar el procés de votació, el servidor obre el programa *Initial_Server*, tria la longitud de les claus i executa el codi.

Per dotar el codi de persistència, es guarden totes les claus, públiques i privades, en fitxers. Totes les claus públiques es guarden en el fitxer *pubKeys.xml*, en canvi, les claus privades es guarden en dos fitxers separats, *privKeyECIES.xml* i *privKeyElGamal.xml*. Aquesta decisió s'ha pres només per motius de disseny i implementació, ja que per a les claus privades es guarda el mateix paràmetre (la clau privada), i per tant, es pot utilitzar el mateix codi. En canvi, per a les claus públiques s'han de desar diferents paràmetres depenent del criptosistema, tal i com es pot apreciar en la figura 5.1.

Figura 5.1: Exemple de sortida de codi: generació de les claus

Tot i que guardar les claus privades en fitxers disminueix la seguretat del programa, és necessari, ja que si, per exemple, marxés la llum i només s'haguessin guardat les claus públiques, no es podria arribar a desxifrar el vot. Per augmentar-ne la seguretat, aquestes claus es podrien guardar xifrades o dintre d'una tarja intel·ligent.

Per defecte, el programa treballa amb unes claus de longitud de 160 bits amb corbes el·líptiques i de 1024 bits amb ElGamal. Per evitar càlculs i

disminuir el temps de processament, s'han pre-generat una sèrie de claus de ElGamal de longitud de 1024 i 2048 bits que es poden recuperar des d'un fitxer.

Seguint les recomanacions estàndard criptogràfiques del NIST (*National Institute of Standards and Technology*) [NIST], per tal que les corbes el·líptiques siguin segures han de complir una sèrie de requeriments:

- S'ha de complir que l'ordre del grup $E(\mathbb{F}_p)$ sigui de la forma $f \cdot q$, on q és un primer i f un enter petit [Kov98].

- La corba ha de ser no supersingular i no anòmala. Les corbes supersingulars són aquelles que tenen cardinal $p+1$. Per aquests tipus de corbes l'atac MOV [MOV96] transforma el ECDLP sobre $E(\mathbb{F}_p)$ en el DLP sobre $\mathbb{F}^*_{p^k}$, amb $k \leq 6$. Les corbes anòmales són aquelles amb cardinal p i que, tot i que resisteixen l'atac MOV, existeix un algoritme polinomial que resol el DLP sobre aquest grup de punts.

El càlcul del nombre de punts d'una corba el·líptica es pot efectuar amb l'algoritme Schoof [Sch85] amb una complexitat polinomial $O(\log^8 p)$, però la seva implementació esdevé infactible a mesura que p augmenta.

Per aquest motiu, en aquest treball no s'han calculat els paràmetres sinó que s'ha treballat amb les corbes recomanades [SECG2] pel NIST.

- Corbes definides sobre cossos finits $\mathbb{F}_p$,
  $\lceil \log_2 p \rceil \in \{112, 128, 160, 192, 224, 256, 384, 521\}$

- Poden estar associades a una corba de Koblitz o elegides a l'atzar. Una corba de Koblitz és una corba el·líptica definida sobre $\mathbb{F}_{2^m}$ amb $m \geq 1$ i $a, b \in \{0, 1\}$.

- N'hi ha 15 de diferents:
  i.e: $secp160k1$, $secp160r1$, $secp160r2$, on k indica que està associada a una corba de Koblitz i r que ha estat elegida a l'atzar.

- Venen definides per un conjunt $t = \{p, a, b, P, n\}$

  - p (primer) característica del cos $\mathbb{F}_p$
  - a, b coeficients de la corba el·líptica $y^2 = x^3 + ax + b$
  - $P = (X_P, Y_P)$ punt de la corba
  - n ordre del punt P

El punt P no es dóna en forma de les seves dues coordenades, sinó en forma compacta. Per exemple, la corba *secp128r1* té $P = 03\ 161FF752\ 8B899B2D\ 0C28607C\ A52C5B86_{16}$. Per obtenir les coordenades a partir d'aquest nombre cal aplicar l'algorisme *Octet-String-to-Elliptic-Curve-Point Conversion* [SECG1]. Els dos primers bits Y'_P representen la coordenada Y_P i la resta, la X_P.

Si $Y'_P = 02_{16}$, aleshores es pren $y_P = 0$. Si $Y'_P = 03_{16}$, $y_P = 1$. Altrament, és un valor invàlid, ja que es treballa mòdul 2. Es calcula Y''_P substituint X_P en l'equació de la corba i calculant-ne el residu quadràtic mitjançant l'algorisme del Símbol de Jacobi [MOV96]. S'en pren el resultat mòdul 2 i es compara amb y_P. Si coincideixen, $Y_P = Y''_P$. Altrament, $Y_P = -Y''_P$.

Un cop s'ha elegit una de les corbes i s'han obtingut tots els paràmetres, també cal:

- Elegir la clau privada $d \in \mathbb{Z}$ tal que $d \in [1, n-1]$
- Calcular la clau pública Q, punt pertanyent a la corba tal que $Q = d \cdot P$

5.3 Generació del vot

Quan l'usuari es connecta a la pàgina per a votar, està executant el programa *Applet_Election*. El primer que li apareix és una finestra preguntant-li si accepta executar l'*applet* en el seu ordenador, tal i com es mostra en la figura 5.2 . És necessari que l'usuari accepti, altrament no podrà votar.

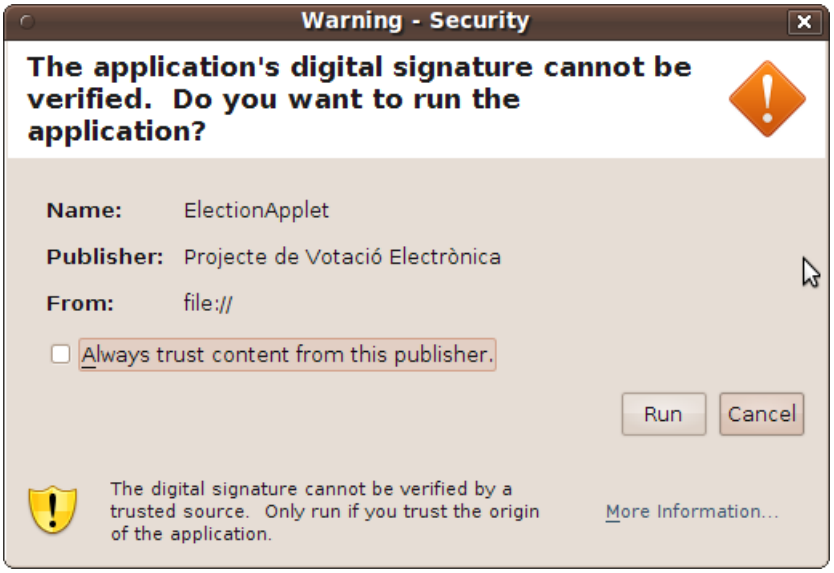


Figura 5.2: Warning

Un cop ha acceptat, l'usuari pot accedir a l'*applet*. Mitjançant els mòduls d'administrador i d'interfície que complementen aquest treball, un mateix *applet* serveix per a diferents tipus de votacions, tal i com ja s'ha explicat anteriorment. En la figura 5.3 es mostra el tipus de votació de prova que s'ha emprat durant la implementació d'aquest projecte. El votant emet el seu vot dins de l'*applet*. En el cas que es tracti d'un vot múltiple, és a dir, si cal emetre més d'un vot, per exemple en una votació tipus Eurovisió, el vot s'"encapsula" per tal de ser tractat com un únic nombre. A partir d'aquest moment, s'entendrà com a "vot" el nombre final que envia l'*applet* cap al servidor i com "puntuació" cada una de les valoracions que el votant dóna a cada un dels candidats.

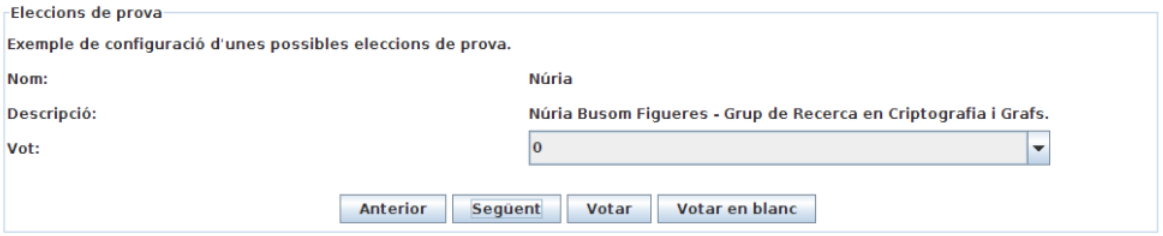


Figura 5.3: Applet de prova

Per tal d'"encapsular" les puntuacions i de poder-les diferenciar s'aplica la fórmula 5.1.

$$vot = [\sum_{i=0}^{n-1}[(p_i + 1) * (n + 1)] * b^i] + 1, \tag{5.1}$$

on p_i és la puntuació, n és el nombre total de candidats i b el nombre màxim de puntuacions que es poden emetre.

El vot en blanc es representa donant un -1 a tots els candidats, aleshores $p_i + 1 = -1 + 1 = 0$. No es pot xifrar un 0 amb ElGamal, per això a la fòrmula s'afegeix el $+1$ al final.

Un cop obtingut el *vot*, l'*applet* el xifra amb l'algorisme ECIES explicat en l'apartat 3.3.1. A continuació, es pren el resultat del xifratge anterior i es xifra amb ElGamal (veure 3.2.1). D'aquesta manera el vot v no viatja mai en clar per la xarxa.

5.4 Signatura del vot

Quan el votant prem el botó per enviar el seu vot v, li apareix una finestra del navegador (figura 5.4) que li demana el seu DNI electrònic en el lector i introdueixi la seva contrasenya. També caldria especificar la ruta on es troba la llibreria $PKCS\#11$, tot i que només es necessari la primera vegada, ja que les següents se'n recorda. En un sistema UNIX es pot localitzar fàcilment fent *$ locate opensc-pkcs11.so*.

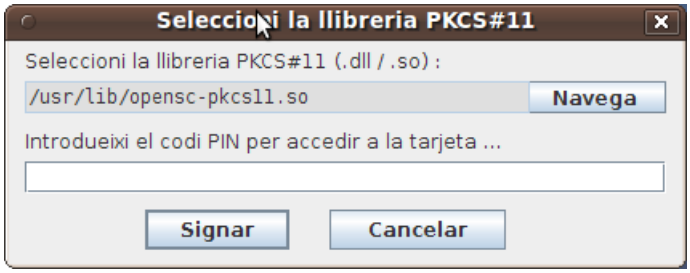


Figura 5.4: Finestra del navegador per signar el vot

Els certificats digitals del DNI electrònic ja ens proporcionen tots els paràmetres necessaris (veure 3.4.1):

- Clau pública: N, e
- Clau privada: d, p, q

L'*applet* signa el vot v (veure 3.4.2) amb les llibreries de Java ($java.security.*$) i envia al servidor a través d'un socket una estructura de dades $V = (v, sign(v), cert)$ que contè el vot v, la signatura $sign(v)$ i el certificat d'autenticació $cert$ del votant.

5.5 Emmagatzemar vots

Abans que s'obri el procés de votació, cal que el servidor executi el programa $Server\_Election$ que és l'encarregat de rebre els vots. Aquest codi, obre un *socket* al port 8085 i es queda escoltant durant tot el procés electoral.

En arribar el vot V al servidor, es llegeix el certificat i si el votant es troba en el cens es comprova la correctesa de la signatura. Si aquesta no és correcta, o bé si el votant no es troba en el cens, el vot es descarta. En cas contrari, el vot V s'emmagatzema en un fitxer en el servidor fins al tancament de les urnes.

A continuació es pot veure un exemple de vot emmagatzemat.

```
<ballot>
<cert>[
[
  Version: V3
  Subject: CN="BUSOM FIGUERES, NURIA (AUTENTICACIÓN)", GIVENNAME=NURIA,
  SURNAME=BUSOM, SERIALNUMBER=43747203F, C=ES
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

  Key:  Sun RSA public key, 2048 bits
  modulus: 22742956879957208376961183923482743199343907168784259859646
  59416722700665888437557561409423742610119673258450879350919861290393
  22254628927423612861059599118634096115753108956112823078536869897914
  94511578488756331750084555355982977668775584184509510926106883528088
  89390167004327236728071588794458183064292147600509101821178628775890
  74931731858776686129118203339609214142298105736468923378545702852770
  68301976626665281789025507626804437968786797447780173420684038627457
  11556741327707588857000513561084169474660843325493179988129288442582
  66239728277084171444160529839920937728300949198657326266003785876787
  15074704325959
  public exponent: 65537
  Validity: [From: Thu Sep 24 08:50:51 CEST 2009,
              To: Sat Mar 24 09:20:49 CET 2012]
  Issuer: CN=AC DNIE 002, OU=DNIE, O=DIRECCION GENERAL DE LA POLICIA,
  C=ES SerialNumber: [4498a828]

Certificate Extensions: 10
[1]: ObjectId: 2.5.29.9 Criticality=false
Extension unknown: DER encoded OCTET string =
0000: 04 21 30 1F 30 1D 06 08   2B 06 01 05 05 07 09 01  .!0.0...+.......
0010: 31 11 18 0F 31 39 38 33   30 31 30 39 31 32 30 30  1...198301091200
0020: 30 30 5A                                           00Z


[2]: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 3A A6 89 EC 15 E8 24 64   71 E0 25 7E C9 B1 62 31  :.....dq.%...b1
0010: 07 E9 06 A2                                        ....
]

]

[3]: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [
  [
   accessMethod: 1.3.6.1.5.5.7.48.1
   accessLocation: URIName: http://ocsp.dnie.es,
   accessMethod: 1.3.6.1.5.5.7.48.2
```

```
   accessLocation: URIName: http://www.dnie.es/certs/ACRaiz.crt]
]

[4]: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 66 5D EC AA 0E D0 94 8A   87 A6 16 BE 54 56 70 BF  f]..........TVp.
0010: 1D B4 09 51                                        ...Q
]
]

[5]: ObjectId: 1.3.6.1.5.5.7.1.3 Criticality=false
Extension unknown: DER encoded OCTET string =
0000: 04 16 30 14 30 08 06 06   04 00 8E 46 01 01 30 08  ..0.0......F..0.
0010: 06 06 04 00 8E 46 01 04                            .....F..


[6]: ObjectId: 1.3.6.1.5.5.7.1.2 Criticality=false
Extension unknown: DER encoded OCTET string =
0000: 04 81 E3 30 81 E0 30 32   02 01 01 30 0B 06 09 60  ...0..02...0...‘
0010: 86 48 01 65 03 04 02 01   04 20 7D 05 AC 4C 8B 8F  .H.e..... ...L..
0020: 65 05 43 29 74 DD 76 68   68 FC 3E 21 45 D1 4B 83  e.C)t.vhh.&gt;!E.K.
0030: 07 C5 E5 30 94 53 82 05   40 EC 30 32 02 01 00 30  ...0.S..@.02...0
0040: 0B 06 09 60 86 48 01 65   03 04 02 01 04 20 6F 87  ...‘.H.e..... o.
0050: 3C BB 0C F2 9D 08 E6 90   A0 4F 2E FE 9C F4 70 13  &lt;........O....p.
0060: 63 F2 71 DA C7 76 FD 48   F1 73 5C FD B4 93 30 3A  c.q..v.H.s\...0:
0070: 06 09 60 85 54 01 02 02   04 02 01 30 0B 06 09 60  ..‘.T......0...‘
0080: 86 48 01 65 03 04 02 01   04 20 30 88 A6 8B 62 08  .H.e..... 0...b.
0090: FD 23 5F 57 CA C3 BF B3   4D A5 6D 4B 6A 29 8B D3  .#_W....M.mKj)..
00A0: 8C 5F 71 6C 05 86 91 8C   D9 BB 30 3A 06 09 60 85  ._ql......0:..‘.
00B0: 54 01 02 02 04 02 06 30   0B 06 09 60 86 48 01 65  T......0...‘.H.e
00C0: 03 04 02 01 04 20 B5 AD   F4 EA EB 02 6E 9D C8 78  ..... ......n..x
00D0: 7E 00 4C 7F DE 35 06 D4   F5 EE BA 6A 51 8B 44 F9  ..L..5.....jQ.D.
00E0: F9 9A D4 DC 1D 97                                  ......


[7]: ObjectId: 2.5.29.32 Criticality=false
CertificatePolicies [
  [CertificatePolicyId: [2.16.724.1.2.2.2.4]
[PolicyQualifierInfo: [
  qualifierID: 1.3.6.1.5.5.7.2.1
  qualifier: 0000: 16 16 68 74 74 70 3A 2F   2F 77 77 77 2E 64 6E 69
             ..http://www.dnie.es/dpc
             0010: 65 2E 65 73 2F 64 70 63
]]  ]
]

[8]: ObjectId: 2.16.724.1.2.2.4.1 Criticality=false
Extension unknown: DER encoded OCTET string =
```

```
0000: 04 36 30 34 30 32 02 01   02 30 0B 06 09 60 86 48  .60402...0...‘.H
0010: 01 65 03 04 02 01 04 20   BE BC 95 D6 63 C6 2C 3A  .e..... ....c.,:
0020: 20 E0 E4 26 CA A0 EE 9A   38 61 C0 78 17 D9 D3 20   ..&amp;....8a.x...
0030: 8F 5D C7 B4 2B DA 68 85                            .]..+.h.


[9]: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:false
  PathLen: undefined
]

[10]: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  DigitalSignature
]

]
  Algorithm: [SHA1withRSA]
  Signature:
0000: 01 1D 22 DB BF 7B 25 49   01 17 23 6D 7F 2F B9 3C  .."...%I..#m./.&lt;
0010: C6 14 1C 64 21 26 50 5C   F5 B4 87 B3 45 F5 2F 94  ...d!&amp;P\....E./.
0020: 21 A3 1B 02 36 21 73 D3   76 90 62 C2 66 29 95 68  !...6!s.v.b.f).h
0030: 9D 78 07 72 B9 78 75 83   A1 DD 52 FA 98 E6 23 42  .x.r.xu...R...#B
0040: EB 2B 2A 1A 1C 1F 07 3D   1E 98 83 84 A7 F8 87 2B  .+*....=.......+
0050: 3B 55 8E E4 A3 43 D3 C1   2C 19 6D A2 09 8E 09 EE  ;U...C..,.m.....
0060: C6 3A 9D 48 EE E4 F1 A1   F9 43 71 30 26 98 74 5B  .:.H.....Cq0&amp;.t[
0070: AA D1 C9 69 69 C4 99 13   8B 78 47 A2 64 E1 30 A9  ...ii....xG.d.0.
0080: 15 D9 19 79 20 11 3D 0E   11 7D 2A B8 B9 32 5E 79  ...y .=...*..2^y
0090: 5E 9D 33 BD F3 DA 5C 9F   F6 27 88 9B EF 11 DB B3  ^.3...\..'......
00A0: FF 66 86 C8 23 28 73 39   39 39 54 BC 09 C5 7F 05  .f..#(s999T.....
00B0: 4B 41 C4 D9 4A 2E 70 9E   0D 4E 95 EF 98 C0 14 05  KA..J.p..N......
00C0: AF C9 9C 46 00 CB 69 E0   16 44 9C 0F C3 AD BF 2E  ...F..i..D......
00D0: 57 0A A7 B4 37 97 06 98   FA 4C 59 A7 DB 43 38 42  W...7....LY..C8B
00E0: 57 01 30 23 89 74 F3 58   2B BB 55 B4 89 3A 70 4F  W.0#.t.X+.U..:pO
00F0: 00 4B 4E 69 E3 9F 4F CC   CF 69 88 BB D2 42 0A 84  .KNi..O..i...B..

]</cert>
<sign>ijL1qq1EiFxBTaZnauJOQaKLkTn4/oCnAIBh3rAY7dwIT+6niuiknSWjksx+R7ieRWY6O/
tDjzUDZaBhTIL/iPhHOAPVxo7lN3ORediBgN3cwJ11c98nOMqdjbMR5TNvcndkRTwiWI8vMTQu1w
FqZM+rCgv6PaM4iiE9VG/M3apkTJQBU7KLfqkIvktNpYBwhJ8R6t/fm0F8VxHguUieDRpogsJUKL
3lMDJP/tiWKx3VDSHRXgCxgBE1lPV6rx7swj/uyGPJ7l79LecP6ACodUp1T2orZLsw7BslQp7f3s
+wuglcJr4KA5DwFk7zgwOCdqw/Olsag8begQWJhv7WYQ==</sign>
<vote>
<g>32189668408671759899375852331797829076567745471021005973182374615862417305
44379580657458349969127640791685360310141756381531402766750110027512705325758
53655219702604905541567946143554432022194878669568496121030135081080916473312
31571826498286592742365126816204648252263996713245795922240836001786138956894
881</g>
```

<mg>29921405454069032665067773309455915036536668979521598735884030672102337269789759830988476360509414590174604052944657523686881978375855122274499889921770206356423797504679826515154022809326313053567698702725600575880426757967755486764671419543068121752456990874099475218653826764138900543898607908707167675070701</mg>
</vote>
</ballot>

5.6 Mescla

Un cop s'han tancat les urnes, es recuperen les claus per restablir els criptosistemes. També es llegeixen els vots xifrats dels fitxers. A continuació, es llegeix cada un dels vots $V = (v, sign(v), cert)$ i es guarden els v en el mateix ordre en que s'han rebut en una nova estructura de dades (un array). A continuació es rexifren amb el criptosistema ElGamal (veure 3.2.2) i es guarden en una posició totalment aleatòria d'una altra estructura de dades idèntica a l'anterior. Això permetrà que no es pugui relacionar el vot desxifrat amb els vots v que es guarden junt amb el certificat del votant.

5.7 Obertura dels vots

Un cop tots els vots han estat rexifrats i mesclats, es llegeixen un a un i es desxifren. Donat que cada vot ha estat xifrat amb diferents criptosistemes, cal desxifrar en l'ordre invers en que s'ha xifrat.

Per les propietats homomòrfiques del criptosistema ElGamal es pot desxifrar alhora el xifratge i rexifratge ElGamal amb una sola operació (veure 3.2.1). El resultat és la concatenació del vot xifrat amb corbes el·líptiques i del seu *hash*.

Es comprova que el *hash* sigui correcte. Si no ho és, es descarta el vot. En aquest punt, també s'hauria de comprovar la correctesa del *mixing* proposada en [SMPP10], tot i que per motius de temps, finalment, no s'ha implementat. El fet d'haver xifrat amb dos criptosistemes, ens garantitza que no es rebel·len resultats en cas que la votació hagi estat alterada, doncs aquesta comprovació és prèvia al segon desxifrat dels vots.

Si tot és correcte, es desxifra amb l'algorisme ECIES i s'obté M que és "l'encapsulació" de cada una de les puntuacions que el votant ha emés.

Finalment, el servidor "desencapsula" M i comptabilitza els vots.

Capítol 6

Treball futur

En un projecte futur, tot seguint la línia d'aquest treball es podrien incorporar i desenvolupar tota una sèrie d'aspectes:

- Un cop l'usuari ha afegit el seu certificat i ha premut el botó per enviar el vot, caldria afegir una nova pàgina informant a l'usuari de com ha anat el procés. En cas d'èxit, es podria calcular un *hash* del vot amb alguna dada personal del votant i mostrar-ho per pantalla, això serviria de rebut. En cas de que alguna cosa hagués anat malament, també caldria informar-ne l'usuari tot indicant-li el motiu.

- Comprovar que el certificat públic enviat està signat per una entitat certificadora vàlida.

- Millorar l'aspecte visual de l'*applet*. Actualitzar l'aspecte actual en forma de llista desplegable a algun altre sistema que permetés a l'usuari veure d'un sol cop d'ull tots els candidats. També es podria optar per afegir alguna icona o quelcom que representés el tipus de votació que s'està realitzat, així com afegir alguna fulla d'estil.

- Per dotar d'unicitat la votació, caldria ficar marca en el cens quan algú ha votat de forma que no pogués votar més d'un cop.

- Fer l'*applet* compatible amb la versió 1.4.7 de l'opensc del DNI electrònic, ja que actualment només funciona amb la versió 1.4.6. i això dóna problemes en alguns sistemes operatius i/o distribucions que empren la versió més nova, per exemple, Fedora 12 i superiors.

- Implementar altres criptosistemes i permetre a l'administrador triar quin vol emprar.

- Implementar un sistema anticoacció.

- Acabar d'implementar el sistema SMPP [SMPP10] que s'ha emprat només parcialment en aquest treball. Més concretament, caldria implementar una prova de correctesa, per tal de verificar el *mixing*.
- Afegir una pàgina de *login* prèvia a l'*applet*, cosa que disminuiria part de la feina del servidor. Ara, s'envien tots els vots, tot i que, només s'emmagatzemen els dels usuaris que es troben al cens. Amb aquesta senzilla millora, només podrien votar aquells que es troben en el cens.
- Cal dissenyar una pàgina final en la que es mostren els resultats finals, per tal que els usuaris puguin buscar el seu rebut i comprovar que el seu vot s'ha tingut en compte.

Bibliografia

[Cab02] Caballero, P., *Introducción a la Criptografía*. Ed Ra-Ma, 2002.

[DH76] Diffie, W., Hellman, M. New directions in Cryptography. IEE Trans. Inform. Theory IT-22. 1976.

[Elg85] ElGamal, T., *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory, 1985.

[HMV03] Hankerson, D., Menezes, A., Vanstone, S., Guide to Elliptic Curve Cryptography. Ed. Springer, 2003.

[HPS00] Hoffstein, J., Pipher, J., Silverman, J.H., *An introduction to Mathematical Cryptography*. Ed Springer, 2000.

[Kov98] Koblitz, N., *Algebraic aspects of cryptography*. Ed. Springer, 1998.

[MOV91] Menezes, A., Okamoto, T., Vanstone, S., *Reducing elliptic curve logarithms to logarithms in a finite field*. Annual ACM Symposium on Theory of Computing, 1991.

[MOV96] Menezes, A., van Oorschot, P., Vanstone, S., *Handbook of Applied Cryptography*. CRC Press, 1996.

[MMPV07] Miret, J.M., Moreno, R., Pujolàs, J., Valls, M., *Algorithms and cryptographic protocols using elliptic curves*. Contributions to science, 3(4): 481-491, 2007.

[NIST] National Institute of Standards and Technology, *Digital Signature Standard*, FIPS PUB 186-2. 2000.

[Riv92] Rivest, R.L., *The MD5 Message-Digest Algorithm*, RFC1321, MIT LCS and RSA Data Security, Inc, 1992.

[RSA78] Rivest, R.L., Shamir A., Adleman, R.L., *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, vol 21 (2), 120-126, 1978.

[Sch85] Schoof, R., *Elliptic curves over finite fields and the computation of square roots mod p.* Mathematics of computation 44, 1985.

[SMPP10] Sebé, F., Miret, J.M., Pujolàs, J., Puiggali, J., *Simple and efficient hash-based verifiable mixing for remote electronic voting.* Computer Communications, vol 33, num 6. 2010.

[SECG1] Standards for Efficient Cryptography Group: Secure Elliptic Curves. Elliptic Curve Cryptography version 2.0, disponible a www.secg.org

[SECG2] Standards for Efficient Cryptography Group: Secure Elliptic Curves. Recommended Elliptic Curve Domain Parameters version 2.0, disponible a www.secg.org

[Sta10] Stallings, W., *Network Security Essentials: Applications and Standards.* Prentice Hall, 2010.

[Was08] Washington, L. C., *Elliptic Curves: Number Theory and Cryptography.* CRC Press, 2008.