Treball de Final de Màster

Fast and private web search using android devices

Pau Rodríguez Esmerats
Màster Interuniversitari en Seguretat de les TIC

Directors:
Jordi Castellà Roca
Alexandre Viejo Galicia
Coordinador:
Francesc Serratosa Casanelles

10 de gener de 2013

Llicència del TFM CreativeCommons by-sa

CREATIVE COMMONS CORPORATION NO ÉS UN BUFET D'ADVOCATS I NO PROPORCIONA SERVEIS JURÍDICS. LA DISTRIBUCIÓ D'AQUESTA LLICÈNCIA NO CREA CAP RELACIÓ ADVOCAT-CLIENT. CREATIVE COMMONS PROPORCIONA AQUESTA INFORMACIÓ TAL QUAL. CREATIVE COMMONS NO OFEREIX CAP GARANTIA RESPECTE A LA INFORMACIÓ FACILITADA, NI ASSUMEIX CAP RESPONSABILITAT PELS POSSIBLES DANYS PRODUÏTS PER L'ÚS QUE SE'N PUGUI FER.

Llicència

L'OBRA O LA PRESTACIÓ TAL COM ES DEFINEIX MÉS ENDAVANT, S'OFEREIX SEGONS ELS TERMES D'AQUESTA LLICÈNCIA PÚBLICA DE CREATIVE COMMONS (D'ARA ENDAVANT CCPL O LLICÈNCIA). L'OBRA O LA PRESTACIÓ ESTÀ PROTEGIDA PER LA LLEI ESPANYOLA DE PROPIETAT INTEL·LECTUAL O QUALSEVOL ALTRA NORMA APLICABLE. QUEDA PROHIBIT QUALSEVOL ÚS DE L'OBRA O LA PRESTACIÓ DIFERENT DEL QUE AUTORITZI AQUESTA LLICÈNCIA O DISPOSIN LES LLEIS DE PROPIETAT INTEL·LECTUAL. L'EXERCICI DE QUALSEVOL DRET SOBRE L'OBRA O LA PRESTACIÓ COMPORTA QUE ACCEPTEU LES LIMITACIONS I OBLIGACIONS D'AQUESTA LLICÈNCIA I HI CONSENTIU, SENSE PERJUDICI DE LA NECESSITAT DE CONSENTIMENT EXPRÉS EN CAS DE VIOLACIÓ PRÈVIA DE LES SEVES CONDICIONS. EL LLICENCIADOR US CONCEDEIX ELS DRETS CONTINGUTS EN AQUESTA LLICÈNCIA, SEMPRE QUE N'ACCEPTEU LES CONDICIONS QUE HI SÓN PRESENTS.

1. **Definicions**

   (a) **. L'obra és la creació literària, artística o científica oferta en els termes d'aquesta llicència.**

   (b) **. En aquesta llicència es considera una prestació qualsevol interpretació, execució, fonograma, enregistrament audiovisual, emissió o transmissió, mera fotografia o qualsevol altre objecte protegit per la legislació de propietat intel·lectual vigent aplicable.**

(c) . L'aplicació d'aquesta llicència a una col·lecció (definida més endavant) afectarà únicament l'estructura com a forma d'expressió de la selecció o disposició dels continguts, i no es farà extensiva a aquests. En aquest cas la col·lecció tindrà la consideració d'obra a efectes d'aquesta llicència.

(d) . El titular originari és:

i. . En el cas d'una obra literària, artística o científica, la persona natural o el grup de persones que l'ha creada.

ii. . En el cas d'una obra col·lectiva, la persona que l'editi i el divulgui amb el seu nom, llevat que hi hagi un pacte contrari.

iii. . En el cas d'una interpretació o execució, l'actor, cantant, músic, o qualsevol altra persona que representi, canti, llegeixi, reciti, interpreti o executi de qualsevulla manera una obra.

iv. . En el cas d'un fonograma, el productor fonogràfic, és a dir, la persona natural o jurídica sota la iniciativa i responsabilitat de la qual es fa per primera vegada una fixació exclusivament sonora de l'execució d'una obra o d'altres sons.

v. . En el cas d'un enregistrament audiovisual, el productor de l'enregistrament, és a dir, la persona natural o jurídica que tingui la iniciativa i assumeixi la responsabilitat de les fixacions d'un pla o una seqüència d'imatges, amb o sense so.

vi. . En el cas d'una emissió o una transmissió, l'entitat de radiodifusió.

vii. . En el cas d'una mera fotografia, aquella persona que l'hagi realitzada.

viii. . En el cas d'altres objectes protegits per la legislació de la propietat intel·lectual vigent, la persona que aquesta indiqui.

(e) . Es consideraran obres derivades aquelles obres creades a partir de l'objecte d'aquesta llicència com, per exemple, les traduccions i adaptacions; les revisions, actualitzacions i anotacions; els compendis, resums i extractes; els arranjaments musicals, i en general qualsevol transformació d'una obra literària, artística o científica. Per aclarir dubtes, si l'obra consisteix en una composició musical o

en un enregistrament de sons, la sincronització temporal de l'obra amb una imatge en moviment (synching) serà considerada com una obra derivada als efectes d'aquesta llicència.

(f) . Tindran la consideració de col·leccions la recopilació d'obres alienes, de dades o d'altres elements independents com ara les antologies i les bases de dades que per la selecció o disposició de llurs continguts constitueixin creacions intel·lectuals. La mera incorporació d'una obra en una col·lecció no donarà lloc a una obra derivada a efectes d'aquesta llicència.

(g) . El llicenciador és la persona o l'entitat que ofereix l'obra o la prestació subjecta als termes d'aquesta llicència i que en concedeix els drets d'explotació conforme al que s'hi disposa.

(h) . Vós sou la persona o l'entitat que exerceix els drets concedits mitjançant aquesta llicència i que prèviament no ha violat els termes d'aquesta llicència respecte a l'obra o la prestació, o que ha rebut un permís exprés del llicenciador per exercir els drets concedits mitjançant aquesta llicència malgrat una violació anterior.

(i) . La transformació d'una obra comprèn la traducció, l'adaptació i qualsevol altra modificació de la qual derivi una obra diferent. La creació resultant de la transformació d'una obra tindrà la consideració d'obra derivada.

(j) . S'entén per reproducció la fixació directa o indirecta, provisional o permanent, per qualsevol mitjà i en qualsevol forma, de tota l'obra o la prestació o de part d'ella, que en permeti la comunicació o l'obtenció de còpies.

(k) . S'entén per distribució el fet de posar a disposició del públic l'original o les còpies de l'obra o la prestació mitjançant la venda, el lloguer, el préstec o qualsevol altre sistema.

(l) . S'entén per comunicació pública qualsevol acte mitjançant el qual una pluralitat de persones, que no pertanyin a l'àmbit domèstic de qui la duu a terme, pot tenir accés a l'obra o la prestació sense que se'ls hagi fet una distribució prèvia d'exemplars. Es considera comunicació pública el fet de posar l'obra o la prestació a disposició del públic

per procediments electrònics amb fil o sense fil de manera que qualsevol persona pugui accedir-hi des d'on vulgui i quan vulgui.

(m) . L'explotació de l'obra o la prestació comprèn la reproducció, la distribució, la comunicació pública i, en el seu cas, la transformació.

(n) . Els elements de la llicència són les característiques principals de la llicència segons la selecció efectuada pel llicenciador i indicades en el títol d'aquesta llicència: Reconeixement, CompartirIgual.

(o) . Una llicència equivalent és:

i. . Una versió posterior d'aquesta llicència de Creative Commons amb els mateixos elements de la llicència.

ii. . La mateixa versió o una versió posterior d'aquesta llicència de qualsevol altra jurisdicció reconeguda per Creative Commons amb els mateixos elements de la llicència (per exemple: Reconeixement-CompartirIgual 3.0 Japó).

iii. . La mateixa versió o una versió posterior de la llicència de Creative Commons no adaptada a cap jurisdicció (Unported) amb els mateixos elements de la llicència.

iv. . Una de les llicències compatibles que apareix a http://creativecommons.org/compatiblelicenses i que ha estat aprovada per Creative Commons com a essencialment equivalent d'aquesta llicència perquè, com a mínim:

A. . Conté termes amb el mateix propòsit, el mateix significat i el mateix efecte que els elements d'aquesta llicència.

B. . Permet explícitament que les obres derivades d'obres subjectes a aquella puguin ser distribuïdes mitjançant aquesta llicència, la llicència de Creative Commons no adaptada a cap jurisdicció (Unported) o una llicència de qualsevol altra jurisdicció reconeguda per Creative Commons, amb els mateixos elements de la llicència.

2. . Límits dels drets. No hi ha res en aquesta llicència que pretengui reduir o restringir qualsevol dels límits legals dels drets

exclusius del titular dels drets de propietat intel·lectual d'acord amb la Llei de propietat intel·lectual o qualsevol altra normativa aplicable, ja siguin drets derivats d'usos legítims —com ara la còpia privada o la citació— ja siguin altres limitacions —com la que resulta de la primera venda d'exemplars (exhauriment).

3. . Concessió de llicència. Conforme a les condicions d'aquesta llicència, el llicenciador concedeix, pel termini de protecció dels drets de propietat intel·lectual i a títol gratuït, una llicència d'àmbit mundial no exclusiva que inclou els drets següents:

   (a) . Dret de reproduir, distribuir i comunicar públicament l'obra o la prestació.

   (b) . Dret d'incorporar l'obra o la prestació en una o més col·leccions.

   (c) . Dret de reproduir, distribuir i comunicar públicament l'obra o la prestació lícitament incorporada en una col·lecció.

   (d) . Dret de transformar l'obra per crear-ne una obra derivada sempre que s'hi inclogui una indicació de la transformació o modificació efectuada.

   (e) . Dret de reproduir, distribuir i comunicar públicament les obres derivades creades a partir de l'obra subjecta a la llicència.

   (f) . Dret d'extraure i reutilitzar l'obra o la prestació d'una base de dades.

   (g) . Per aclarir qualsevol dubte, el titular originari:

      i. . Conserva el dret de percebre les remuneracions o compensacions previstes por actes d'explotació de l'obra o la prestació, qualificades per la llei com a irrenunciables i inalienables i subjectes a gestió col·lectiva obligatòria.

      ii. . Renuncia al dret exclusiu de percebre, tant individualment com mitjançant una entitat de gestió col·lectiva de drets, qualsevol remuneració derivada d'actes d'explotació de l'obra o la prestació que feu.

   Aquests drets es poden exercir en tots els mitjans i formats, tangibles o intangibles, coneguts en el moment de concessió d'aquesta llicència. Els drets esmentats inclouen el dret

a efectuar les modificacions que siguin tècnicament necessàries per exercir els drets en altres mitjans i formats. Tots aquells drets no cedits expressament pel llicenciador queden reservats, incloent-hi, a títol enunciatiu però no limitatiu, els drets morals irrenunciables reconeguts per la llei aplicable. En la mesura que el llicenciador tingui drets exclusius previstos per la llei nacional vigent que implementa la directiva europea en matèria de dret sui generis sobre bases de dades, renuncia expressament a aquests drets exclusius.

4. . Restriccions. La concessió de drets que implica aquesta llicència està subjecta i limitada a les restriccions següents:

   (a) . Podeu reproduir, distribuir o comunicar públicament l'obra o la prestació només en els termes d'aquesta llicència i n'heu d'incloure una còpia, o l'identificador uniforme de recursos (URI) corresponent. No podeu oferir o imposar cap condició sobre l'obra o la prestació que alteri o restringeixi els termes d'aquesta llicència o l'exercici dels drets per part dels seus concessionaris. No podeu subllicenciar l'obra o la prestació. Heu de mantenir intactes tots els avisos referents a aquesta llicència i a l'absència de garanties. No podeu reproduir, distribuir o comunicar públicament l'obra o la prestació amb mesures tecnològiques que en controlin l'accés o l'ús d'una manera contrària als termes d'aquesta llicència. Aquesta secció 4.a també afecta l'obra o la prestació incorporada en una col·lecció, però no implica que la col·lecció en conjunt automàticament quedi o hagi de quedar subjecta als mateixos termes. En el cas que us sigui requerit, mitjançant una comunicació prèvia del llicenciador, si incorporeu l'obra o la prestació en una col·lecció o en creeu una obra derivada, haureu d'eliminar qualsevol dels crèdits requerits en l'apartat 4.c, fins on sigui possible.

   (b) . Podeu distribuir o comunicar públicament una obra derivada, en el sentit d'aquesta llicència, només en els seus termes o mitjançant una llicència equivalent. Si useu aquesta mateixa llicència n'heu d'incloure una còpia o bé l'URI, amb cada obra derivada que distribuïu o comuniqueu públicament. No podeu oferir o imposar cap condició respecte de l'obra derivada que alteri o restrin-

geixi els termes d'aquesta llicència o l'exercici dels drets per part dels seus concessionaris. Quan distribuïu o comuniqueu públicament l'obra derivada, heu de mantenir intactes tots els avisos referents a aquesta llicència i a l'absència de garanties. No podeu reproduir, distribuir o comunicar públicament l'obra derivada amb mesures tecnològiques que en controlin l'accés o l'ús d'una manera contrària als termes d'aquesta llicència. Si useu una llicència equivalent haureu de complir-ne els requisits establerts quan distribuïu o comuniqueu públicament l'obra derivada. Totes les condicions d'aquesta secció 4.b també afecten l'obra derivada incorporada en una col·lecció, però no implica que la col·lecció en conjunt automàticament en quedi o n'hagi de quedar subjecta.

(c) . Si reproduïu, distribuïu o comuniqueu públicament l'obra o la prestació, una col·lecció que la incorpori o qualsevol obra derivada, heu de mantenir intactes tots els avisos sobre la propietat intel·lectual i indicar, de manera raonable conforme al mitjà o als mitjans que useu:

  i. . El nom de l'autor original, o el pseudònim si és el cas, així com el del titular originari, si s'ha facilitat.

  ii. . El nom d'aquelles parts (per exemple: institució, publicació, revista) que el titular originari o el llicenciador designin perquè siguin reconegudes en l'avís legal, les condicions d'ús, o de qualsevol altra manera raonable.

  iii. . El títol de l'obra o la prestació si s'ha facilitat.

  iv. . L'URI, si n'hi ha, que el llicenciador especifiqui perquè sigui vinculat a l'obra o la prestació, llevat que aquest URI no es refereixi a l'avís legal o a la informació sobre la llicència de l'obra o la prestació.

  v. . En el cas d'una obra derivada, un avís que identifiqui la transformació de l'obra en l'obra derivada (p. ex., «traducció catalana de l'obra de l'autor original», o «guió basat en l'obra original de l'autor original»).

(d) . Aquest reconeixement s'ha de fer de manera raonable. En el cas d'una obra derivada o d'una incorporació en una col·lecció, aquests crèdits hauran d'aparèixer com a mínim en el mateix lloc on hi hagi els corresponents a altres

autors o titulars i de forma comparable a aquests. Per aclarir qualsevol dubte, els crèdits requerits en aquesta secció només s'usaran a efectes d'atribució de l'obra o la prestació en la manera especificada anteriorment. Sense un permís previ per escrit, no podeu afirmar ni fer entendre implícitament ni explícitament cap connexió, patrocini o aprovació per part del titular originari, el llicenciador o les parts reconegudes cap a vós o cap l'ús que feu de l'obra o la prestació.

(e) . Per aclarir qualsevol dubte, s'ha de fer notar que les restriccions anteriors (paràgrafs 4.a, 4.b i 4.c) no s'aplicaran a les parts de l'obra o la prestació subjecta a aquesta llicència que únicament puguin ser protegides mitjançant el dret sui generis sobre bases de dades recollit per la llei nacional vigent que implementa la directiva europea de bases de dades.

5. . Exoneració de responsabilitat LLEVAT QUE S'ACORDI MÚTUAMENT ENTRE LES PARTS, EL LLICENCIADOR OFEREIX L'OBRA O LA PRESTACIÓ TAL COM ÉS (ON AN AS-IS BASIS) I NO CONFEREIX GARANTIA DE CAP MENA RESPECTE DE L'OBRA O LA PRESTACIÓ O LA PRESÈNCIA O L'ABSÈNCIA D'ERRORS QUE ES PUGUIN DETECTAR O NO. ALGUNES JURISDICCIONS NO PERMETEN L'EXCLUSIÓ D'AQUESTES GARANTIES, DE MANERA QUE AQUESTA EXCLUSIÓ POT NO SER APLICABLE EN EL VOSTRE CAS.

6. . Limitació de responsabilitat. LLEVAT QUE LA LLEI APLICABLE HO DISPOSI EXPRESSAMENT I IMPERATIVAMENT, EN CAP CAS EL LLICENCIADOR SERÀ RESPONSABLE DAVANT VOSTRE PER QUALSEVOL DANY QUE EN PUGUI RESULTAR, TANT SI ÉS GENERAL COM ESPECIAL (INCLOSOS EL DANY EMERGENT I EL LUCRE CESSANT), FORTUÏT O CAUSAL, DIRECTE O INDIRECTE, PRODUÏT PER L'ÚS DE L'OBRA O LA PRESTACIÓ DURANT EL PERÍODE EN QUÈ S'ESTÉN AQUESTA LLICÈNCIA, ENCARA QUE EL LLICENCIADOR HAGUÉS ESTAT INFORMAT DE LA POSSIBILITAT D'AQUESTS DANYS.

7. . Finalització de la llicència

(a) . Aquesta llicència i la concessió dels drets que conté acabaran automàticament en cas que se n'incompleixi qualsevol dels termes. No obstant això, no s'anul·larà la llicència d'aquelles persones o entitats que hagin rebut obres derivades o col·leccions vostres amb aquesta llicència, sempre que aquestes persones o entitats es mantinguin en el compliment íntegre d'aquesta llicència. Les seccions 1, 2, 5, 6, 7, i 8 continuaran vigents malgrat que finalitzi aquesta llicència.

(b) . Conforme a les condicions anteriors, la concessió de drets d'aquesta llicència és vigent per tot el termini de protecció dels drets de propietat intel·lectual segons la llei aplicable. Malgrat això, el llicenciador es reserva el dret a divulgar o publicar l'obra o la prestació en condicions diferents a les presents, o a retirar-la en qualsevol moment. Tanmateix, això no suposarà donar per conclosa aquesta llicència (o qualsevol altra llicència que hagi estat concedida, o calgui concedir, en els termes d'aquesta llicència), sinó que continuarà vigent i amb efectes complets, a menys que no hagi finalitzat conforme allò establert anteriorment, sense perjudici del dret moral de penediment en els termes reconeguts per la llei de propietat intel·lectual aplicable.

8. . Miscel·lània

(a) . Cada vegada que exploteu d'alguna manera l'obra o la prestació, o una col·lecció que la incorpori, el llicenciador ofereix als tercers i successius llicenciataris la concessió de drets sobre l'obra o la prestació, en les mateixes condicions que la llicència que us ha estat concedida.

(b) . Cada vegada que exploteu d'alguna manera una obra derivada, el llicenciador original ofereix als tercers i successius llicenciataris la concessió de drets sobre l'obra subjecta a aquesta llicència, en les mateixes condicions que la llicència que us ha estat concedida.

(c) . Si alguna disposició d'aquesta llicència resulta invàlida o inaplicable segons la llei vigent, això no afectarà la validesa o l'aplicabilitat de la resta dels termes d'aquesta llicència i, sense cap acció addicional per a qualsevol de les parts d'aquest acord, aquesta disposició s'entendrà re-

formada en allò estrictament necessari per fer que aquesta disposició sigui vàlida i executiva.

(d) . No s'entendrà que hi ha renúncia pel que fa a algun terme o disposició d'aquesta llicència, ni que se'n consent cap violació, llevat que la renúncia o el consentiment figuri per escrit i estigui firmat per la part que renuncia o que consent.

(e) . Aquesta llicència constitueix l'acord ple entre les parts respecte de l'obra o la prestació objecte d'aquesta llicència. No hi ha interpretacions, acords o condicions respecte de l'obra o la prestació que no s'especifiquin expressament en aquesta llicència. El llicenciador no estarà obligat per cap disposició complementària que pugui aparèixer en qualsevol comunicació vostra. Aquesta llicència no es pot modificar sense que ho acordeu per escrit amb el llicenciador.

Avís de Creative Commons

Creative Commons no és part d'aquesta llicència, i no ofereix cap garantia en relació amb l'obra o la prestació. Creative Commons no serà responsable davant vostre —o davant cap part—, ni per cap dany de cap tipus, tant si és general o especial (incloent-hi el dany emergent i el lucre cessant), com si és fortuït o causal, que estigui relacionat amb aquesta llicència. A pesar de les dues (2) afirmacions anteriors, si Creative Commons s'identifica expressament com a llicenciador, tindrà tots els drets i les obligacions del llicenciador. Llevat del propòsit limitat d'indicar al públic que l'obra o la prestació està subjecta a una llicència CCPL, cap part usarà la marca registrada Creative Commons ni cap altra marca registrada o insígnia relacionada amb Creative Commons sense el seu consentiment per escrit. Qualsevol ús permès es farà conforme a les pautes vigents en cada moment sobre l'ús de la marca registrada per Creative Commons, publicades en el seu web o proporcionades amb petició prèvia. Per aclarir qualsevol dubte, les restriccions en l'ús de la marca no formen part d'aquesta llicència.

Podeu contactar amb Creative Commons a: http://creativecommons.org/.

Índex

Índex de figures **iii**

Índex de taules **v**

1 Introducció **1**
- **1.1 Estat de l'art** 2
- **1.2 Justificació** 3
- **1.3 Objectius** 6
- **1.4 Metodologia** 7
- **1.5 Planificació** 8
- **1.6 Organització de la memòria** 11

2 Requeriments **13**
- **2.1 O1 Protocol UUP amb comunicacions centralitzades** 13
  - **2.1.1 R01** 13
- **2.2 O2 Servidor UUP** 14
  - **2.2.1 R02** 14
  - **2.2.2 R03** 14
  - **2.2.3 R04** 15
- **2.3 O3 Client UUP** 15
  - **2.3.1 R05** 15
- **2.4 O4 Millora de les comunicacions centralitzades** . . . 15
  - **2.4.1 R06** 15
  - **2.4.2 R07** 15

3 Disseny **17**
- **3.1 Disseny del protocol** 17
  - **3.1.1 UUP amb comunicacions centralitzades** . . . 17
  - **3.1.2 Model d'atacant** 18
  - **3.1.3 Proteccions contra els atacs** 22
  - **3.1.4 Especificació del nou protocol** 25

3.2 Arquitectura del sistema 30
3.2.1 Tecnologies de base 30
3.2.2 Model de dades 30
3.2.3 Casos d'ús 31
3.2.4 Diagrames de seqüència 32

4 Implementació **33**
4.1 Implementació del node central amb servlets 34
4.1.1 Diagrama de classes 34
4.1.2 Diagrames de seqüència 35
4.1.3 Gestió de les sessions d'usuari 36
4.1.4 Paràmetres de configuració 37
4.1.5 Components 37
4.2 Implementació del client UUP en Android 39
4.2.1 Diagrama de classes 39
4.2.2 Diagrames de seqüència 40
4.2.3 Components Android 41
4.2.4 Altres components 43
4.3 Criptografia 44
4.4 Persistència de dades 45
4.4.1 Model relacional de dades 45

5 Test **47**

6 Conclusions **51**

A Nocions prèvies **53**

B Format dels missatges **59**

C Implementació de les operacions criptogràfiques **77**

D Persistència de dades **85**

E Dades de test **87**

Bibliografia **91**

Índex de figures

1.1 **Diagrama de Gantt complet** 10

3.1 **Diagrama de classes del domini** 31
3.2 **Diagrama de casos d'ús** 31
3.3 **Diagrama de casos d'ús** 31
3.4 **Diagrama de seqüència del servlet** 32
3.5 **Diagrama de seqüència** 32

4.1 **Diagrama de classes** 34
4.2 **Diagrama de seqüència simplificat del funcionament del node central en el moment de crear un grup d'usuaris** 35
4.3 **Diagrama de seqüència simplificat del funcionament del node central en el moment que un node envia un missatge** 36
4.4 **Diagrama de classes** 39
4.5 **Diagrama de seqüència simplificat del funcionament del client UUP en Android** 40
4.6 **Diagrama de seqüència simplificat del funcionament del client UUP en Android durant un dels passos del protocol** 40

5.1 **Temps dels passos del protocol per execucions amb $n = 2$, 3 i 4 usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Interneti la resta a la mateixa xarxa local que el node central** 48

E.1 **Temps dels passos del protocol per execucions amb $n = 2$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari a la mateixa xarxa local que el node central i l'altre usuari des de Internet** 87

E.2 Temps dels passos del protocol per execucions amb $n = 2$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari a la mateixa xarxa local que el node central i l'altre usuari des de Internet 87
E.3 Temps dels passos del protocol per execucions amb $n = 3$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Internet i la resta a la mateixa xarxa local que el node central 87
E.4 Temps dels passos del protocol per execucions amb $n = 3$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Internet i la resta a la mateixa xarxa local que el node central 88
E.5 Temps dels passos del protocol per execucions amb $n = 4$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Internet i la resta a la mateixa xarxa local que el node central 88
E.6 Temps dels passos del protocol per execucions amb $n = 4$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Internet i la resta a la mateixa xarxa local que el node central 88
E.7 Comparativa dels temps parcials del protocol amb diferent nombre d'usuaris per a una longitud de clau $l = 1024$ 88
E.8 Comparativa dels temps parcials del protocol amb diferent nombre d'usuaris per a una longitud de clau $l = 1024$ 88
E.9 Comparativa dels temps parcials del protocol amb diferent nombre d'usuaris per a una longitud de clau $l = 2048$ 89
E.10 Comparativa dels temps parcials del protocol amb diferent nombre d'usuaris per a una longitud de clau $l = 2048$ 89
E.11 Comparativa dels temps parcials del protocol amb $n = 2$, 3 i 4 usuaris per a una longitud de clau $l = 2048$ i 1024 89

Índex de taules

1.1 Planificació de les fases del projecte 9

Capítol 1

Introducció

Els motors de cerca (web search engines - WSEs) com Google, Yahoo o Bing són eines bàsiques per trobar informació de manera ràpida entre el gran volum de dades que hi ha a Internet. Els motors de cerca responen a les consultes de cerca dels usuaris però també les analitzen i les guarden.

Els WSEs utilitzen aquesta informació oficialment per millorar els algorismes de ranking que calculen la rellevància de les pàgines web, corregir errors en les consultes que envien els usuaris, donar suggeriments de consultes als usuaris, i per crear perfils d'usuari (User Profiles). Però tota aquesta informació també serveix per incrementar els beneficis econòmics dels WSEs amb el màrqueting personalitzat a pàgines web [14], obtenint més usuaris pel bon resultat i per tant incrementant beneficis per anunciant, o inclús venent perfils d'usuari a certes agències governamentals relacionades amb la justícia [14, 22, 15].

Per a la construcció del perfil d'un usuari, aquest és identificat utilitzant pseudo-identificadors com l'adreça IP, les Galetes o Cookies, les barres de cerca en els navegadors (com la Google Toolbar) o amb les dades de versió i configuració del navegador (browser fingerprinting). Aquests pseudo-identificadors es poden relacionar amb la identitat real si el proveïdor de Internet (ISP) revela la identitat del propietari de l'adreça IP, si l'usuari ha accedit a la seva compte de Google i envia consultes, o també si l'usuari envia consultes sobre informació que l'identifiqui. Però cal remarcar, que tot i que consultes aïllades no revelin la identitat de l'usuari, l'agregació de múltiples consultes aparentment inofensives poden

revelar aquesta identitat real[5] (i no caldria doncs utilitzar dades de l'ISP o la compte de Google a la que l'usuari ha accedit).

La problemàtica d'aquest funcionament és que els perfils d'usuari poden contenir informació especialment sensible com dades de salud, tendències sexuals, estatus econòmic o creences religioses. Això representa una amenaça per la privacitat. A més a més aquestes dades s'emmagatzemen en bases de dades que no estan sota el nostre control. Aquestes bases de dades podrien ser atacades, les dades es podrien publicar per error o ser venudes directament per els WSEs. Un exemple d'això és l'escàndol de AOL [5] on 20 milions de consultes generades per milers d'usuaris es van publicar a Internet per facilitar diversos estudis científics.

1.1 Estat de l'art

Per fer front a aquesta amenaça han aparegut nous mètodes per utilitzar els motors de cerca web tot preservant la privacitat dels usuaris. La problemàtica d'aquests mètodes es centre en aconseguir un nivell de privacitat acceptable i un temps de resposta factible a les cerques. Aquestes alternatives es poden classificar com a protocols single-party i protocols multi-party.

Els protocols single-party treballen directament a l'ordinador de l'usuari i no utilitzen entitats externes. Aquestes alternatives obtenen el millor temps de resposta ja que no utilitzen cap entitat externa per obtenir els resultats de cerca. Un exemple d'aquestes solucions és l'us de IP dinàmica amb navegador estàndard i sense cookies. En aquest cas l'inconvenient és que no tots els usuaris tenen accés a una IP dinàmica, ni en cas de tenir-la no tots la poden canviar fàcilment. Altres exemples són les solucions GooPir [12] i TrackMeNot [16] que envien consultes generades automàticament als motors de cerca, de manera que el perfil d'usuari creat pel motor de cerca no sigui fidel a la realitat. El nivell de privacitat obtingut amb aquests mètodes no és acceptable com es demostra a [9] i [17], ja que és possible discernir les consultes generades automàticament de les consultes generades per l'usuari.

Els protocols multi-party requereixen la col·laboració d'entitats externes per generar o enviar consultes al WSE. Las entitats ex-

ternes s'utilitzen per formar una xarxa de anonimització mitjançant proxies o grups d'usuaris. Un exemple de xarxa anònima amb proxies és la xarxa Tor [11] que utilitza eines criptogràfiques. L'inconvenient d'aquesta alternativa és el temps de resposta elevat que es requereix per enviar una consulta, 25 vegades més lent que una consulta directa [20].

En quant a alternatives de protocols multi-party utilitzant grups d'usuaris es tracta de crear un grup d'usuaris cooperadors on cadascú realitza cerques al WSE sota la demanda d'altres usuaris. D'aquesta manera s'evita la creació de un perfil d'usuari que mostri dades de la seva privacitat, ja que cada usuari envia consultes d'altres usuaris però mai les seves. El cost de les comunicacions és menor que en el cas d'una xarxa de proxies ja que la xarxa és més simple i s'utilitza menys criptografia.

Aquestes últimes alternatives es poden classificar en dos grups, les solucions que utilitzen grups d'usuaris estàtics i les que utilitzen grups d'usuaris dinàmics. Pel que fa a les solucions amb grups estàtics d'usuaris, on els usuaris son sempre els mateixos, les propostes es troben explicades en els treballs [18] i [23]. El problema principal d'aquestes alternatives es que els grups estàtics d'usuaris son més vulnerables a atacs interns contra el protocol, on un usuari deshonest dintre del grup pot col·laborar amb el WSE per crear perfils d'usuaris desmuntant la xarxa anònima. Pel que fa a solucions amb grups d'usuaris dinàmics, les solucions presentades en [8] i [] que expliquen el protocol anomenat Useless User Profile (UUP), formen grups d'usuaris aleatoris interessats en realitzar consultes al WSE. Aquestes solucions son més segures contra atacs interns. La versió del Useless User Profile (UUP) presentada en [19] aconsegueix un temps de resposta de 3.2 segons.

1.2 Justificació

Aquest treball de final de màster es basa en la versió millorada del protocol Useless User Profile presentada en [19], on es proposa de crear grups dinàmics d'usuaris on es puguin intercanviar de manera segura les consultes de cerca que es volen realitzar.

El pes dels dispositius Smartphone és important en el mercat

avui en dia, i com s'ha vist actualment l'evolució d'aquests darrers anys no s'atura encara sinó que segueix creixent [7]. També es promociona constantment la integració del smartphone en l'estil de vida per estar "connectat" des de qualsevol lloc i en qualsevol moment. S'incentiva l'us de l'smartphone per comunicacions de xat, publicacions en xarxes socials, consultes de geolocalització, sincronització de continguts en el cloud i també cerques en motors web.

Per tant la plataforma smartphone es un sector important a tenir en compte a l'hora de difondre una eina per preservar la privacitat i alhora conscienciar a la comunitat de internautes de la importància de protegir la seva privacitat. Concretament s'ha escollit com a primer pas implementar la versió del protocol UUP per a la plataforma Android ja que és un sistema operatiu lliure i permet el desenvolupament d'aplicacions de manera senzilla en una versió del llenguatge Java, molt conegut per nosaltres, adaptada per a Android.

La tasca de crear una versió del protocol Useless User Profile per a Smartphones aporta nous reptes i problemàtiques al problema inicial. Hi ha una característica inherent als dispositius mòbils en general que modifica l'escenari de partida. Aquests dispositius estan pensats per a ser estables i segurs (sobretot de cara a les funcions de telefonia) i com a conseqüència d'això les aplicacions d'usuari que s'hi executen no poden accedir o modificar certes funcions del sistema operatiu. No és fàcil fer funcionar una aplicació en un dispositiu mòbil que treballi com a servidor, esperant rebre dades o peticions de xarxa. No obstant això, s'han descobert tècniques per a dispositius Android i Iphone que permeten obtenir permisos d'administrador o executar qualsevol aplicació [21]. Aquestes modificacions en el dispositiu poden malmetre el seu funcionament i n'anul·len la garantia de compra. Per tant d'entrada no es pot suposar que un usuari normal tingui els coneixements necessaris per a obtenir permisos d'administrador ni que accedeixi a perdre la garantia de compra del seu producte.

Aquesta implementació del protocol UUP no és directa ni planera i bona part d'aquest treball consistirà en implementar una arquitectura client servidor per a que els dispositius mòbils es comuniquin entre sí, centralitzant la comunicació en un servidor. Aquesta arquitectura també aporta el repte de assegurar el fun-

cionament del protocol UUP suposant que el servidor per on passa la comunicació entre dispositius (el node central) sigui deshonest o no fiable. Aquest node central podria enviar informació al motor de cerca per tal de espatllar el funcionament del protocol UUP i amenaçar la privacitat dels usuaris.

1.3 Objectius

Per a dur a terme la implementació de la versió del protocol Useless User Profile de [19] adaptada per a smartphones Android els objectius son:

1. **Dissenyar una nova versió del protocol UUP explicat en [19] per a funcionar en un escenari on la comunicació està centralitzada en el node central.**

2. **Dissenyar e implementar una aplicació servidor per realitzar les tasques de node central del protocol UUP en la versió de l'objectiu 1. El node central s'encarrega de realitzar la funció de creació de grups dinàmics d'usuaris i de gestionar la comunicació centralitzada.**

3. **Implementar una aplicació client per a Android del protocol UUP en la versió de l'objectiu 1.**

4. **Aplicar millores al protocol UUP de l'objectiu 1 per tal de preservar la privacitat en front de atacants que explotin vulnerabilitats del protocol per cooperar amb el motor de cerca per crear perfils d'usuari.**

1.4 Metodologia

La realització del treball tindrà en compte les següents fases i les seves tasques:

1. **Requeriments**

   R01 **comunicacions centralitzades en el node central (client i servidor)**

   R02 **servidor UUP en el node central**

   R03 **client UUP**

   R04 **comunicacions centralitzades UUP segures**

2. **Anàlisi i Disseny**

   AD01 **comunicacions centralitzades en el node central (client i servidor)**

   AD02 **servidor UUP en el node central**

   AD03 **client UUP**

   AD04 **comunicacions centralitzades UUP segures**

3. **Implementació**

   I01 **comunicacions centralitzades en el node central (client i servidor)**

   - **ID01.1** **comunicacions de xarxa en Android**
   - **ID01.2** **servidor**
   - **ID01.3** **client**

   I02 **servidor UUP en el node central**

   I03 **client UUP**

   - **ID03.1** **criptografia en Android**
   - **ID03.2** **arquitectura d'aplicacions Android e interfície gràfica**
   - **ID03.3** **protocol UUP**

   I04 **comunicacions centralitzades UUP segures**

4. **Tests**

5. **Memòria**

   M01 **Introducció**

   M02 **Requeriments**

   M03 **Anàlisi i Disseny**

   M04 **Implementació**

   M05 **Tests**

   M06 **Conclusions**

6. **Presentació**

1.5 Planificació

En aquest apartat es mostra la temporització de les tasques associades a cada fase del projecte. En la taula 1.1 es mostra la temporització de cada tasca i en la figura 1.1 es mostra el diagrama de Gantt del projecte.

Nom	Data inici	Data finalització	Durada
Requeriments	22/10/12	29/10/12	6
R01 - comunicacions centralitzades	22/10/12	23/10/12	2
R02 - servidor UUP	24/10/12	24/10/12	1
R03 - client UUP	25/10/12	25/10/12	1
R04 - comunicacions centralitzades UUP segures	26/10/12	29/10/12	2
Anàlisi i Disseny	30/10/12	16/11/12	13
AD01 - comunicacions centralitzades	30/10/12	02/11/12	3
AD02 - servidor UUP	05/11/12	07/11/12	3
AD03 - client UUP	08/11/12	12/11/12	3
AD04 - comunicacions centralitzades UUP segures	13/11/12	16/11/12	4
Implementació	19/11/12	13/12/12	18
I01 - comunicacions centralitzades	19/11/12	26/11/12	6
I01.1 - comunicacions de xarxa en Android	19/11/12	19/11/12	1
I01.2 - servidor	20/11/12	22/11/12	3
I01.3 - client	23/11/12	26/11/12	2
I02 - servidor UUP	27/11/12	27/11/12	1
I03 - client UUP	28/11/12	05/12/12	6
I03.1 - criptografia en Android	28/11/12	29/11/12	2
I03.2 - arquitectura d'aplicacions Android e interfície gràfica	30/11/12	30/11/12	1
I03.3 - UUP	03/12/12	05/12/12	3
I04 - comunicacions centralitzades UUP segures	07/12/12	13/12/12	5
Tests	14/11/12	24/12/12	7
Memòria	16/10/12	28/12/12	51
M01 - Introducció	16/10/12	19/10/12	4
M02 - Requeriments	22/10/12	30/10/12	7
M03 - Anàlisi i Disseny	31/10/12	19/11/12	13
M04 - Implementació	20/11/12	14/12/12	18
M05 - Tests	17/12/12	24/12/12	6
M06 - Conclusions	26/12/12	28/12/12	3
Presentació	31/12/12	08/01/13	6

Taula 1.1: Planificació de les fases del projecte

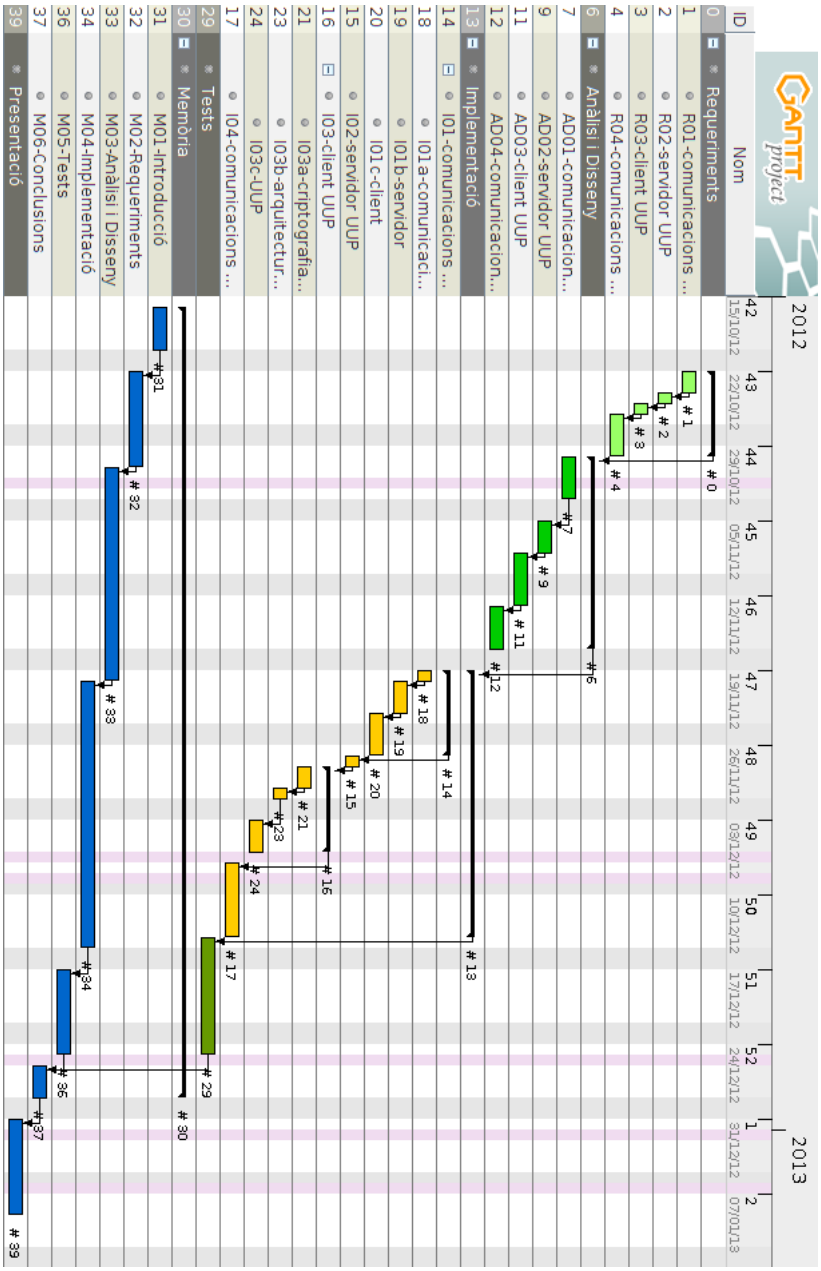

Figura 1.1: Diagrama de Gantt complet

1.6 Organització de la memòria

La memòria d'aquest treball de final de màster està dividida en capítols que expliquen la seva evolució des de la motivació que ha impulsat a la seva realització fins a la implementació final. Aquesta evolució plasmada en els diferents capítols aporta tot el necessari per entendre com s'ha arribat a la solució final.

En aquest capítol introductori s'ha definit l'àmbit del treball així com l'estat de l'art i els antecedents per entendre en quin context neix aquest projecte. També es presenten les tasques bàsiques en la fase inicial del projecte: els objectius del treball així com la metodologia i la planificació per a dur-lo a terme.

El capítol Nocions Bàsiques explica les nocions sobre protocols criptogràfics que s'utilitzen en aquest projecte. També s'explica el funcionament del protocol Useless User Profile del paper [19] en la qual es basa tot el treball.

Tot seguit es troba el capítol dedicat als requeriments. Aquest capítol serveix per tancar les fases inicials del projecte, i és on es fa un recull de necessitats, objectius detallats i propietats que es volen complir. És en aquest punt on es defineixen les bases per a l'especificació de la solució.

En el capítol anomenat Disseny es detalla l'especificació de la solució. Aquesta especificació es centra en les propietats que finalment complirà el sistema i les mesures de seguretat que s'implementaran, així com en l'arquitectura del sistema i el protocol de comunicacions dissenyat. Igualment es detallen totes les decisions de disseny basades en l'estudi d'alternatives per adequar el protocol UUP[19] a l'entorn del projecte, mitigar problemes de seguretat i mantenir el compromís amb una planificació temporal estricte per realitzar el treball.

El capítol Implementació explica el desenvolupament del programari que implementa la solució dissenyada. El capítol dedicat als Tests mostra la validació de les funcionalitats del sistema per mitjà de tests unitaris.

Finalment l'apartat Conclusions analitza els punts dèbils del

projecte, comenta les desviacions respecte de la planificació i detalla les ampliacions possibles.

Capítol 2

Requeriments

En aquest apartat s'especifiquen els requeriments d'aquest treball. Es tracta de definir les propietats que es volen complir de cara a la funcionalitat i seguretat del sistema. Per a cada objectiu definit en l'apartat "Introducció" es defineixen els requeriments específics. Per a l'objectiu O1 de creació d'una nova versió del protocol UUP adaptada per a comunicacions centralitzades es detallen les funcionalitats que es volen cobrir. Per als objectius O2 de creació del servidor del protocol UUP i O3 de creació de l'aplicació client UUP s'expliquen les funcionalitats heretades de l'especificació del protocol UUP [19] i l'adaptació d'aquest protocol al sistema de comunicacions de l'objectiu O1. Per a l'objectiu O4 es fa un recull de les vulnerabilitats del sistema heretades del protocol UUP [19] i les que apareixen en el nou escenari, així com un recull de les mitigacions corresponents.

2.1 O1 Protocol UUP amb comunicacions centralitzades

2.1.1 R01

Aquesta versió del protocol UUP ha de preveure que les comunicacions entre nodes d'un mateix grup es faran de manera centralitzada. Això vol dir que tots els missatges entre els nodes passaran per el node central que els gestionarà de manera que arribin al seu destí. A tal efecte el format dels missatges que s'intercanvien els nodes ha de preveure camps per a especificar el node origen, el

node destí, així com altres informacions que siguin necessàries en el nou escenari.
A més a més s'ha de modificar el format del missatge de creació de grup de usuaris per a que aquest inclogui la informació necessària per utilitzar aquest sistema de comunicacions centralitzades.

2.2 O2 Servidor UUP

2.2.1 R02

El node central, o servidor, ha de permetre la comunicació entre diferents nodes, que seran terminals android. Els terminals android, o nodes, no poden rebre connexions directes de xarxa. Per tant s'utilitzarà un sistema de polling, els nodes actuen només com a clients que fan peticions al servidor node central i aquest les respon. Cada node podrà rebre missatges de un o més nodes. El node ha de consultar al node central si té missatges per ell. Cada node podrà enviar missatges a un o més nodes. Els missatges s'envien al node central i aquest els farà arribar al seu destinatari quan aquest últim ho demani.

2.2.2 R03

El node central ha de seguir els passos estipulats a la versió millorada del protocol UUP de [19]. Com a node central les seves tasques son:

1. Estar a l'espera de peticions de xarxa de usuaris per entrar en un grup UUP.

2. Quan el node central ha rebut suficients peticions de usuaris (el nombre de peticions n necessàries per formar un grup s'extreu de [19]) es posa a formar un nou grup amb els usuaris $\{U_1, ..., U_n\}$. Realitza l'enviament a tots els usuaris de la informació per contactar amb els altres usuaris del grup i l'enviament de paràmetres necessaris per a la generació de la clau de grup. Els paràmetres per a la generació de la clau de grup són un nombre primer prou gran p on $p = 2q + 1$ i q es igualment primer, i un element $g \in Z_q^*$.

2.2.3 R04

Com a node central del protocol UUP, aquest servidor rep peticions del usuaris directament des de Internet. En el moment de formar el grup de usuaris, el node central també facilitarà la informació necessària als nodes per a que utilitzin el sistema de comunicacions centralitzades en el protocol UUP.

2.3 O3 Client UUP

2.3.1 R05

L'aplicació client UUP permetrà a un usuari de smartphone utilitzar aquest protocol. Aquesta aplicació client ha de seguir els passos del protocol UUP millorat estipulats en [19]. Aquests passos es mostren a l'apèndix A secció 2 "El protocol Useless User Profile". S'utilitza el terme aplicació client UUP, node o usuari per referir-se indistintament a un usuari executant l'aplicació client del protocol UUP en un terminal smartphone.

2.4 O4 Millora de les comunicacions centralitzades

2.4.1 R06

Cal estudiar i definir en aquest nou escenari totes les vulnerabilitats del protocol UUP de cara a poder aplicar millores. Cal revisar els atacs ja existents en la versió del protocol UUP de [19] per veure si segueixen sent vàlids i verificar si n'hi ha de nous. L'objectiu és definir un model d'atacant.

2.4.2 R07

Dissenyar i/o implementar millores de seguretat al protocol UUP per la versió centralitzada tenint en compte els recursos de còmput, el temps de resposta i la viabilitat del projecte.

Capítol 3

Disseny

3.1 Disseny del protocol

En aquest apartat es detallen tots els aspectes del protocol que s'han de treballar per adaptar-lo al nou escenari. En cada subsecció s'analitzen diferents solucions per tal d'arribar a les decisions de disseny finals.

3.1.1 UUP amb comunicacions centralitzades

Primerament es repassa el funcionament del protocol UUP de [19] en el nou escenari. Per cada pas del protocol es detallen les modificacions necessàries per a executar-lo en un entorn amb terminals mòbils. Aquests terminals no poden rebre connexions de xarxa directes i per tant la comunicació de xarxa s'ha de realitzar de manera centralitzada. Tots els missatges han de passar per el node central.

1. Formació del grup d'usuaris
   Quan el node central té prous usuaris forma un grup. Tots els nodes que formen aquest nou grup reben la informació necessària per formar part de un grup d'usuaris $\{U_1, ..., U_n\}$ del protocol UUP, així com els paràmetres per generar la clau de grup (nombre primer p i element $g \in Z_q^*$).
   En aquesta versió del protocol UUP on totes les comunicacions passen per el node central, els nodes usuari no necessiten gaire informació per a comunicar-se amb la resta de nodes del mateix grup. Com que el node central controla quins nodes pertanyen al grup d'usuaris, qualsevol missatge que envii un

node es pot re-enviar a la resta de nodes. Amb un identificador per cada node del grup n'hi hauria prou per poder enviar missatges entre nodes concrets.
El node central en el moment de formar el grup enviarà els identificadors de tots els nodes del grup (pot ser un identificador intern o la numeració dels nodes que s'utilitzarà en els següents passos).
A partir d'aquest punt el node central només actua com a enrutador de missatges o com a bústia de missatges. Els missatges que envia un usuari es faran arribar a tot el grup o a un usuari concret del grup.

2. Generació de la clau de grup
   L'intercanvi de claus públiques parcials es realitza entre tots els nodes i per tant d'entrada no caldria tenir en compte identificadors dels nodes. Seria convenient que en cada missatge es faci referència a quin pas del protocol s'està executant.

3. Anonimització de les consultes de cerca
   L'intercanvi de la llista de consultes es farà per ordre i aquí sí que seria necessari indicar l'identificador o el número del node destí en cada comunicació.
   Igualment serà útil utilitzar l'indicador del pas del protocol, ja que així el node que rep el missatge sabrà què ha de fer a continuació.

4. Enviament de les consultes al WSE i recepció del resultat
   Aquest punt del protocol no es veu afectat per la comunicació centralitzada ja que cada node té la llista de consultes desencriptades i només les ha de enviar al WSE i obtenir la resposta.

3.1.2 Model d'atacant

En aquesta secció s'analitzen les vulnerabilitats del protocol enfront a possibles atacants. Hi ha tres models d'atacant rellevants: un usuari del protocol deshonest (com a atacant intern), un node central deshonest o un WSE deshonest. Els atacants externs no es consideren una amenaça real tenint en compte l'ús de criptografia de clau pública del protocol. Aquest anàlisi es basa en l'apartat V anomenat "Privacy Analysis" del paper [19], el qual es repassa

tenint en compte el nou escenari.

A. Node Deshonest

En el paper [19] s'analitzen tres atacs que pot realitzar un usuari o node deshonest per descobrir les consultes d'altres usuaris del mateix grup. Aquests atacs provenen del paper [24].

1. Stage-skipping attack: En aquest atac l'usuari U_n re-emmascara i permuta la llista original de consultes xifrades enlloc de la llista que rep de l'usuari U_{n-1}. El següent pas és enviar a tots els usuaris la llista final, i que després els usuaris enviïn els corresponents "shares" (que son part del missatge transformat amb la clau privada tal que si $m_i = (c_1, c_2)$ és el missatge encriptat, cada usuari envia $c_1^{\alpha_j}$) per realitzar el desxifrat. Al realitzar el desxifrat de la llista obtinguda, U_n sabrà per cada consulta de la llista a quin usuari pertany. Aquest atac no és realitzable en el protocol UUP de [19] ni tampoc en l'actual protocol UUP d'aquest treball. Tant el protocol UUP de [19] com el d'aquest treball utilitzen l'operació de desxifrat parcial. La llista original de consultes xifrades s'encripta amb la clau de grup mentre que la llista que es rep de U_{n-1} només està encriptada amb la clau privada de U_n. Per tant ja no és útil reemplaçar la llista que envia l'usuari U_{n-1} amb la llista que envia l'usuari U_1 perquè aquesta última no es podrà desencriptar. Així doncs U_n no pot obtenir les consultes en clar de la llista original de consultes encriptades, només de la llista $n-1$ transformada amb $n-1$ operacions de remasking i reordenació. I per tant ja no pot associar les consultes als usuaris que les han emès originalment.

2. Input-replacement attack: En aquest atac el primer usuari U_1 pot arribar a conèixer la consulta que envia un altre usuari del grup. Inicialment U_1 rep la llista original de consultes xifrades amb la clau pública i n'elimina totes les consultes excepte la de la víctima. Llavors l'usuari U_1 crea una nova llista amb copies de la consulta de la víctima i realitza la operació de "remasking" de cada una de les còpies (aquest "remasking" s'ha de fer amb números aleatoris diferents per cada valor de la llista). També pot utilitzar paraules clau escollides específicament. Quan l'últim usuari re-envia a tot el grup la última

llista de consultes, l'usuari U_1 pot identificar quina consulta pertany a la víctima ja que o bé totes les consultes són les de la víctima, o bé la llista es composa de la consulta de la víctima i de les paraules clau escollides anteriorment.
Per evitar aquest atac es proposa a [19] la introducció de mecanismes de "zero-knowledge proof" o proves de coneixement nul de [6]. Cada usuari haurà de provar que la llista de consultes que envia correspon a la llista que ha rebut aplicant la reordenació i el "remasking".

3. Targeted public-key attack: En la generació de claus del protocol UUP (veure secció II-A1 de [19]), si les claus públiques parcials de tots els nodes no es publiquen alhora, un dels nodes podria realitzar aquest atac. L'atac consisteix en utilitzar els valors de les claus públiques parcials de la resta de nodes per construir la seva pròpia clau pública parcial de la manera següent: $y_j = g^{\alpha_j}/\Pi_{i=1}^{n-1} y_i = g^{\alpha_j - \alpha_1 - \ldots - \alpha_{n-1}}$. D'aquesta manera la clau pública del grup seria : $y = \Pi_{1 \leq i \leq n} y_i = g^{\alpha_1} \cdot g^{\alpha_2} \cdot \ldots \cdot g^{\alpha_{n-1}} \cdot g^{\alpha_j - \alpha_1 - \ldots - \alpha_{n-1}} = g^{\alpha_j}$. D'aquesta manera U_j coneixeria la clau privada del grup també.
Una solució per aquest atac és dividir la fase de compartir les claus públiques parcials en dugues fases. En una primera fase es calculen les claus privades i públiques a cada node i es comparteixen els resums criptogràfics hash de cada clau pública. En una segona fase es comparteixen les claus públiques. D'aquesta manera cada node pot verificar que les claus públiques que ha rebut dels altres nodes corresponen al hash corresponent, i que per tant han estat calculades en la primera fase independentment de les claus públiques dels altres nodes.

B. Node Central Deshonest

En el nou escenari el node central intervé activament durant tots els passos del protocol i això li atorga més poder que en l'escenari de [19]. A continuació es mostra un repàs de les responsabilitats, l'accés a informació e influència del node central en el nou protocol UUP:

1. Té accés a tots els missatges que s'intercanvien els nodes en tots els passos del protocol (els inicials i els que contenen les consultes encriptades)

2. Coneix els paràmetres per a la generació de la clau de grup que són un nombre primer prou gran p on $p = 2q + 1$ i q es igualment primer, i un element $g \in Z_q^*$

3. Té accés a les claus públiques de tots els nodes i per tant pot calcular la clau pública del grup

4. És l'encarregat de decidir com es formen els grups d'usuaris

Tot seguit veurem les implicacions d'això en quant a atacs que pot realitzar el node central.
Tenint en compte que el node central té accés a tots els missatges i a la clau pública del grup, aquest no pot accedir a les consultes o realitzar atacs com a atacant extern, és a dir sense modificar la comunicació entre els nodes. El node central pot calcular la clau pública del grup però com que les consultes estan xifrades amb l'algorisme de clau pública "ElGamal" no tindrà prou informació per a desxifrar la llista de consultes.
D'altra banda, el node central podria suplantar la identitat d'algun dels nodes i actuar com a node deshonest. D'aquesta manera el node central podria realitzar els atacs "Input-replacement attack" i "Targeted public key attack" que s'han explicat anteriorment.
Finalment el node central pot utilitzar la seva autoritat per formar els grups amb la finalitat de construir grups fantasma amb nodes sota el seu control. D'aquesta manera el node central pot assignar un usuari legítim a un grup de n nodes amb $n - 1$ nodes controlats per ell per tal de conèixer la consulta que vol enviar aquest usuari. Una manera de protegir els usuaris contra la formació de grups fantasma seria identificar els nodes amb algun tipus de credencial, i que cada usuari comprovi que no coincideix massa vegades amb altres usuaris. Cada node hauria de generar les seves credencials, sense informació vinculada amb la identitat de l'usuari o número de identificació del terminal. Aquestes credencials s'intercanviarien en la fase de formació del grup i així cada usuari podria guardar una llista dels usuaris amb qui ha compartit grup. Si un usuari U_i detecta que ha compartit grup amb un usuari U_j moltes vegades pot abandonar el grup.

C. WSE Deshonest

L'objectiu del WSE és guardar les consultes que envia cada usuari per formar el seu perfil. Amb el protocol UUP de [19] el WSE no

pot conèixer si una consulta enviada per un usuari ha sigut generada per aquest. Quan un usuari U utilitza el protocol UUP de [19], està amagant la seva consulta entre $n-1$ consultes d'altres usuaris. Recordem que en aquesta versió cada usuari/node envia tota la llista de consultes del grup.
Aquestes $n-1$ consultes d'altres usuaris no s'han generat automàticament i són consultes legítimes. Per tant el WSE no pot utilitzar mètodes com els explicats en [9] o [17] per separar les consultes legítimes de les generades artificialment.
El WSE pot seleccionar les consultes que pertanyen a l'usuari U de manera correcte amb una probabilitat de $1/n$. Això no és suficient per a crear un perfil d'usuari fidel a la realitat ja que la probabilitat de seleccionar sempre consultes que pertanyen realment a l'usuari U és molt baixa. El perfil d'usuari obtingut queda distorsionat.
De totes maneres, aquesta probabilitat pot augmentar si l'usuari U envia vàries consultes sobre el mateix tema. En cada execució del protocol, l'usuari amaga la seva consulta entre n consultes. Aquestes consultes són generades per usuaris escollits aleatòriament i per tant, normalment estaran relacionades amb temes diferents. En conseqüència, després de vàries execucions del protocol, el WSE podria ser capaç de trobar el tema comú en les consultes enviades per l'usuari U i construir el seu perfil.
Aquesta situació es pot evitar enviant periòdicament consultes sobre temes irrellevants per a U. Per tal d'evitar que el WSE utilitzi tècniques com les descrites a [9] i [17], l'usuari U hauria de emmagatzemar les consultes que obté d'altres usuaris en cada execució del protocol i utilitzar-les periòdicament per distorsionar el seu perfil. D'aquesta manera hi haurà temes irrellevants que també apareixeran vàries vegades en les consultes de U, i a més a més seran consultes generades per humans i per tant el WSE no les descartarà.
El WSE podria descartar consultes repetides i així detectar aquesta última estratègia. No seria infal·lible ja que l'usuari pot coincidir amb usuaris que enviïn consultes sobre el mateix tema.

3.1.3 Proteccions contra els atacs

En aquest apartat es repassen les proteccions als atacs esmentats i la seva viabilitat.

Input-replacement attack

La protecció contra aquest atac consisteix a introduir "Zero-knowledge proofs" en el protocol. Cada usuari hauria de provar que la llista de consultes que envia correspon a la llista que ha rebut aplicant les operacions de reordenació i "Remasking". El cost en còmput de la criptografia a utilitzar i el cost en temps fan que aquesta solució sigui poc viable per aquest treball.
És important remarcar que per a que l'atac tingui realment èxit, l'atacant hauria de coincidir amb la víctima vàries vegades i sempre en primera posició en el grup. Com que els grups es formen de manera aleatòria i dinàmica, la probabilitat de que l'atacant coincideixi amb la mateixa víctima dos vegades és molt baixa i es redueix molt més la probabilitat de coincidir més de dos vegades amb la víctima.
Per una banda aquesta modificació causaria un augment del temps de resposta del sistema de com a mínim el doble. Per altra banda la probabilitat d'èxit de l'atac és baixa. Finalment la implementació d'aquesta solució implicaria modificar la planificació del projecte i això no és possible. Aquesta protecció no s'implementarà en aquesta versió del protocol UUP.

Targeted public-key attack

La protecció contra el "Targeted public-key attack" consisteix a compartir primer un resum criptogràfic de la clau pública de cada node $h_i = H(y_i)$, de manera que la clau pública de cada node no sigui calculable a partir del resum criptogràfic. Tots els nodes del grup envien la seva h_i i es guarden els resums h_j de la resta de nodes. Posteriorment els nodes envien la seva clau pública. Cada node ha de verificar que les claus públiques que rep dels altres nodes corresponen amb el resum criptogràfic enviat prèviament. La funció de resum ha de donar el mateix resum criptogràfic amb la clau pública compartida en aquesta segona fase, $h_j = H(y_j)$ per $j = (1, ..., n)$.
Aquesta protecció és factible ja que només introdueix un pas més en el protocol i el cost en còmput del resum criptogràfic no és molt elevat.

Formació de grups fantasma

Com a protecció contra la formació de grups fantasma cada node s'hauria d'identificar amb credencials sense informació vinculada amb la identitat de l'usuari o número de identificació del terminal.

El més adient seria que cada node obtingués un certificat digital (sense vincular-lo amb la seva identitat). Aquest certificat s'intercanvia en la fase de formació del grup i així cada usuari pot guardar una llista de certificats dels usuaris amb qui ha compartit grup. Si un usuari detecta que en el seu grup hi han varis usuaris amb els que ha compartit grup moltes vegades pot abandonar el grup per seguretat. El certificat s'utilitzaria per firmar els missatges on es comparteix la clau publica parcial (i el hash de la clau pública parcial).

Aquesta protecció implica que el problema es traslladi a l'emissió de credencials. Si l'emissió de credencials recau sobre una entitat deshonesta com pot ser el node central o els propis nodes que generin els seus certificats, llavors aquesta protecció no tindria efecte (ja que es podrien generar certificats nous cada vegada per amagar la formació de grups fantasma). Per tant l'entitat que emet els certificats ha de ser independent del WSE. Una solució seria que la responsabilitat de l'emissió de certificats recaigui sobre un consell amb membres que tenen interessos oposats. D'aquesta manera per a emetre un certificat cal que un mínim número de membres del consell hi estiguin d'acord.

Aquesta protecció implica la obtenció del certificat digital que es realitzaria prèviament a la utilització del protocol. També implicaria afegir un parell de passos més en la formació dels grups d'usuaris: guardar els certificats dels usuaris amb els quals es coincideix en un grup i comprovar que el grup és prou segur(que no s'ha repetit moltes vegades amb cap usuari). Tot això no aportarà gaire retard al protocol i per tant es una protecció viable.

WSE common subjects profile

Per evitar que el WSE construeixi un perfil d'usuari amb les consultes sobre temes comuns a l'usuari, és a dir temes que l'usuari consulta vàries vegades i que per tant el WSE podria detectar,

cal distorsionar encara més el perfil enviant consultes sobre temes irrellevants per a l'usuari. Per evitar que aquestes consultes irrellevants siguin detectades per el WSE es guarden les $n-1$ consultes de la resta d'usuaris del grup en cada execució del protocol, i s'aniran reutilitzant de manera aleatòria per repetir consultes sobre temes aliens a l'usuari. Aquestes consultes s'han de enviar de manera que no siguin detectades pel WSE per causa de fer servir patrons temporals detectables. El més fàcil és enviar aquestes consultes cada vegada que l'usuari realitza una execució del protocol UUP.
Aquesta protecció no modifica el temps de resposta del protocol i és una protecció viable ja que actua en el últim pas, quan les consultes s'envien al WSE.

3.1.4 Especificació del nou protocol

En base a l'estudi dels apartats anteriors, es presenta a continuació l'especificació del protocol UUP en la versió per a terminals Smartphone.

Funcionament del protocol

Com a precondició, cada node ha de disposar d'un certificat digital propi. Aquest certificat ha de contenir un pseudònim i no hi ha d'haver cap dada que identifiqui l'usuari (nom real, dni, etc) o el terminal mòbil (número de telèfon, imei, etc).
Els nodes no poden rebre connexions directes de xarxa, per tant tota comunicació amb el node central s'inicia des de el propi node. Tota vegada que un node hagi de rebre una comunicació, ha de anar preguntant al node central fins que aquest tingui la informació disponible i li pugui respondre.

A Formació del grup d'usuaris
El node central està a l'espera de rebre peticions d'usuaris per formar un grup.

1 L'aplicació client de l'usuari U_i es comunica amb el node central per demanar ser inclosa en un grup. En aquesta comunicació li envia el seu certificat

2 Quan el node central té prous usuaris forma un grup. Tots els nodes que formen aquest nou grup reben la informació necessària per formar part de un grup d'usuaris

$\{U_1, ..., U_n\}$ del protocol UUP: l'identificador del grup, els identificadors dels nodes del grup i els certificats digitals de cada node. També rebrà els paràmetres per generar la clau de grup (nombre primer p i element $g \in Z_p^*$).

3 Quan els nodes reben la llista de identificadors i certificats comproven que no estan en un grup fantasma. Enviaran la seva decisió de continuar l'execució o abandonar el grup.

4 El node central envia a tots els nodes la confirmació de la creació del grup o el missatge d'abandó del grup.

A partir d'aquest punt, si l'execució tira endavant, el node central només actua com a enrutador de missatges o com a bústia de missatges. Els missatges que envia un usuari es faran arribar a tot el grup o a un usuari concret del grup.

B Generació de la clau de grup. Cada aplicació client de l'usuari U_i seguirà els passos:

1 Generació d'un número aleatori $a_i \in Z_q^*$ que serà la clau privada parcial d'aquest usuari.

2 Càlcul del valor $y_i = g^{a_i} \ mod \ p$.

3 Enviament a la resta d'usuaris del hash de la seva part de la clau pública amb firma digital (utilitzant el certificat propi).

4 Recepció de tots els resums criptogràfics (o hash) de la clau pública parcial de cada node. Aquests valors s'emmagatzemen per comprovar-los més endavant.

5 Enviament a la resta d'usuaris de la seva clau pública parcial y_i firmada digitalment amb el certificat propi.

6 Recepció de les claus públiques parcials de la resta d'usuaris del grup y_j per $j = (1, ..., n), j \neq i$. Cal verificar per cada clau parcial que la firma és vàlida i posteriorment calcular el resum criptogràfic per veure que dóna el mateix valor que en el pas B4, el qual s'ha emmagatzemat temporalment.

7 Càlcul de la clau pública de grup: $y = \prod_{1 \leq j \leq n} y_j = g^{a_1} \cdot g^{a_2} \cdot ... \cdot g^{a_n}$

C Anonimització de les consultes de cerca.

1 **Cada usuari U_i realitza el xifrat de la seva consulta m_i**

   i. **U_i genera un número aleatori r_i.**

   ii. **U_i encripta la seva consulta de cerca m_i amb la clau de grup y: $c_i^0 = E_y(m_i, r_i) = (g^{r_i}, m_i \cdot y^{r_i}) = (c1_i, c2_i)$**

2 **Per $i = (2, ..., n)$ cada usuari U_i envia el seu c_i^0 al primer membre del grup U_1.**

3 **Per $i = (1, ..., n-1)$ cada usuari U_i realitza les següents operacions:**

   i. **Rep la llista de consultes xifrades $\{c_1^{i-1}, ..., c_n^{i-1}\}$.**

   ii. **Utilitza la seva clau privada parcial per realitzar el desxifrat parcial de les consultes xifrades (segons [19] III-B) obtenint $\{c_1^{i-1'}, ..., c_n^{i-1'}\}$**

   iii. **La llista obtinguda es processa amb l'algoritme de "re-masking" de [19] II-A2 amb una clau $y' = \Pi_{j=i+1}^{n} g^{\alpha_j}$ obtenint $\{e_1^{i-1}, ..., e_n^{i-1}\}$.**

   iv. **Els elements de la nova llista són permutats aleatòriament obtenint $\{e_{\sigma(1)}^{i-1}, ..., e_{\sigma(n)}^{i-1}\}$.**

   v. **Envia la llista nova de consultes xifrades $\{c_1^i, ..., c_n^i\} = \{e_{\sigma(1)}^{i-1}, ..., e_{\sigma(n)}^{i-1}\}$ a l'usuari U_{i+1}.**

4 **L'últim usuari U_n realitza les següents accions:**

   i. **Rep la última llista de consultes xifrades $\{c_1^{i-1}, ..., c_n^{i-1}\}$.**

   ii. **Realitza el desxifrat parcial de la llista amb la seva clau privada parcial usant l'algoritme descrit a [19] III-B. En aquest punt U_n obté la llista de consultes en clar.**

   iii. **Re-envia la llista de consultes desencriptades obtinguda a tots els membres del grup $\{U_1, ..., U_{n-1}\}$.**

D **Enviament de les consultes al WSE i recepció del resultat. Cada usuari té la llista de consultes i les envia totes al WSE. Només es guarda el resultat de la seva consulta.**

Format dels missatges

S'ha decidit utilitzar la tecnologia XML per als missatges del protocol UUP. Per a cada pas del protocol estipulat en l'apartat anterior, s'ha creat un tipus de missatge. L'estructura interna dels missatges es modifica segons el pas del protocol ja que els camps que conté

són variables.
Tots els missatges contenen un node "UUPMessage" o un node contenidor "UUPMessages" que conté diversos nodes "UUPMessage". D'aquesta manera es poden transmetre en el mateix document XML diferents missatges que vagin destinats al mateix node. Això serà útil quan un node interroga al node central per veure si té missatges per ell i resulta que diferents nodes li han enviat missatges, tots aquests missatges s'enviaran amb la resposta Http en un XML.
Tots els nodes "UUPMessage" han de contenir un node que indica el pas del protocol, aquest node s'anomena "step" i un String amb el valor del pas del protocol.
A grans trets hi ha tres tipus de missatges: els de formació de grup, els de intercanvi de informació entre els nodes d'un grup i els missatges d'error. Els missatges de formació de grup estan formats per un node "joinRequest" o un node "joinResponse". Els missatges per enviar i rebre informació entre els nodes de un grup estan formats per nodes "groupMessage" per enviar o "getMessages" per rebre. Els missatges d'error estan formats pel node "errorMessage" o "Quit".

Pel que fa a la inclusió de dades encriptades i firmes digitals en els missatges, s'ha pensat en utilitzar dos estàndards: el XML-DSIG [3] per a les firmes digitals i el XML-ENC [2] per a les dades encriptades .
Quan un node envii el missatge que conté la seva clau pública en el pas B.5 o el missatge amb el resum criptogràfic de la seva clau pública en el pas B.3, el firmarà amb el seu certificat, i aquesta firma que s'inclou en el missatge XML amb un element "Signature" seguirà l'especificació de [3].
Pel que fa a l'enviament de les consultes encriptades, com que amb el sistema de clau pública ElGamal un missatge encriptat té dos components, no és adient utilitzar l'estàndard XML-ENC [2] ja que aquest preveu un missatge encriptat amb un sol component. De totes maneres es podria utilitzar una modificació d'aquest i introduir un missatge encriptat de dos components ja que aquesta aplicació de moment no ha de interaccionar amb cap sistema exterior. S'ha escollit no utilitzar XML-ENC.
Cal remarcar que no s'enviaran en el mateix missatge XML les claus o certificats per a desencriptar o verificar la informació tot i que l'especificació d'aquests dos estàndards ho permeti, ja que

el certificat de cada usuari s'ha obtingut prèviament. Per a l'enviament dels certificats dels usuaris en els passos de formació de grup (de A.1 a A.4) cada certificat s'enviarà dins d'un element "X509Data".

A l'annex B es detalla el format dels missatges per a cada pas del protocol.

Altres observacions

Aquesta versió del protocol UUP està pensada per a utilitzar una capa de transport que faciliti les coses al màxim possible.
Per una banda la capa de transport s'ha d'ocupar de tots els temes de detecció d'errors de transmissió, pèrdua d'informació, control de flux, control de seqüència i enrutament. Per tant l'ideal és utilitzar TCP/IP.
Per altra banda el fet d'utilitzar dispositius smartphone dels quals hem de suposar que la majoria no podran modificar la seva configuració, juntament amb el fet que l'objectiu del protocol és realitzar cerques a la web fan que l'ideal sigui utilitzar el protocol Http com a suport per a enviar el missatges en XML, tot utilitzant el port 80 i com a client (cada dispositiu inicia la comunicació ja que no pot rebre comunicacions externes actuant com a servidor).
A banda d'això s'ha de dir que, si bé la capa de transport TCP/IP s'encarrega de fer arribar els missatges des de els nodes fins al node central (i per tant s'encarrega dels errors de transmissió, el control de seqüència, etc), per fer arribar els missatges de un node a un altre és el propi protocol UUP que implementa mecanismes per a fer-ho. El protocol UUP també gestiona els errors que puguin succeir tant interna com externament com per exemple enviament de un missatge amb contingut il·legal o amb un valor del pas equivocat.

3.2 Arquitectura del sistema

En aquest apartat es fa una proposta d'arquitectura del sistema que executi el protocol Useless User Profile amb dispositius Android. Primerament es definiran les tecnologies bàsiques sobre les quals es construeix el projecte: protocol de comunicació, llenguatge de programació, servidor i persistència de dades. Seguidament es detalla el model de dades utilitzat. Després s'explica l'arquitectura i el funcionament bàsics del component software que realitza la tasca de node central. Finalment es defineix l'arquitectura i funcionament bàsics del component software que realitza la tasca de node en un dispositiu Android.

3.2.1 Tecnologies de base

Com hem vist en la secció anterior el protocol UUP es defineix en XML i s'adapta perfectament a un transport dels missatges per mitjà de peticions Http sobre TCP/IP.
Pel que fa als llenguatges de programació, els dispositius Android executaran el component client en una versió de Java per a dispositius Android. El component software que realitza la tasca de node central utilitzarà la tecnologia Java. Aquest component ha de fer de servidor web per atendre les peticions Http que porten missatges UUP en xml. Per tant s'utilitzarà la tecnologia Servlet de J2EE per al node central.
Per executar el servlet que farà de node central cal utilitzar un servidor web que sigui un contenidor de servlets. S'ha escollit utilitzar el servidor Apache Tomcat.
Per a la persistència de dades s'utilitzarà el gestor de bases de dades MySQL.

3.2.2 Model de dades

El model de dades del sistema és bastant simple. Es tracta de una classe "Group" que conté els paràmetres criptogràfics p i g, la qual té una relació de agregació amb la classe "User". La classe "User" conté la seva clau privada, la seva clau publica, la seva consulta de cerca, el seu certificat digital i a més està associada a diversos missatges representats per la classe "Message". La classe "Message" esta associada a 2 usuaris i conté un atribut amb el cos del missatge

o "content". A continuació es mostra el diagrama de classes.

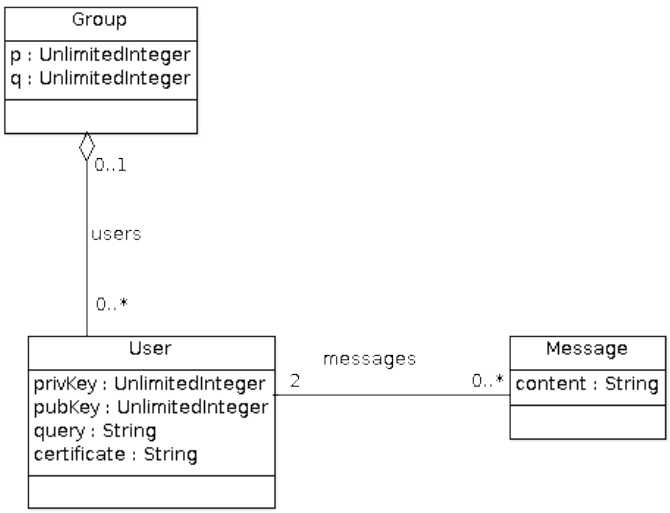

Figura 3.1: Diagrama de classes del domini

3.2.3 Casos d'ús

En aquest projecte el sistema disposa d'un sol cas d'ús: la realització d'una consulta de cerca a un motor de cerca web a través del

protocol Useless User Profile (UUP). Els actors que intervenen en aquest cas d'ús són els usuaris que formaran el grup de nodes en el protocol UUP i el motor de cerca web (WSE). Seguint aquesta visió el sistema està format per les aplicacions client en Android que fan de nodes i el servidor que fa de node central.

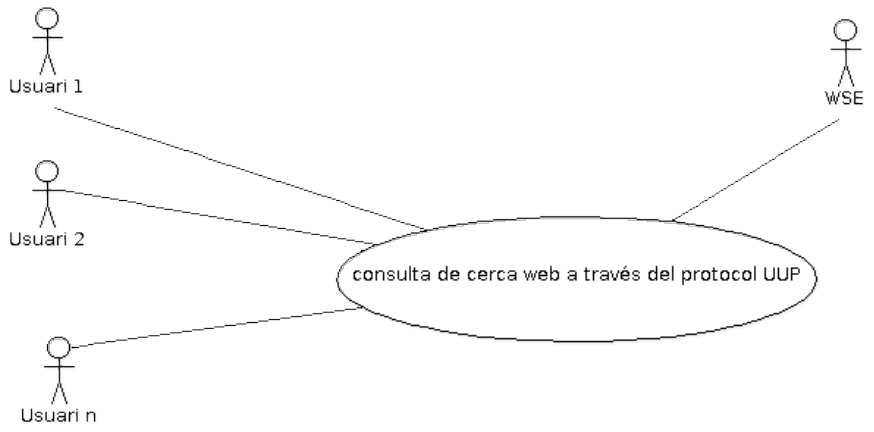

Figura 3.2: Diagrama de casos d'ús

També és important de cara al disseny del protocol considerar el node central com a entitat externa no controlable. Això ha ajudat a definir el model d'atacant i pensar en les proteccions a implementar.

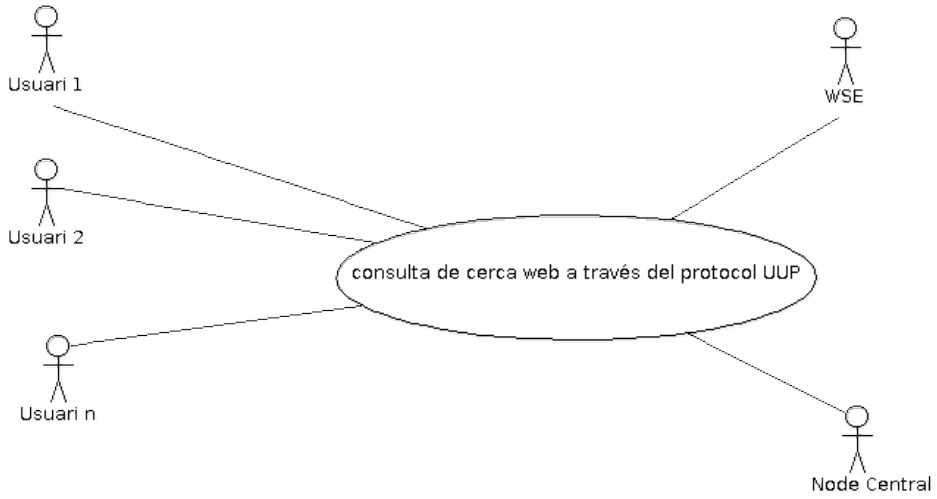

Figura 3.3: Diagrama de casos d'ús

3.2.4 Diagrames de seqüència

En aquest apartat es mostra el funcionament bàsic dels components del sistema.

Per una banda el Node Central està dissenyat com un servidor que aten peticions dels nodes. Per cada petició concreta i tenint en compte el pas del protocol al qual pertany, el node central realitzarà les accions adients. Hi ha dos tipus d'accions que pot realitzar el Node Central, les de formació de grups d'usuaris on la interacció és més gran, i les de recepció i distribució de missatges entre nodes. En tots els casos el Node Central farà un gran ús de la base de dades.

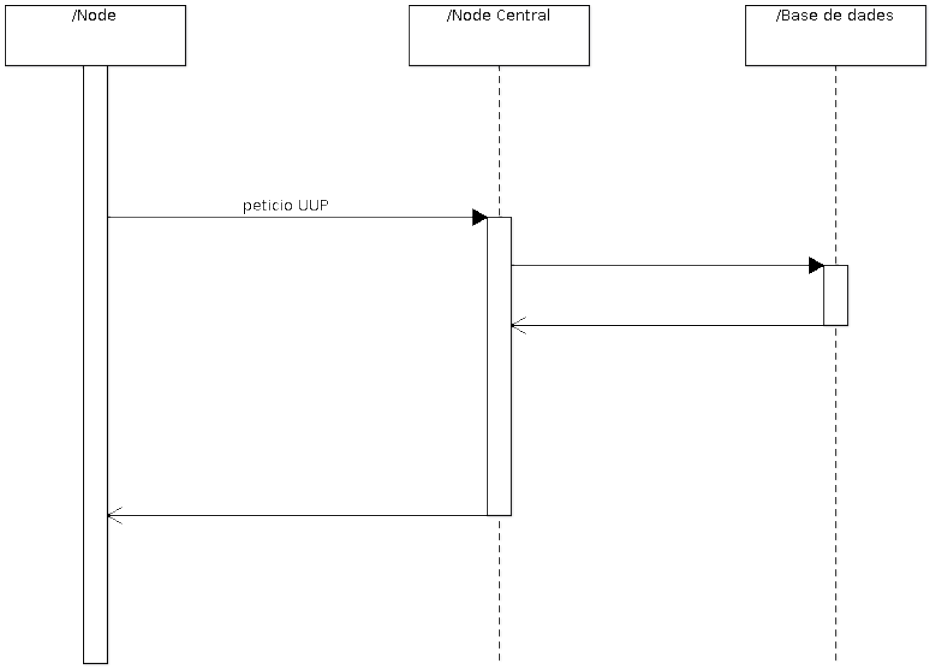

Figura 3.4: Diagrama de seqüència del servlet

L'aplicació client en Android que executen els nodes realitza tots els passos del protocol UUP. Per a això ha de interactuar amb el node central per a la formació de grups i per a enviar i rebre missatges d'altres nodes. Finalment interactua amb el motor de cerca WSE per obtenir els resultats de les cerques.

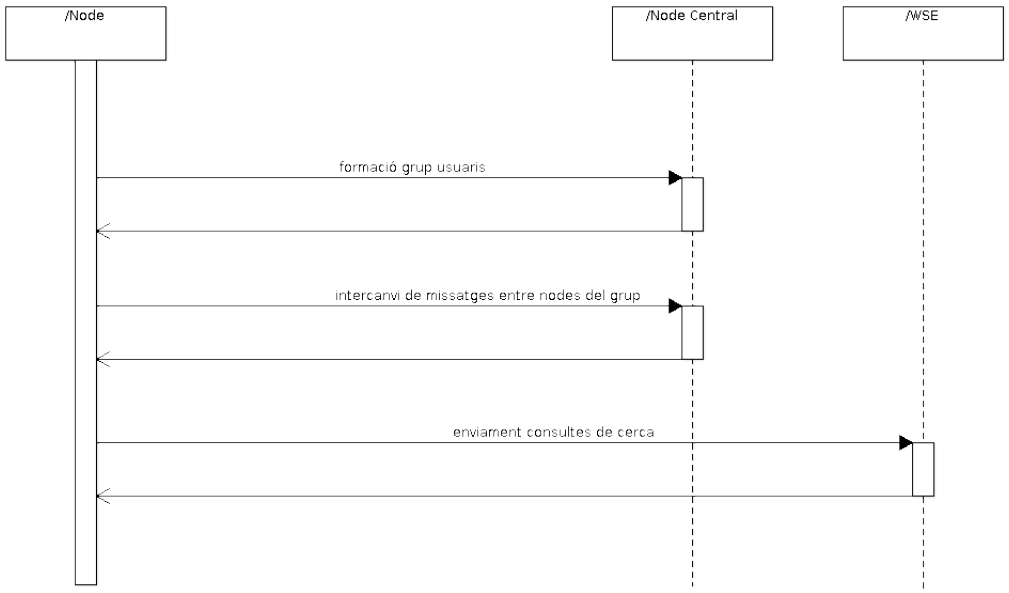

Figura 3.5: Diagrama de seqüència

Capítol 4

Implementació

Aquest capítol explica els detalls clau de la implementació d'aquest projecte. Primerament es mostra l'estructura i funcionament del Servlet que fa de node central en el protocol UUP. Després s'analitza l'estructura i funcionament de l'aplicació en Android que fa de node en el protocol UUP. Més tard es dedica un apartat a repassar la implementació de les operacions criptogràfiques del protocol UUP. També s'expliquen els detalls de la persistència de dades i finalment es fa un repàs de les llibreries emprades i les seves llicències.
Com és normal, al desenvolupar el codi que segueix l'especificació creada a la fase de disseny, es troben nous aspectes i dificultats no contemplades anteriorment. Per tant en aquesta fase poden sorgir modificacions del que ja s'ha estipulat anteriorment.
S'ha de dir també que a causa de la planificació a la qual està sotmès el treball hi ha proteccions del protocol que no s'han pogut implementar: la protecció contra grups fantasma i la protecció contra la detecció de temes repetits en les consultes de cerca. A resultes d'això, el format dels missatges XML que s'utilitza no fa ús dels elements previstos per a firmar digitalment o per enviar els certificats dels nodes. Es manté però una especificació i definició del format dels missatges compatible tant per a missatges que utilitzin aquests elements com per missatges que no els utilitzin.

4.1 Implementació del node central amb servlets

Per a implementar el node central s'ha desenvolupat un servlet que s'encarrega d'atendre totes les peticions dels nodes. Aquest servlet té dos funcions bàsiques que són organitzar els grups de nodes (crear-los, assignar els usuaris i desfer-los) i permetre la comunicació entre nodes (exercint com a bústia de missatges).

4.1.1 Diagrama de classes

El node central es composa de un Servlet que aten les peticions dels nodes anomenat "RegistrationServlet" i un Listener que s'encarrega de gestionar la caducitat de les sessions d'usuari (veure apartat "Gestió de les sessions d'usuari"). Aquests components utilitzen vàries classes per a realitzar diferents tipus de tasques. Aquestes classes es separen en packages. En el diagrama de classes següent es mostra la relació entre aquestes classes.

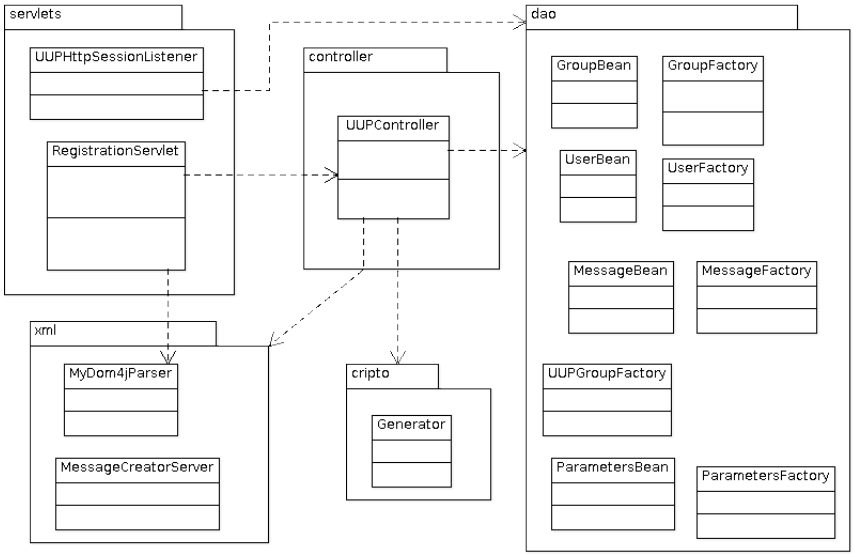

Figura 4.1: Diagrama de classes

Com es pot veure disposem de 5 packages diferents:

servlets amb les dues classes de servlets "RegistrationServlet" i "UUPHttpSessionListener"

controller amb la classe "UUPController" que aglutina totes les funcions de controlador de domini

xml amb les classes "MyDom4jParser" i "MessageCreatorServer" que s'encarreguen del tractament i generació de documents i missatges XML

cripto amb la classe "Generator" que realitza les operacions criptogràfiques

dao que conté les classes per a accedir al model del domini seguint el patró Data Access Objects(DAO). Es tracta de tenir un conjunt de objectes que fan de interfície i permeten realitzar les accions CRUD (Create, Read, Update i Delete) sobre la base de dades. Les classes que conté son "GroupBean", "GroupFactory", "UserBean", "UserFactory", "MessageBean", "MessageFactory", "ParametersBean", "ParametersFactory" i "UUPGroupFactory".

4.1.2 Diagrames de seqüència

El funcionament del servlet "RegistrationServlet" que rep les peticions dels nodes és molt senzill. Per a cada petició obté el pas del protocol del camp "step" del missatge XML. Segons el pas del protocol tractarà la resta del contingut adequadament i realitzarà les accions pertinents a la base de dades. A continuació es mostren dos diagrames de seqüència a mida d'exemple.

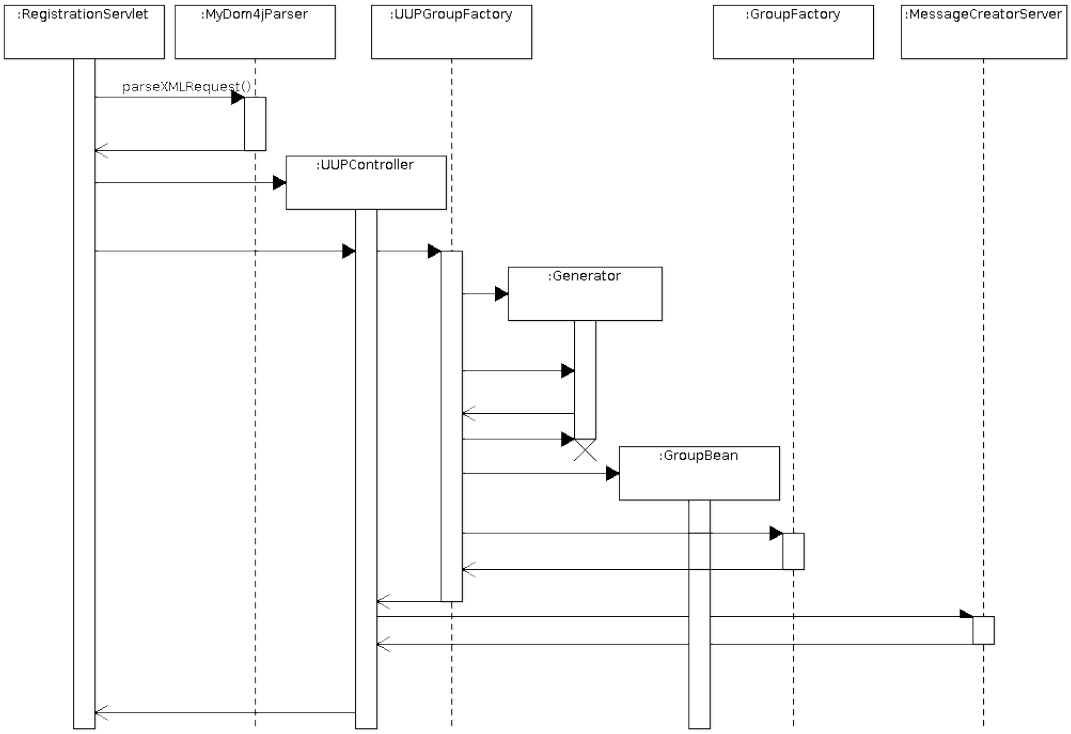

Figura 4.2: Diagrama de seqüència simplificat del funcionament del node central en el moment de crear un grup d'usuaris

En aquest diagrama de seqüència es pot veure com el servlet que ha rebut una petició Http que conté un missatge XML realitza el seu tractament amb la classe MyDom4jParser per a poder-lo manegar. Tot seguit crearà una instància del controlador UUPController passant-li el missatge XML i aquest realitzarà les accions necessàries per, en aquest cas, crear un nou grup d'usuaris. El controlador utilitzarà la factoria DAO UUPGroupFactory per a la creació del grup. Aquesta factoria genera els paràmetres criptogràfics i realitza les accions pertinents a la base de dades (només es mostra el tractament amb la factoria GroupFactory per simplificar). Finalment el controlador generarà un missatge XML com a resposta. Aquest missatge contindrà la llista de integrants del grup, amb els seus

certificats, el id del grup i els paràmetres criptogràfics. El missatge XML de resposta es retorna al servlet i aquest l'enviarà com a resposta al node.

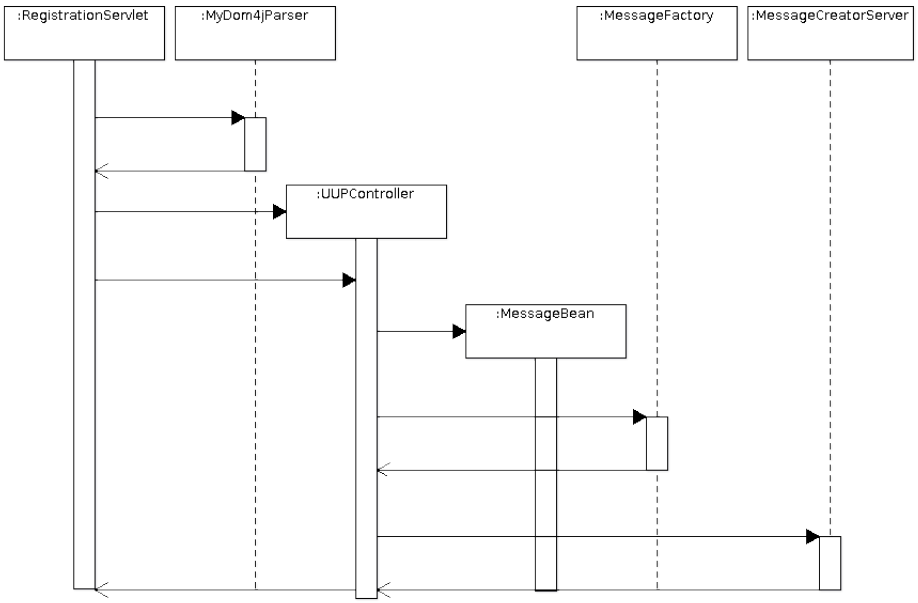

Figura 4.3: Diagrama de seqüència simplificat del funcionament del node central en el moment que un node envia un missatge

En aquest segon diagrama es mostra d'una manera simplificada la manera com es tracten els missatges que els nodes envien al node central per a que siguin transmesos més tard al node destí.

4.1.3 Gestió de les sessions d'usuari

Per a identificar els usuaris en el sistema hi ha diferents opcions. Es pot pensar en l'adreça IP però en el cas de nostre projecte això no és adient ja que els terminals mòbils poden estar en moviment i això pot fer canviar la seva IP (quan un terminal mòbil s'associa a una nova antena de telefonia la seva IP podria canviar). També es podria utilitzar alguna dada identificativa com el IMEI o el número de telefon. Això tampoc és aconsellable ja que per una banda son

dades identificadores que poden portar problemes de privacitat, i per altra banda s'han de considerar dispositius mòbils que no siguin telèfons mòbils també com les tablets.
Finalment s'utilitzen les sessions o Cookies de sessió per a identificar els usuaris en el sistema. Aquests identificadors caduquen tant bon punt el client ho decideixi (només cal que no re-envii la cookie de sessió) i així aquest té la llibertat de tornar a començar quan vulgui. Això també implica que un client pot abandonar el grup abans d'acabar tots els passos del protocol lo qual deixaria penjats a la resta d'usuaris. De totes maneres les execucions del protocol no estan pensades per a durar gaire i per tant això no és un problema greu.
El contenidor de servlets Tomcat ja gestiona les sessions d'usuari. A més a més el client en Android també té eines per a gestionar les sessions automàticament.
Pel que fa a la caducitat de les sessions el servlet permet configurar quan tarden a caducar les sessions d'usuari inactives. Per gestionar les sessions inactives caducades s'ha implementat un Listener de tipus SessionListener que actua en el moment de eliminar una sessió d'usuari (mecanisme gestionat automàticament per el contenidor de servlets). En el moment de esborrar una sessió d'usuari el SessionListener s'encarrega de eliminar de la base de dades l'usuari associat a aquesta sessió i però també el grup al qual pertany i tots els integrants d'aquell grup.

4.1.4 Paràmetres de configuració

El servlet permet configurar per mitjà del fitxer web.xml diferents paràmetres que afecten al seu funcionament.
Per una banda es pot ajustar el número de usuaris que accepta cada grup. Aquest paràmetre és important ja que com s'ha vist a [19] es pot trobar la mida adient per al menor temps d'execució del protocol.
Per altra banda es pot ajustar el temps que tarden en caducar les sessions inactives. Aquest temps normalment ha de ser baix ja que el protocol ha de tardar poc en executar-se. Però segons el cas caldrà ajustar-ho. Si la mida en bits dels paràmetres criptogràfics creix les operacions criptogràfiques tarden més en realitzar-se i aquest increment s'ha de multiplicar per el nombre d'usuaris d'un grup. Si el nombre d'usuaris del grup creix també augmentarà el

temps total d'execució.
Finalment es pot ajustar la mida en bits del paràmetre criptogràfic p utilitzat com a base per a tots els càlculs criptogràfics relacionats amb els passos del protocol UUP.

4.1.5 Components

La resta de classes que utilitza el servlet són les que permeten tractar documents XML, realitzar operacions criptogràfiques, gestionar la capa de persistència de dades i escriure un log del funcionament de l'aplicació.
Per al tractament de XML el servlet utilitza les classes "MyDom4jParser" i "MessageCreatorServer". Aquestes classes del package "xml" utilitzen la llibreria Dom4j per al parseig o transformació de document XML en objectes en memòria tot tenint en compte la definició XML Schema que s'ha fet dels missatges del protocol UUP en el fitxer "uup.xsd". El servlet utilitza aquest fitxer de tipus XML Schema definition per a comprobar la validesa dels missatges xml del protocol UUP en la versió del treball. Aquest fitxer es defineix a l'apèndix B i ha de ser accessible des del servlet. També permet manipular i modificar aquests documents. La classe "MessageCreatorServer" s'encarrega de formar els missatges XML de resposta.
Per a les operacions criptogràfiques s'ha creat la classe "Generator" la qual es basa en la classe "BigInteger" de "java.math". Aquest classe Generator permet generar el paràmetre enter i primer p i el paràmetre g. Tot això s'explica en l'apartat Criptografia més endavant.
Per a la persistència de dades s'ha escollit utilitzar el patró de disseny DAO (Data Access Objects) i l'accés a la base de dades MySQL per mitjà de JDBC.
Per a la generació d'un log del funcionament de l'aplicació s'utilitzen les llibreries "slf4j" i "log4j". La primera es una façana o interface genèrica que utilitza l'aplicació. Aquesta es configura per a que utilitzi les llibreries de log4j tot i que permet utilitzar altres llibreries de logging sense modificar el codi de l'aplicació.

4.2 Implementació del client UUP en Android

Per a la implementació del client UUP s'ha creat una aplicació Android que es basa en una sola "Activity". Tal com el seu nom indica les activities en Android son activitats que l'usuari pot realitzar. Usualment una activity interacciona amb l'usuari creant una finestra on es poden desplegar els elements de la UI (User Interface). Per tant l'aplicació client UUP es basarà en una sola finestra on l'usuari interacciona introduint la seva consulta de cerca. Internament aquesta activity utilitza diferents classes per a realitzar totes les tasques necessàries per a seguir el protocol UUP (connexió de xarxa, tractament de XML, operacions criptogràfiques, etc).

4.2.1 Diagrama de classes

A continuació es mostra el diagrama de classes per a veure les relacions entre les classes que conformen el sistema. L'activity "MainActivity" instància un objecte de la classe UUPController i aquest fa totes les tasques de controlador utilitzant diferents classes per a cada tasca. També hi ha una activity "SettingsActivity" per a que l'usuari pugui modificar detalls del funcionament.

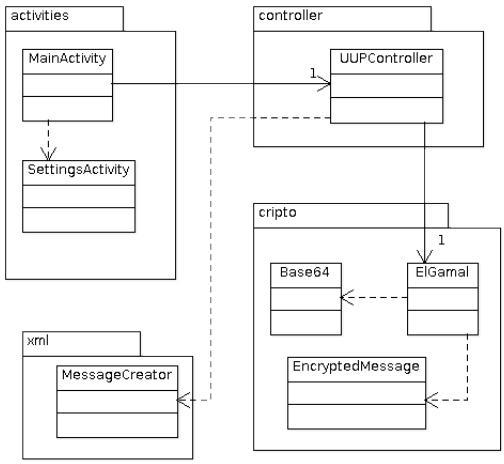

Figura 4.4: Diagrama de classes

Es disposa de 4 packages diferents:

activities amb les dues activities "MainActivity" i "SettingsActivity"

controller amb la classe "UUPController" que aglutina totes les funcions de controlador de domini

xml amb la classe "MessageCreator" que s'encarrega de la generació de missatges XML

cripto amb la classe "ElGamal", "Base64" i "EncryptedMessage" que realitzen les operacions criptogràfiques

El controlador de domini interacciona amb altres classes que no apareixen al diagrama de classes. Són classes que pertanyen a llibreries del sistema o a llibreries externes (com per exemple les classes per a gestionar connexions de xarxa o enviar dades per xarxa).

4.2.2 Diagrames de seqüència

El funcionament del programa consta de, per una part la activity "MainActivity" que s'encarrega de la interacció amb l'usuari, i per altra el controlador i la resta de classes que utilitza. Des de la activity, quan l'usuari dona l'ordre es comença l'execució del controlador. Aquest s'executa en un fil d'execució separat per tal de no bloquejar la interfície d'usuari.
El funcionament del controlador "UUPController" implementa una màquina d'estats. El controlador té una funció principal que és un bucle. Una variable guarda el valor de l'estat i en cada execució del bucle i segons els resultats de les operacions aquest estat es modifica. Els estats son els mateixos que els passos del protocol UUP. L'execució a dins del bucle s'aturarà quan la variable d'estat prengui el valor de finalització ("F") o de fallada ("ABORT").
A continuació es mostren dos diagrames de seqüència simplificats.

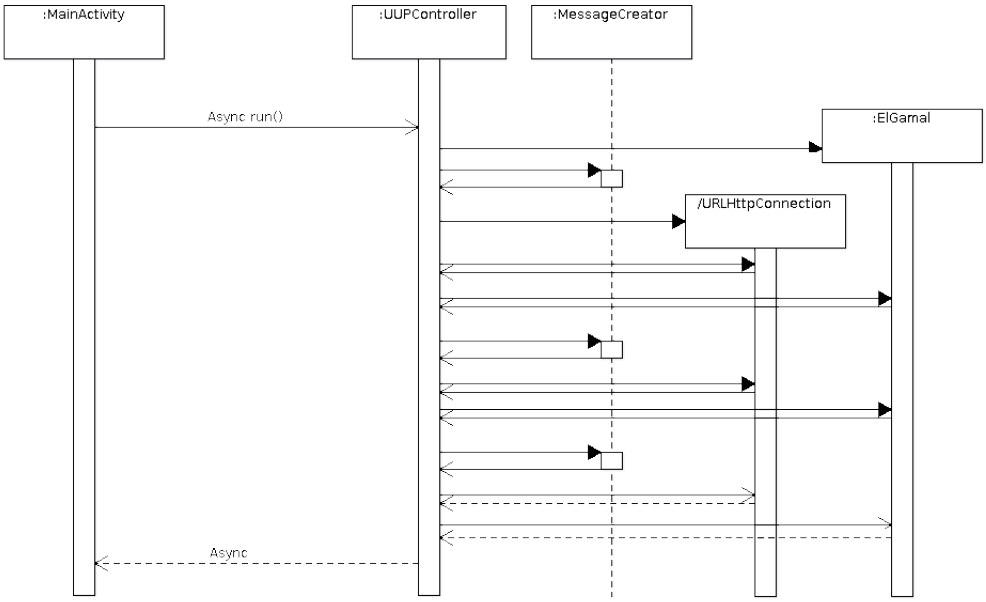

Figura 4.5: Diagrama de seqüència simplificat del funcionament del client UUP en Android

Aquest diagrama simplificat mostra el funcionament bàsic del client UUP en Android. En aquest cas el bucle que implementa la màquina d'estats del controlador s'ha representat desenrotllat.

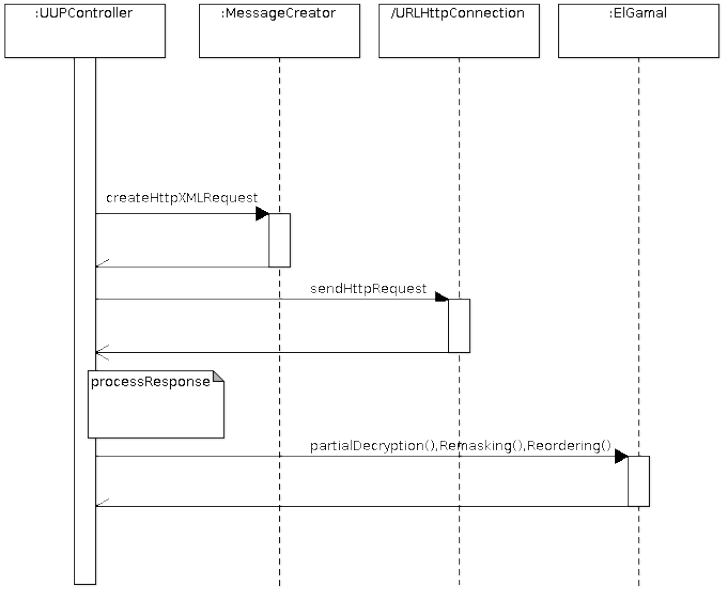

Figura 4.6: Diagrama de seqüència simplificat del funcionament del client UUP en Android durant un dels passos del protocol

En aquest diagrama de seqüència es mostra molt resumidament l'execució de una iteració del bucle del controlador. La seqüència d'accions és sempre la mateixa:

1. **hi ha primer unes accions prèvies,**
2. **seguidament es forma la petició Http amb el missatge XML que correspongui a l'estat actual,**
3. **després s'envia la petició i es rep la resposta (les respostes es guarden en una cua FIFO ja que això permetrà processar la recepció de més d'un missatge XML)**
4. **finalment es processa la resposta i s'actualitza la variable d'estat**

Cal afegir que el controlador disposa de dos paràmetres modificables que influeixen en el comportament de l'aplicació.
El primer és el número de repeticions que es poden arribar a fer dins del bucle de la màquina d'estats quan una operació dona error o cal esperar que arribin missatges per al node. Aquest paràmetre és important i cal que sigui gran ja que quan hi ha molts usuaris

o el nombre de bits de la clau criptogràfica és gran cal esperar un cert temps a que les respostes dels altres nodes arribin.
El segon és el temps d'espera abans de re-enviar una petició http, aquest paràmetre pot fer que el protocol funcioni més ràpidament però s'ha de vigilar de no saturar el servidor amb peticions massa seguides.

4.2.3 Components Android

En aquest sub apartat s'expliquen breument els components del sistema Android que formen l'aplicació client UUP. Es detallen les activities que composen l'aplicació, la utilització de la classe "AsyncTask" per a realitzar accions asíncrones, la utilització de "Handlers" per a comunicar entre fils d'execució , el menú de configuració o "Settings" la comunicació entre activities per mitjà de "Intents", i finalment la interfície d'usuari.

Com ja s'ha comentat les activities en Android son la base de les aplicacions i van lligades usualment a les interfícies d'usuari. Aquesta aplicació es basa en una sola activity "MainActivity" tot i que per mostrar el menú de configuració de l'aplicació s'utilitza una altra activity anomenada "SettingsActivity". La activity principal "MainActivity" s'encarrega de interaccionar amb l'usuari i llançar l'execució del controlador que executarà el protocol UUP.
Com que l'execució del protocol no és immediata, aquesta s'haurà de realitzar en un altre fil d'execució. Per això la classe "MainActivity" té una classe interna que hereta de "AsyncTask". Una classe de tipus AsyncTask ha de implementar dos mètodes: doInBackground() i onPostExecute(). Per a executar les tasque s'ha de cridar al mètode execute() d'aquesta classe, el qual ja s'encarregarà de cridar als mètodes doInBackGround() i onPostExecute(). És doncs en aquesta classe on s'executarà el controlador de domini UUPController.
Durant l'execució del protocol UUP, per a que el controlador es comuniqui amb l'activity principal fa falta utilitzar un mecanisme anomenat "Handler". El "Handler" es crea a dins de l'activity i se li assigna un mètode per a gestionar els missatges rebuts. Aquest objecte handler es passa al controlador per a que durant l'execució el faci servir per enviar missatges. Aquest mecanisme s'ha utilitzat per a actualitzar la finestra de log de la interfície gràfica, indicant a cada iteració del bucle l'estat i els possibles errors.

La activity "SettingsActivity" s'encarrega de mostrar uns menús per a configurar el funcionament del client UUP. Aquesta permet introduir la URL del servidor node central, modificar el número de repeticions de un pas dins del bucle de la màquina d'estats i seleccionar el cercador que es vol utilitzar.
Per a comunicar entre diferents activities i per a canviar de activity cal utilitzar un mecanisme anomenat "Intent". Aquest permet activar altres activities i fins i tot altres aplicacions o enviar dades.
La interfície gràfica és molt senzilla. Disposa de:

1 un camp de text editable on es mostra la URL del servidor node central

2 un camp de text editable per a introduir la consulta de cerca

3 un botó per a començar l'execució del protocol UUP

4 un menú desplegable per escollir el motor de cerca o WSE a utilitzar

5 un camp de text multi-línia per mostrar els missatges de log de l'aplicació

6 un component "WebView" per a mostrar el resultat de la cerca

4.2.4 Altres components

La resta de classes que utilitza l'aplicació client UUP són les classes per a realitzar operacions criptogràfiques i les classes per a tractar els missatges XML.
En el package "cripto" hi ha les classes Base64, ElGamal i EncryptedMessage. La classe Base64 serveix per a realitzar la codificació i descodificació en base64 de cadenes de bytes per a representar-les en format String. La classe EncryptedMessage serveix per a encapsular un missatge encriptat amb l'algoritme ElGamal ja que aquest està format de dos components. La classe ElGamal concentra totes les operacions criptogràfiques que necessita realitzar el client UUP i s'expliquen en el següent apartat. Tot el seu funcionament es basa en la classe BigInteger de Java.
Pel que fa al tractament de XML el package "xml" disposa de la classe MessageCreator que serveix per a crear els missatges XML adients per al protocol UUP. El tractament de XML en el client UUP es realitza amb la llibreria DOM de Java.

4.3 Criptografia

En l'aplicació servlet que fa de node central es generen els paràmetres p i g. En l'aplicació client es generen les claus pública i privada i també es realitzen les operacions de xifrat, desxifrat parcial i remasking. Tota la implementació està basada en la classe BigInteger de Java. A l'apèndix C es mostren tots els detalls de la implementació.

4.4 Persistència de dades

4.4.1 Model relacional de dades

Un cop aplicat el procés de normalització (en aquest cas afegir OID's independents de les dades i mapejar les associacions a claus foranes), i modificacions de última hora, el model de dades queda transformat en un model relacional de dades. Tenim com a resultat les tres taules que es defineixen a continuació.

La taula amb informació sobre els grups:

Groups	
id	identificador enter clau primària
numusers	enter que indica el numero de usuaris que composen el grup
available	booleà que indica si un grup esta lliure per a ser utilitzat novament
numaccepted	enter que serveix per saber quants usuaris han acceptat el grup
p	Text paràmetre criptogràfic
g	Text paràmetre criptogràfic

La taula amb informació sobre els usuaris:

Users	
id	identificador enter clau primària
session	String que identifica l'usuari durant la sessió al servidor web
groupnum	enter clau forana que indica a quin grup pertany l'usuari
usergroupnum	enter que identifica l'usuari o node a dins del grup al qual pertany
groupAccepted	enter que indica si l'usuari ha acceptat o rebutjat el grup

La taula amb informació sobre els missatges que s'intercanvien els usuaris:

Messages	
id	identificador enter clau primària
contents	String amb el contingut del missatge
sourceiduser	enter clau forana que indica quin usuari és l'emissor
destiduser	enter clau forana que indica quin usuari és el destinatari

La taula amb paràmetres criptogràfics p calculats a priori:

parameters	
id	identificador enter clau primària
p	Text amb el contingut del nombre primer p

Cal afegir que per raons de flexibilitat, la columna "groupnum" de la taula usuaris finalment no es defineix com a clau forana tot i que encara ho és.

Capítol 5

Test

Per testejar l'aplicació s'han realitzat diverses execucions de la versió del protocol Useless User Profile del present treball en diferents escenaris. Per cada escenari s'ha anat variant el número d'usuaris per grup entre 2 i 4 usuaris. També s'ha provat amb els nodes i el node central en la mateixa xarxa local o amb alguns nodes a través de Internet.

	dispositius												
Test	A	B	C	D	A	B	C	D	A	B	C	D	Mitjana
2 nodes - LAN	19	29	-	-	20	-	23		14	-	14	-	20
2 n - 3G/LAN	18	-	19	-	11	21	-	-	17	-	18	-	17
2 n - 3G	29	-	-	45	18	-	-	30					31
3 nodes - LAN	35	-	37	37	37	47	35	-	20	30	21	-	33
3 n - 1 3G/ 2 LAN	22	34	22	-	30	44	32	-	30	41	30	-	32
3 n - 2 3G/1 LAN	39	65	-	37	27	37	-	27	30	40	-	35	33
4 nodes - LAN	49	63	57	56	38	47	37	40	49	55	45	53	49
4 n - 1 3G/ 3 LAN	52	70	50	54	52	62	54	54					56

A HTC Sensation Android 4.0.3,Dual Core 1.2GHz

B Airis OnePad 720 Android 2.3 ,ARM11 iMAPx210 fins a 1.00 GHz

C emulador Android 4.0.3 sobre Mac OS X, Core i5 Intel 1.8GHz (2.8Ghz)

D emulador Android 4.0.3 sobre Linux, Dual Core 2.2GHz

A continuació es mostra un gràfic resum dels diferents tests executats amb diferent nombre d'usuaris i diferent longitud de clau. Tot i que no es pot considerar una longitud de clau de 1024 bits

segura, s'han realitzat proves amb aquesta longitud per tal de veure com afecta aquesta mesura i també per a poder comparar amb els treballs de base del protocol UUP. A l'apèndix E es mostren tots els gràfics comparatius entre diferent nombre d'usuaris i diferent longitud de clau.

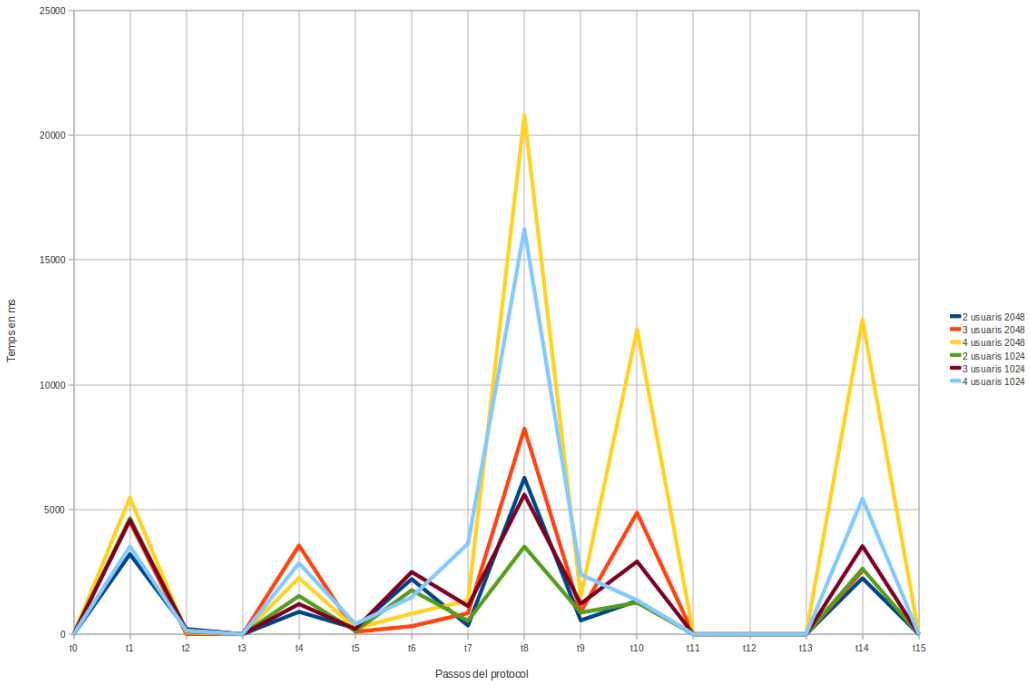

Figura 5.1: Temps dels passos del protocol per execucions amb $n = 2$, 3 i 4 usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Interneti la resta a la mateixa xarxa local que el node central

Observacions

La principal observació és que el temps de resposta d'aquest sistema no és mai menor de 10 segons. Comparant amb els resultats obtinguts en [19] es pot dir que aquesta implementació no obté un bon resultat. Si bé es pot considerar que un usuari pot ser capaç d'esperar-se 10 segons per obtenir un resultat en un cercador web, la majoria de vegades l'execució del protocol en aquest sistema dura més, lo qual es pot fer pesat per un usuari.

Els passos del protocol que més penalitzen són:

1 t8 i t10, rebre la llista de consultes i realitzar les operacions criptogràfiques (desxifrat parcial, remasking i reordenació) o esperar a rebre la llista de consultes en clar. S'acostuma a repartir entre aquests dos passos el temps màxim.

2 t14, consulta al WSE de totes les consultes de la llista.

3 t1, formació del grup

4 t4, intercanvis de claus públiques parcials i els seus resums criptogràfics

El pas que inclou els càlculs criptogràfics és el que més penalitza. Això és degut en bona part a les plataformes sobre les quals s'executa el client UUP. Es nota molt la plataforma hardware sobre la qual s'executa l'aplicació client i en aquest cas els 4 dispositius tenien processadors de velocitats diferents i no gaire potents. Per exemple en el cas de 2 nodes per Internet només s'ha pogut provar amb un dels dispositius més lents.
També s'ha de remarcar que el fet de no poder establir connexions directes entre els nodes fa créixer el temps de resposta. Aquest aspecte es podria millorar ajustant els paràmetres de temps entre peticions successives.

Cal dir que els temps obtinguts en el dispositiu Android 2.3 son sempre més elevats (de l'ordre de 5 o 10 segons més) a causa d'un bug en el component Webview per a aquesta versió de la plataforma. Aquest bug obliga a recórrer el contingut HTML que es pretén mostrar en el widget Webview i modificar certs caràcters com el "%" i el "#" transformant-los en el seu equivalent html entity [1].

Es podria millorar una mica el resultat si el contingut html de la consulta al wse (per la consulta bona) es passés al component Webview abans de que hagin acabat totes les consultes al wse. De totes maneres això acceleraria el procés com a màxim un segon, i per tant és despreciable en front els temps associats als càlculs criptogràfics.

Capítol 6

Conclusions

El desenvolupament de una versió del protocol Useless User Profile per a dispositius mòbils ha resultat molt interessant, sobretot la implementació del sistema de comunicacions centralitzades i tota la criptografia.

Els resultats obtinguts tenen uns temps de resposta superiors als obtinguts en els estudis [8] i [19], tot i que en aquests es mostren els resultats amb longituds de clau de 1024 bits només. Avui en dia però, una clau de 1024 bits ja no és prou segura i cal utilitzar 2048 bits.
El fet d'executar el protocol amb dispositius mòbils fa que els processadors de què es disposa puguin ser poc potents i això implica elevar el temps de resposta considerablement. Una altra causa és el fet prou comú en l'àmbit dels dispositius mòbils que aquests no puguin rebre connexions directes de xarxa. Això implica molt més trafic de xarxa ja que totes les comunicacions estan centralitzades en el node central, i no només això sinó que a més cal que cada dispositiu interrogui al node central per a rebre missatges dels altres nodes.
De tot això en resultat que la usabilitat es veu perjudicada. De totes maneres el compromís entre usabilitat i privacitat segueix sent millor envers la resta d'alternatives.

Malauradament no s'ha pogut complir amb la planificació inicial i la implementació de la protecció contra la formació de grups fantasma no s'ha implementat.

Com a ampliacions d'aquest treball cal mencionar les protecci-

ons que no s'han desenvolupat. Aquestes són les proteccions contra l'atac "Input-replacement", els mecanismes per a dificultar la identificació per el WSE de temes repetits en les consultes de cerca i la protecció contra la creació de grups fantasma que no s'ha implementat per manca de temps.

Personalment la valoració general del projecte és molt satisfactòria tant per les coses apreses com per tot el que s'ha pogut gaudir des de el punt de vista tècnic durant la realització d'aquest.

Apèndix A

Nocions prèvies

A.1 Criptografia

El protocol Useless User Profile es basa en el criptosistema de clau pública ElGamal, en la seva versió per a més de dos usuaris (criptografia orientada a grups o threshold-cryptography). A continuació s'expliquen els detalls d'aquest criptosistema.

A.1.1 n-out-of-n threshold ElGamal encryption

Segons [10] , en aquest criptosistema n usuaris comparteixen una clau pública y. La clau privada corresponent α està dividida en n claus privades parcials. Amb aquest protocol, un missatge m es pot encriptar utilitzant la clau pública y i només es pot desencriptar si tots els n usuaris col·laboren en el procés ja que es necessiten totes les claus privades parcials. La generació de claus i els processos de xifrat i desxifrat s'expliquen a continuació:

1. **Generació de claus**
   Primer de tot es genera un nombre primer p aleatori i prou gran, tal que $p = 2q + 1$ on q també és un nombre primer. També s'escolleix un generador g del grup multiplicatiu Z_q^*. Després cada usuari genera una clau privada aleatòria $\alpha_i \in Z_q^*$ i comparteix amb la resta d'usuaris la seva clau privada parcial $y_i = g^{\alpha_i}$. La clau publica comú, és a dir la clau pública del grup, es calcula com $y = \prod_{i=1}^{n} y_i = g^{\alpha}$ on $\alpha = \alpha_1 + ... + \alpha_n$.

2. **Operació de xifrat**
   La operació de xifrat es du a terme amb la funció estàndard

de xifrat ElGamal [13]. Donat un missatge m i una clau pública y, es genera un valor aleatori r i el text xifrat es calcula com:

$$E_y(m,r) = c = (c1, c2) = (g^r, m \cdot y^r)$$

3 Operació de desxifrat
Donat un missatge xifrat amb una clau pública y, $E_y(m,r) = (c1, c2)$, l'usuari U_i pot desxifrar aquest valor de la següent manera:

a cada usuari $j \neq i$ publica $c1^{\alpha_j}$

b l'usuari U_i recupera el missatge m com:

$$m = \frac{c2}{c1^{\alpha_i}(\Pi_{j \neq i} c1^{\alpha_j})}$$

El resultat es pot verificar per cada usuari realitzant una prova d'equivalència del logaritme discret.
Aquesta operació s'utilitza en la versió inicial del protocol UUP de [8].

4 Operació de desxifrat parcial
El missatge xifrat abans de qualsevol desxifrat parcial és $E_y(m,r) = (c1, c2)$, i està xifrat amb una clau pública $y = g^{\alpha_1 + ... + \alpha_n}$, on α_i és la clau privada generada per l'usuari U_i. Per realitzar un desxifrat parcial de $E_y(m,r) = (c1, c2)$ l'usuari U_i utilitza la seva clau privada de la següent manera:

$$c2' = \frac{c2}{c1^{\alpha_i}}$$

El resultat d'aquesta operació s'utilitza per construir un nou missatge xifrat $E_{y'}(m,r) = (c1, c2')$. En aquest cas, el missatge xifrat està encriptat amb una clau pública $y' = g\alpha_{i+1} + ... + \alpha_n$. Amb aquesta operació, els usuaris poden contribuir individualment al desxifrat de un missatge sense haver de intercanviar els valors $c1^{\alpha_i}$. Quan un usuari ha desxifrat parcialment el missatge, el missatge xifrat resultant ja no està encriptat sota la seva clau privada. Amb la col·laboració de tots els usuaris, el missatge es pot desxifrar totalment.
Aquesta operació de desxifrat parcial és la que s'utilitza en la versió de protocol UUP de [19] i en el present treball.

A.1.2 ElGamal re-maskig

La operació de "Re-masking" realitza certes transformacions sobre un text xifrat. D'aquesta manera, el contingut del text original que correspon al resultat de l'operació de "Re-masking" no canvia però el missatge re-emmascarat o "Re-masked" no es pot relacionar amb el text xifrat de partida.
Donat un text xifrat amb ElGamal $E_y(m, r)$, l'operació de "Re-masking" sobre aquest es computa de la següent manera [4]:

$$E_y(m, r) \cdot E_y(1, r')$$

Per $r' \in Z_q^*$ escollit aleatòriament i amb $\cdot$ el producte escalar per components o producte de Hadammard. El text xifrat resultant d'aquesta operació dona com a resultat un cop desxifrat el mateix text en clar m.

A.2 El protocol Useless User Profile

En aquesta secció es descriu el protocol Useless User Profile de [19]. S'utilitza el terme aplicació client UUP, node o usuari per referir-se indistintament a un usuari executant l'aplicació client del protocol UUP en un terminal smartphone.

A Formació del grup d'usuaris

1. El node central està a l'espera de rebre peticions d'usuaris per formar un grup.
2. L'aplicació client de l'usuari U_i es comunica amb el node central per demanar ser inclosa en un grup.
3. Quan el node central té prous usuaris forma un grup. Tots els nodes que formen aquest nou grup reben la informació necessària per formar part de un grup d'usuaris $\{U_1, ..., U_n\}$ del protocol UUP. Cada usuari rep un missatge amb les adreces IP i el port corresponent de la resta de membres del grup per tal de poder establir un canal de comunicació amb cada membre. Després d'aquest pas, els nodes poden enviar-se missatge directament i el node central ja no és necessari.

Com a millora, el node central també enviarà els paràmetres per generar la clau de grup (nombre primer p i element $g \in Z_q^*$), evitant d'aquesta manera que els membres del grup s'hagin de posar d'acord per decidir aquests valors.

B Generació de la clau de grup. Cada aplicació client de l'usuari U_i seguirà els passos:

1. **Generació d'un número aleatori $a_i \in Z_q^*$ que serà la clau privada parcial d'aquest usuari.**
2. **Càlcul del valor $y_i = g^{a_i} \ mod \ p$ que serà la clau pública parcial d'aquest usuari.**
3. **Enviament a la resta d'usuaris de la seva clau pública parcial y_i.**
4. **Recepció de les claus públiques parcials de la resta d'usuaris del grup y_j per $j = (1, ..., n), j \neq i$.**
5. **Càlcul de la clau pública de grup: $y = \prod_{1 \leq j \leq n} y_j = g^{a_1} \cdot g^{a_2} \cdot ... \cdot g^{a_n}$**

C Anonimització de les consultes de cerca.

1. **Cada usuari U_i realitza el xifrat de la seva consulta m_i**
   - **i. U_i genera un número aleatori r_i.**
   - **ii. U_i encripta la seva consulta de cerca m_i amb la clau de grup y: $c_i^0 = E_y(m_i, r_i) = (g^{r_i}, m_i \cdot y^{r_i}) = (c1_i, c2_i)$**
2. **Per $i = (2, ..., n)$ cada usuari U_i envia el seu c_i^0 al primer membre del grup U_1.**
3. **Per $i = (1, ..., n-1)$ cada usuari U_i realitza les següents operacions:**
   - **i. Rep la llista de consultes xifrades $\{c_1^{i-1}, ..., c_n^{i-1}\}$.**
   - **ii. Utilitza la seva clau privada parcial per realitzar el desxifrat parcial de les consultes xifrades (segons [19] III-B) obtenint $\{c_1^{i-1'}, ..., c_n^{i-1'}\}$**
   - **iii. La llista obtinguda es processa amb l'algoritme de "remasking" de [19] II-A2 amb una clau $y' = \Pi_{j=i+1}^{n} g^{\alpha_j}$ obtenint $\{e_1^{i-1}, ..., e_n^{i-1}\}$.**

iv. **Els elements de la nova llista són permutats aleatòriament obtenint** $\{e^{i-1}_{\sigma(1)}, ..., e^{i-1}_{\sigma(n)}\}$**.**

v. **Envia la llista nova de consultes xifrades** $\{c^i_1, ..., c^i_n\} = \{e^{i-1}_{\sigma(1)}, ..., e^{i-1}_{\sigma(n)}\}$ **a l'usuari** U_{i+1}**.**

4 **L'últim usuari** U_n **realitza les següents accions:**

i. **Rep la última llista de consultes xifrades** $\{c^{i-1}_1, ..., c^{i-1}_n\}$**.**

ii. **Realitza el desxifrat parcial de la llista amb la seva clau privada parcial usant l'algoritme descrit a [19] III-B. En aquest punt** U_n **obté la llista de consultes en clar.**

iii. **Re-envia la llista de consultes desencriptades obtinguda a tots els membres del grup** $\{U_1, ..., U_{n-1}\}$**.**

D **Enviament de les consultes al WSE i recepció del resultat. Cada usuari té la llista de consultes i les envia totes al WSE. Només es guarda el resultat de la seva consulta.**

Apèndix B

Format dels missatges

Aquest apèndix detalla l'especificació del format dels missatges del protocol UUP implementat en aquest projecte. Primerament es mostren exemples de missatges en cada pas del protocol i finalment s'adjunta l'especificació en format XML Schema del format d'aquests missatges.

B.1 Especificació XML Schema

En aquesta secció es mostren, per a cada pas del protocol UUP del projecte, un exemple de missatge XML corresponent.
S'ha decidit utilitzar la tecnologia XML per als missatges del protocol UUP. Per a cada pas del protocol estipulat en el capítol de disseny s'ha creat un tipus de missatge. L'estructura interna dels missatges es modifica segons el pas del protocol ja que els camps que conté són variables.

Tots els missatges contenen un node "UUPMessage" o un node contenidor "UUPMessages" que conté diversos nodes "UUPMessage".
Tots els nodes "UUPMessage" han de contenir un node que indica el pas del protocol, aquest node s'anomena "step" i un String amb el valor del pas del protocol.
A grans trets hi ha tres tipus de missatges: els de formació de grup, els de intercanvi de informació entre els nodes d'un grup i els missatges d'error. Els missatges de formació de grup estan formats per un node "joinRequest" o un node "joinResponse". Els missatges per enviar i rebre informació entre els nodes de un grup estan

formats per nodes “groupMessage” per enviar o “getMessages” per rebre. Els missatges d’error estan formats pel node “errorMessage” o “Quit”.
Per a afegir firmes digitals s’utilitza l’estàndard XML-DSIG [3]. La inclusió de firmes digitals finalment s’ha especificat com a opcional.

A Formació del grup d’usuaris

A.1 En aquest pas del protocol els nodes han de demanar ser inclosos en un grup. Per això s’ha de enviar un missatge que conté una “joinRequest” que inclogui el certificat del node.

```
<UUPMessages>
  <UUPMessage>
      <step>A.1</step>
      <joinRequest>
          <joinGroup/>
          <X509Data><X509Certificate>base64BIN</
              x509Certificat></X509Data>
      </joinRequest>
  </UUPMessage>
</UUPMessages>
```

La resposta del servidor conté una “joinResponse” indicant al node l’estat de la seva petició.

```
<UUPMessages>
  <UUPMessage>
    <step>A.1</step>
    <joinResponse>
        <state>wait-creation/retry/error</state>
    </joinResponse>
  </UUPMessage>
</UUPMessages>
```

En cas de que el grup s’hagi creat la resposta del node central inclourà l’identificador del grup, els identificadors dels usuaris i els seus certificats i els paràmetres criptogràfics p, g i longitud en bits de p (veure apartat anterior).

```
<UUPMessages>
  <UUPMessage>
    <step>A.1</step>
    <joinResponse>
        <state>created</state>
        <newGroup>
```

```
                <gid>12</gid>
                <uid>2</uid>
                <users>
                    <user id="1"><X509Data><
                        X509Certificate>base64BIN</
                        x509Certificat></X509Data></
                        user>
                    <user id="2"><X509Data><
                        X509Certificate>base64BIN</
                        x509Certificat></X509Data></
                        user>
                    ...
                    <user id="N"><X509Data><
                        X509Certificate>base64BIN</
                        x509Certificat></X509Data></
                        user>
                </users>
                <p>66892709827349182713847 49...</p>
                <g>12341398410928471273412 34...</g>
                <keylength>2048</keylength>
            </newGroup>
        </joinResponse>
    </UUPMessage>
</UUPMessages>
```

A.2 En aquest pas del protocol els nodes interroguen al node central per saber si el grup ha estat creat o encara no.

```
<UUPMessages>
  <UUPMessage>
      <step>A.2</step>
      <joinRequest>
          <joinGroup/>
      </joinRequest>
  </UUPMessage>
</UUPMessages>
```

I la resposta del node central serà la indicació de l'estat de la petició de formar grup. En cas de que el grup no s'hagi format:

```
<UUPMessages>
  <UUPMessage>
    <step>A.2</step>
    <joinResponse>
        <state>wait-creation/retry/error</state>
    </joinResponse>
  </UUPMessage>
</UUPMessages>
```

En cas de que el grup s'hagi creat la resposta del node central serà la següent:

```
<UUPMessages>
  <UUPMessage>
    <step>A.2</step>
    <joinResponse>
        <state>created</state>
        <newGroup>
            <gid>12</gid>
            <uid>2</uid>
            <users>
                <user id="1"><X509Data><
                    X509Certificate>base64BIN</
                    x509Certificat></X509Data></
                    user>
                <user id="2"><X509Data><
                    X509Certificate>base64BIN</
                    x509Certificat></X509Data></
                    user>
                ...
                <user id="N"><X509Data><
                    X509Certificate>base64BIN</
                    x509Certificat></X509Data></
                    user>
            </users>
            <p>66892709827349182713847 49...</p>
            <g>12341398410928471273412 34...</g>
            <keylength>2048</keylength>
        </newGroup>
    </joinResponse>
  </UUPMessage>
</UUPMessages>
```

A.3 En aquest pas del protocol els nodes indican al node central si accepten el grup o el rebutgen.

```
<UUPMessages>
  <UUPMessage>
    <step>A.3</step>
    <joinRequest>
        <groupAcceptance>accept/reject</
            groupAcceptance>
    </joinRequest>
  </UUPMessage>
</UUPMessages>
```

El node central respon amb l'estat de la validació (si tots els nodes han acceptat pot enviar que el grup és valid).

```
<UUPMessages>
  <UUPMessage>
    <step>A.3</step>
    <joinResponse>
        <state>wait-validation/valid/error/abort
            </state>
    </joinResponse>
  </UUPMessage>
</UUPMessages>
```

A.4 En aquest pas els nodes pregunten al node central si finalment el grup ha estat acceptat per tots els nodes i per tant és valid.

```
<UUPMessages>
  <UUPMessage>
    <step>A.4</step>
    <joinRequest>
        <groupValidationRequest/>
    </joinRequest>
  </UUPMessage>
</UUPMessages>
```

El node central respon amb l'estat de la validació.

```
<UUPMessages>
  <UUPMessage>
   <step>A.4</step>
    <joinResponse>
        <state>wait-validation/valid/error/abort
            </state>
    </joinResponse>
  </UUPMessage>
</UUPMessages>
```

B Generació de la clau de grup.

B.1 no s'envien missatges en aquest pas.

B.2 no s'envien missatges en aquest pas.

B.3 En aquest pas els nodes envien el resum criptogràfic de la seva clau pública parcial a tots els nodes del grup. Tot això va firmat amb el seu certificat digital.

```
<UUPMessages>
  <UUPMessage>
    <step>B.3</step>
    <groupMessage>
        <group>4</group>
```

```
            <sourceNode>8</sourceNode>
            <destinationNode>-1</destinationNode>
            <commitmentToShare>hash(share)</
                commitmentToShare>
            <Signature >
              <SignedInfo>
                <CanonicalizationMethod Algorithm="
                    http://www.w3.org/2006/12/xml-
                    c14n11"/>
                <SignatureMethod Algorithm="http://
                    www.w3.org/2001/04/xmldsig-more#
                    rsa-sha256"/>
                <Reference URI="#commitmentToShare">
                  <DigestMethod Algorithm="http://
                      www.w3.org/2001/04/xmlenc#
                      sha256"/>
                  <DigestValue>A7A7D08S0...</
                      DigestValue>
                </Reference>
              </SignedInfo>
              <SignatureValue>A78VA7D08...</
                  SignatureValue>
            </Signature>
        </groupMessage>
    </UUPMessage>
  </UUPMessages>
```


La resposta del node central pot contenir missatges de altres nodes que envien els seus resums criptogràfics. Per exemple:


```
  <UUPMessages>
    <UUPMessage>
        <step>B.3</step>
        <groupMessage>
            <group>4</group>
            <sourceNode>2</sourceNode>
            <destinationNode>8</destinationNode>
            <commitmentToShare>hash(share)</
                commitmentToShare>
            <Signature>...</Signature>
        </groupMessage>
    </UUPMessage>
<UUPMessage>
        <step>B.3</step>
        <groupMessage>
            <group>4</group>
            <sourceNode>3</sourceNode>
            <destinationNode>8</destinationNode>
```

```
        <commitmentToShare>hash(share)</
            commitmentToShare>
        <Signature>...</Signature>
    </groupMessage>
  </UUPMessage>
  ...
</UUPMessages>
```

O pot contenir un missatge de error que explica que no hi ha cap missatge per enviar.

```
<UUPMessages>
  <UUPMessage>
    <step>B.3</step>
    <errorMessage>no messages to send</
        errorMessage>
  </UUPMessage>
</UUPMessages>
```

B.4 En aquest pas els nodes interroguen al node central per veure si té missatges per a cadascun d'ells. Cada node envia.

```
<UUPMessages>
  <UUPMessage>
    <step>B.4</step>
    <getMessages>
  </UUPMessage>
</UUPMessages>
```

El node central enviarà els missatges que tingui dirigits al node corresponent o un missatge d'error indicant que no hi ha missatges per enviar, tal com es descriu en el pas B.3 del protocol

B.5 En aquest pas els nodes envien la seva clau pública parcial a tots els nodes del grup. Tot això va firmat amb el seu certificat digital.

```
<UUPMessages>
  <UUPMessage>
    <step>B.5</step>
    <groupMessage>
        <group>4</group>
        <sourceNode>8</sourceNode>
        <destinationNode>-1</destinationNode>
        <share>share</share>
          <Signature >
          <SignedInfo>
```

```
                <CanonicalizationMethod Algorithm="
                    http://www.w3.org/2006/12/xml-
                    c14n11"/>
                <SignatureMethod Algorithm="http://
                    www.w3.org/2001/04/xmldsig-more#
                    rsa-sha256"/>
                <Reference URI="#share">
                  <DigestMethod Algorithm="http://
                      www.w3.org/2001/04/xmlenc#
                      sha256"/>
                  <DigestValue>A7A7D08S0...</
                      DigestValue>
                </Reference>
              </SignedInfo>
              <SignatureValue>A78VA7D08...</
                  SignatureValue>
            </Signature>
        </groupMessage>
    </UUPMessage>
  </UUPMessages>
```

La resposta del node central pot contenir missatges de altres nodes que envien les seves claus públiques. Per exemple:

```
  <UUPMessages>
    <UUPMessage>
        <step>B.5</step>
        <groupMessage>
            <group>4</group>
            <sourceNode>2</sourceNode>
            <destinationNode>8</destinationNode>
            <share>share</share>
            <Signature>...</Signature>
        </groupMessage>
    </UUPMessage>
<UUPMessage>
        <step>B.5</step>
        <groupMessage>
            <group>4</group>
            <sourceNode>3</sourceNode>
            <destinationNode>8</destinationNode>
            <share>share</share>
            <Signature>...</Signature>
        </groupMessage>
    </UUPMessage>
    ...
  </UUPMessages>
```

O pot contenir un missatge de error que explica que no hi ha cap missatge per enviar.

```
<UUPMessages>
  <UUPMessage>
    <step>B.5</step>
    <errorMessage>no messages to send</
        errorMessage>
  </UUPMessage>
</UUPMessages>
```

B.6 En aquest pas els nodes interroguen al node central per veure si té missatges per a cadascun d'ells. Cada node envia:

```
<UUPMessages>
  <UUPMessage>
    <step>B.6</step>
    <getMessages>
  </UUPMessage>
</UUPMessages>
```

El node central enviarà els missatges que tingui dirigits al node corresponent o un missatge d'error indicant que no hi ha missatges per enviar, tal com es descriu en el pas B.5 del protocol

B.7 no s'envien missatges en aquest pas.

C Anonimització de les consultes de cerca.

C.1 no s'envien missatges en aquest pas.

C.2 En aquest pas cada node ha de enviar la seva consulta encriptada al node número 1.

```
<UUPMessages>
  <UUPMessage>
    <step>C.2</step>
    <groupMessage>
        <group>4</group>
        <sourceNode>8</sourceNode>
        <destinationNode>1</destinationNode>
        <query>
                <c1>AE768AE6F9D9876C897D987AE96<
                    /c1>
                <c2>AE768AE6F9D9876C897D987AE96<
                    /c2>
        </query>
    </groupMessage>
```

```
  </UUPMessage>
</UUPMessages>
```

El node central respondrà amb un missatge d'error informant que no hi ha missatges per al node i per tant que tot ha anat bé, o informant de alguna incidència en cas contrari.

El node número 1 del grup no envia la seva consulta encriptada sinó que interroga al node central per veure si té missatges per a ell esperant rebre totes les consultes dels nodes del grup.

```
<UUPMessages>
  <UUPMessage>
    <step>C.2b</step>
    <getMessages>
  </UUPMessage>
</UUPMessages>
```

El node central respondrà a la petició del node número 1 enviant els missatges que tingui d'altres nodes, que en aquest cas portaran el "step" C.2 o un missatge d'error indicant que no hi ha res per enviar.

C.3i En aquest pas cada node enviara peticions al node central per rebre el missatge amb la llista de consultes encriptades, reenmascarades i reordenades del node amb identificador immediatament inferior al seu.

```
<UUPMessages>
  <UUPMessage>
    <step>C.3i</step>
    <getMessages>
  </UUPMessage>
</UUPMessages>
```

El node central respondrà amb un missatge d'error informant de que no hi ha missatges per el node o amb un missatge tal com:

```
<UUPMessages>
  <UUPMessage>
    <step>C.3v</step>
    <groupMessage>
        <group>4</group>
        <sourceNode>1</sourceNode>
        <destinationNode>2</destinationNode>
        <queryList>
```

```
                <query>
                    <c1>AE768AE6F9D9876C897D987AE96<
                        /c1>
                    <c2>AE768AE6F9D9876C897D987AE96<
                        /c2>
                    </query>
                <query>
                    <c1>AE768AE6F9D9876C897D987AE96<
                        /c1>
                    <c2>AE768AE6F9D9876C897D987AE96<
                        /c2>
                    </query>
                ...
            </queryList>
        </groupMessage>
    </UUPMessage>
</UUPMessages>
```

C.3ii no s'envien missatges en aquest pas.

C.3iii no s'envien missatges en aquest pas.

C.3iv no s'envien missatges en aquest pas.

C.3v Un cop realitzades les transformacions adients sobre la llista de consultes, el node enviarà en aquest pas un missatge amb la llista de consultes destinada al node amb identificador immediatament superior en el grup.

```
<UUPMessages>
 <UUPMessage>
   <step>C.3v</step>
   <groupMessage>
        <group>4</group>
        <sourceNode>4</sourceNode>
        <destinationNode>5</destinationNode>
        <queryList>
            <query>
                <c1>AE768AE6F9D9876C897D987AE96<
                    /c1>
                <c2>AE768AE6F9D9876C897D987AE96<
                    /c2>
                </query>
            <query>
                <c1>AE768AE6F9D9876C897D987AE96<
                    /c1>
                <c2>AE768AE6F9D9876C897D987AE96<
                    /c2>
                </query>
            ...
```

```
        </queryList>
    </groupMessage>
  </UUPMessage>
</UUPMessages>
```

El node central respondrà amb un missatge d'error informant que no hi ha missatges per al node i per tant que tot ha anat bé, o informant de alguna incidència en cas contrari.

C.4i En aquest pas, el node amb identificador més gran en el grup espera rebre la llista de consultes encriptades de l'usuari amb identificador immediatament inferior al seu. A tal efecte envia un missatge de per demanar missatges al node central.

```
<UUPMessages>
  <UUPMessage>
    <step>C.4i</step>
    <getMessages>
  </UUPMessage>
</UUPMessages>
```

El node central respon amb el missatge destinat a aquest node:

```
<UUPMessages>
  <UUPMessage>
    <step>C.3v</step>
    <groupMessage>
        <group>4</group>
        <sourceNode>6</sourceNode>
        <destinationNode>7</destinationNode>
        <queryList>
            <query>
                <c1>AE768AE6F9D9876C897D987AE96<
                    /c1>
                <c2>AE768AE6F9D9876C897D987AE96<
                    /c2>
                </query>
            <query>
                <c1>AE768AE6F9D9876C897D987AE96<
                    /c1>
                <c2>AE768AE6F9D9876C897D987AE96<
                    /c2>
                </query>
            ...
        </queryList>
    </groupMessage>
```

```
  </UUPMessage>
</UUPMessages>
```

O amb un missatge de error indicant que no hi han missatges per a aquest node.

C.4ii no s'envien missatges en aquest pas.

C.4iii En aquest pas l'ultim node del grup reenvia les consultes de cerca desencriptades a tots els nodes del grup

```
<UUPMessages>
  <UUPMessage>
    <step>C.4 iii</step>
    <groupMessage>
        <group>4</group>
        <sourceNode>n</sourceNode>
        <destinationNode>-1</destinationNode>
        <plainTextQueryList>
            <query>how to cook an omelette</
                query>
            <query>DIY guide to arduino</query>
            <query>end of the world 2012 party</
                query>
            ...
        </plainTextQueryList>
    </groupMessage>
  </UUPMessage>
</UUPMessages>
```

El node central respon amb un missatge d'error que indica que no hi ha missatges per a aquest node.

D Enviament de les consultes al WSE i recepció del resultat.

D Els nodes que no son el U_n no tenen les consultes en clar. Per tant en aquest pas del protocol tots els nodes $U_1 ... U_{n-1}$ demanen rebre la llista de consultes en clar.

```
<UUPMessages>
  <UUPMessage>
    <step>D</step>
    <getMessages>
  </UUPMessage>
</UUPMessages>
```

El node central respondrà amb un missatge d'error indicant que no hi ha missatges per enviar. Quan tingui el missatge amb les consultes per a cada node respondrà enviant el missatge corresponent.

```
<UUPMessages>
  <UUPMessage>
    <step>C.4 iii</step>
    <groupMessage>
        <group>4</group>
        <sourceNode>n</sourceNode>
        <destinationNode>-1</destinationNode>
        <plainTextQueryList>
            <query>how to cook an omelette</
               query>
            <query>DIY guide to arduino</query>
            <query>end of the world 2012 party</
               query>
            ...
        </plainTextQueryList>
    </groupMessage>
  </UUPMessage>
</UUPMessages>
```

E Missatge de finalització

F Un cop obtingudes les consultes, i feta tota la la feina (enviament al WSE i processar el resultat) els nodes envien un missatge de finalització

```
<UUPMessages>
  <UUPMessage>
    <step>F</step>
    <Quit>
  </UUPMessage>
</UUPMessages>
```

B.2 Especificació XML Schema

L'especificació del format dels missatges del protocol UUP es du a terme en el format XML Schema. Malauradament no ha sigut possible implementar la protecció contra grups fantasma i per tant no s'ha implementat l'intercanvi de certificats ni l'ús de la firma digital. A causa d'això, s'han afegit variacions en els camps dels missatges XML del protocol previstos per a les firmes i els certificats. En l'especificació del format dels missatges es dona la opció de utilitzar els elements "Signature" i "X509Data" o una versió simplificada de tipus String. D'aquesta manera es manté la compatibilitat entre el sistema complet i la versió sense firma digital.

Per a utilitzar els elements "xmldsig" dins del fitxer "xsd" que especifica el format dels missatges del protocol UUP en la versió d'aquest treball, es descarrega l'especificació en format XML Schema de "www.w3.org/TR/2008/REC-xmldsig-core-20080610/xmldsig-core-schema.xsd" i es fa un "import" d'aquest fitxer i dels prefixos que utilitzi. D'aquesta manera qualsevol element definit en l'esquema "xmldsig-core-schema.xsd" es pot utilitzar.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:ds="http://www.w3.org/2000/09/
                     xmldsig#" >

  <xsd:import namespace="http://www.w3.org/2000/09/
      xmldsig#"
    schemaLocation="xmldsig-core-schema.xsd"/>

  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Useless User Profile Protocol XML Message schema
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="UUPMessages" type="
      UUPMessageContainerType"/>

  <xsd:element name="comment" type="xsd:string"/>

  <xsd:complexType name="UUPMessageContainerType">
        <xsd:sequence>
                  <xsd:element name="UUPMessage" type="
                     UUPMessageType" minOccurs="0"
                     maxOccurs="unbounded"/>
        </xsd:sequence>
  </xsd:complexType>

 <xsd:complexType name="UUPMessageType">
        <xsd:sequence>
                  <xsd:element name="step" type="
                     xsd:string" />
                  <xsd:element name="joinRequest" type="
                     uupJoinRequest" minOccurs="0" />
                  <xsd:element name="joinResponse" type="
                     uupJoinResponse" minOccurs="0" />
                  <xsd:element name="groupMessage" type="
                     uupGroupMessageType" minOccurs="0"/>
                  <xsd:element name="errorMessage" type="
                     xsd:string" minOccurs="0"/>
```

```
                <xsd:element name="getMessages" type="
                    xsd:string" minOccurs="0"/>
                <xsd:element name="Quit" type="
                    xsd:string" minOccurs="0"/>
        </xsd:sequence>
    <xsd:attribute name="messageDate" type="xsd:string"/
        >
  </xsd:complexType>

 <xsd:complexType name="uupJoinRequest">
    <xsd:sequence>
                <xsd:element name="joinRequest" type="
                    xsd:string"  minOccurs="0"/>
                <xsd:element name="certificate" type="
                    xsd:string" minOccurs="0" />
                <xsd:element name="X509Data" type="
                    ds:X509DataType" minOccurs="0"/>
                <xsd:element name="groupAcceptance" type
                    ="xsd:string"  minOccurs="0"/>
                <xsd:element name="
                    groupValidationRequest" type="
                    xsd:string"  minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="uupJoinResponse">
    <xsd:sequence>
                <xsd:element name="state" type="
                    xsd:string"  /><!--  values retry,
                    wait-creation,created, wait-
                    validation, valid  -->
                <xsd:element name="newGroup" type="
                    uupGroupCreationMessage" minOccurs="0
                    " />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="uupGroupCreationMessage">
        <xsd:sequence>
                <xsd:element name="gid" type="
                    xsd:integer" />
                <xsd:element name="uid" type="
                    xsd:integer" />
                <xsd:element name="users" type="
                    uupUserList" />
                <xsd:element name="p" type="xsd:string"
                    />
                <xsd:element name="g" type="xsd:string"
                    />
```

```
                <xsd:element name="keylength" type="
                    xsd:integer" />
        </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="uupUserList">
        <xsd:sequence>
                <xsd:element name="user" type="
                    uupUserItem" maxOccurs="unbounded" />
        </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="uupUserItem" >
        <xsd:sequence>
                <xsd:element name="certificate" type="
                    xsd:string" minOccurs="0"/>
                <xsd:element name="X509Data" type="
                    ds:X509DataType" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:integer"/>
</xsd:complexType>

<xsd:complexType name="uupGroupMessageType">
  <xsd:sequence>
                <xsd:element name="group" type="
                    xsd:integer"/>
                <xsd:element name="sourceNode" type="
                    xsd:integer"/>
                <xsd:element name="destinationNode" type
                    ="xsd:integer"/>
                <xsd:element name="commitmentToShare"
                    type="xsd:string" minOccurs="0" />
                <xsd:element name="commitmentok" type="
                    xsd:string" minOccurs="0" />
                <xsd:element name="share" type="
                    xsd:string" minOccurs="0" />
                <xsd:element name="shareok" type="
                    xsd:string" minOccurs="0" />
                <xsd:element name="Signature" type="
                    ds:SignatureType" minOccurs="0"/>
                <xsd:element name="signature" type="
                    xsd:string" minOccurs="0" />
                <xsd:element name="query" type="
                    encryptedMessage" minOccurs="0" />
                <xsd:element name="queryList" type="
                    queryList" minOccurs="0" />
                <xsd:element name="plainTextQueryList"
                    type="plainTextQueryList" minOccurs="
                    0" />
```

```
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="queryList">
  <xsd:sequence>
              <xsd:element name="query" type="
                 encryptedMessage" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="encryptedMessage">
  <xsd:sequence>
              <xsd:element name="c1" type="xsd:string"
                  minOccurs="0" />
              <xsd:element name="c2" type="xsd:string"
                  minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="plainTextQueryList">
  <xsd:sequence>
              <xsd:element name="query" type="
                 xsd:string" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

Apèndix C

Implementació de les operacions criptogràfiques

En aquest apèndix es detallen els càlculs i operacions relacionades amb la criptografia utilitzada en el protocol UUP. En l'aplicació servlet que fa de node central es generen els paràmetres p i g. En l'aplicació client es generen les claus pública i privada i també es realitzen les operacions de xifrat, desxifrat parcial i remasking.

C.0.1 Generació de claus

La generació dels paràmetres criptogràfics és de fonamental importància ja que la seguretat del protocol es basa en aquests.

Per a generar els paràmetres p i q s'utilitza la classe SecureRandom i BigInteger de Java. La classe BigInteger té una funció creadora que accepta la longitud en bits i un generador aleatori. A continuació es mostra el codi per a generar aquests paràmetres.

```
this.sr = new SecureRandom();

this.p = this.PrimeRandomGenerator(keyLength,this.sr);

this.g = generatorOfMultiplicativeGroup(keyLength, this.
   p, this.sr);

 private BigInteger PrimeRandomGenerator(int numbits,
    SecureRandom sr) {
    boolean correct = false;
    BigInteger p=null;
```

```
    while(!correct) {
            BigInteger two=new BigInteger("2");
            BigInteger q=new BigInteger(numbits-1,sr);
            p=q.multiply(two).add(BigInteger.ONE);   //p
               = 2q+1
            correct=p.isProbablePrime(3);   //87.5%
    }

    return p;
}
```

Per a generar el nombre primer p aleatori es genera primer un nombre q amb longitud menor que p (en concret la longitud de q es de un bit menys que p ja que p es calcula multiplicant q per dos). Seguidament es calcula el valor de $p = 2q + 1$. I finalment es comprova que el nombre p sigui primer amb una probabilitat del 87.5%. El mètode isProbablePrime(int certainty) de la classe BigInteger retorna cert si la probabilitat de que el BigInteger sigui primer supera $(1-1/2^{certainty})$ que en aquest cas val 0.875. Mentre el nombre p trobat no sigui primer es repeteixen aquestes operacions. Aquest paràmetre és el més costós de trobar, notablement quan la longitud de clau és de 2048 bits. L'estratègia seguida és que un cop trobat un valor de p, aquest es re-aprofiti en les següents execucions del protocol UUP, mentre que la resta de paràmetres com la g es generen de nou cada vegada.

```
 private BigInteger generatorOfMultiplicativeGroup(int
    numbits, BigInteger p, SecureRandom sr) {

      BigInteger two=new BigInteger("2");
      BigInteger t = new BigInteger(numbits, sr);
      BigInteger g = t.modPow(two, p);

      return g;
}
```

Per a generar el paràmetre g es genera un nombre t igual de gran que p (amb el mateix número de bits). Amb aquest nombre t es calcula $t^2 mod(p)$ i aquest resultat s'assigna al generador g.

A continuació es mostra la generació de les claus privada, pública parcial i la clau pública de grup.

```
    this.keyLength = keyLength;
    this.numUsers = numUsers;
    this.sr = new SecureRandom();
```

```
    //modulus
    this.p = p;

    //g generator
    this.g = g;

    //private key x
    this.x = newRandomParameter(keyLength - 2, sr);

    //share, partial public key yi
    this.yi = g.modPow(x, p);

 private BigInteger newRandomParameter(int numbits,
    SecureRandom sr){

                return new BigInteger(numbits,sr);
        }
```

En aquest extret de codi s'observa com la clau privada es genera amb un número aleatori de longitud menor que q (per tant la longitud en bits serà de 2 bits menys que p). La clau pública parcial y_i es calcula com $y_i = g^x mod(p)$.

Per al càlcul de la clau de grup cal multiplicat totes les claus parcials de tots els nodes:

```
public boolean generateGroupPublicKey (){
    try {
            BigInteger aux = BigInteger.ONE;
            for (int i = 0; i < this.numUsers; i++){
                    aux = aux.multiply(shares[i]);
            }

            this.y=aux;

    } catch (Exception x){
            Log.d(DEBUG_TAG, "generateGroupPublicKey()␣
               exception:␣"+x);
            return false;
    }
    return true;
}
```

Per a l'operació de remasking del protocol UUP cal utilitzar una clau publica formada per totes les claus publiques parcials dels nodes que no hagin desxifrat parcialment la llista de consultes. Com que els desxifrats es produeixen segons l'ordre assignat a dins del grup, quan un usuari U_i rep la llista de consultes, i un cop ha rea-

litzat el desxifrat amb la seva clau privada, les consultes de la llista es troben encriptades sota una clau pública formada per les claus públiques dels usuaris $U_{i+1}, ..., U_n$. Per tant per a la operació de remasking cal calcular aquesta clau pública parcial. En el moment de calcular la clau pública de grup, aprofitant que es tenen totes les claus públiques de cada node, es calcula també aquesta clau pública parcial per al remasking:

```
public boolean generatePartialPublicKey (int uid){
    try {
            BigInteger aux = BigInteger.ONE;

            for (int i = uid+1; i < this.numUsers; i++){

                    aux = aux.multiply(shares[i]);
            }

            this.ytemp=aux;

    } catch (Exception x){
            Log.d(DEBUG_TAG, "generateGroupPublicKey()␣
               exception:␣"+x);
            return false;
    }
    return true;
}
```

C.0.2 Operacions criptogràfiques

En aquest apartat es comenten totes les operacions criptogràfiques del protocol UUP.

Xifrat ElGamal

A continuació la operació de xifrat amb ElGamal:

```
private BigInteger newRandomParameter(int numbits,
    SecureRandom sr){

    return new BigInteger(numbits,sr);
}

public void  encript(String message) throws Exception {
```

```
//1
BigInteger m = new BigInteger(1, message.getBytes());
originalMessage = new BigInteger(1, message.getBytes());

//1b
if (-1 != m.compareTo(p)) {
        throw new Exception("Message length exceeds, 
           cannot be used.");
}

//2
BigInteger k = newRandomParameter(keyLength - 2, sr);

//3
BigInteger delta = g.modPow(k, p);
if (y==null) {throw new Exception("encript() Public key 
   y is null");}
if (m==null) {throw new Exception("encript() message is 
   null");}
BigInteger sigma = m.multiply(y.modPow(k, p));

this.C1 = delta;
this.C2 = sigma;

this.deltaS = this.C1.toString();
this.sigmaS = this.C2.toString();

}
```

Primerament es transforma el missatge de una String a un vector de bytes. Aquest vector de bytes s'utilitza per crear un enter de la classe BigInteger. Cal verificar que l'enter calculat és menor que p (mòdul del grup multiplicatiu Z_q^*). Després es genera un número aleatori k menor que q. Amb aquest número aleatori ja es pot realitzar l'operació de xifrat del missatge transformat en un enter, es calcula $g^k mod(p)$ i $m(y^k mod(p))$ i s'assignen a $C1$ i $C2$ respectivament. El missatge encriptat està format per $(C1, C2)$.

Desxifrat parcial ElGamal

La operació de desxifrat parcial es mostra a continuació:

```
public BigInteger partialDecryption (BigInteger c1,
   BigInteger c2,BigInteger x, BigInteger p) {

     BigInteger c2prima = c2.multiply(c1.modPow(x.negate
        (), p)).mod(p);
```

```
    return c2prima;
}
```

Es tracta de, donat el missatge encriptat format per $(c1, c2)$ calcular $c2(c1^{-x} mod(p)) mod(p)$ o el que seria el mateix $\frac{c2}{c1^x mod(p)} mod(p)$, i assignar-lo a $c2'$. Llavors el missatge parcialment desencriptat passa a ser $(c1, c2')$.
Aquesta operació es realitza per a cada element de la llista de consultes de cerca.

Remasking

La implementació del remasking es mostra a continuació:

```
public EncryptedMessage remasking(BigInteger c1original,
    BigInteger c2original, BigInteger g, BigInteger p,
   BigInteger ytemp, BigInteger x){

    BigInteger rprima = newRandomParameter(keyLength -
       2, sr);

    BigInteger one = new BigInteger(BigInteger.ONE.
       toString());

    BigInteger  c1prima = g.modPow(rprima, p);
    BigInteger c2prima = ytemp.modPow(rprima, p);

    BigInteger c1final = c1original.multiply(c1prima).
       mod(p);

    BigInteger c2final = c2original.multiply(c2prima).
       mod(p);

    EncryptedMessage result = new EncryptedMessage(
       c1final.toString(), c2final.toString());

    return result;

}
```

Primer es genera un número aleatori de longitud menor que q. Aquest número aleatori s'utilitza per calcular $E_{y'}(1, r')$, és a dir s'utilitza per xifrar un missatge de valor 1 amb la clau publica parcial calculada prèviament (clau publica que correspon al producte de les claus públiques del nodes $U_{i+1}, ..., U_n$). El resultat d'aquest xifrat

$(c1', c2') = (g^{r'} mod(p), y'^{r'} mod(p))$ **s'utilitza per calcular el producte de Hadamard entre el missatge original** $(c1, c2)$ **i** $(c1', c2')$ **tal que el resultat és** $(c1c1', c2c2')$**.**

Reordenació

Per a la reordenació de la llista de consultes s'utilitza el mètode shuffle() de la classe Collections de Java.

```
public boolean queryListReordering () {

  LinkedList<EncryptedMessage> aux = new LinkedList<
     EncryptedMessage>();
  for (int i = 0; i < this.querylist.length; i++){
          aux.add(querylist[i]);
  }
  Collections.shuffle(aux,this.sr);

  return true;
}
```

Protecció contra "targeted-public-key attack"

Per a la protecció contra l'atac "targeted-public-key" cal enviar el resum criptogràfic de la clau pública parcial de cada node. Aquest resum es calcula com segueix:

```
public String computeDigestFromString(String input){
  try {
      MessageDigest digest = java.security.MessageDigest
         .getInstance("SHA-1");
      digest.update(input.getBytes());
      byte messageDigest[] = digest.digest();

      return Base64.encode(messageDigest);

  } catch (NoSuchAlgorithmException e) {
      e.printStackTrace();
  }
  return "";
}
```

En aquest cas s'ha utilitzat l'algoritme de hash SHA-1 però seria aconsellable utilitzar un SHA-256.

En el moment de comprovar que tots els resums criptogràfics coincideixen amb les claus públiques enviades, es recorren simultàniament el vector amb les claus públiques i el vector amb els resums criptogràfics. Per cada clau es calcula el seu resum criptogràfic i es compara amb el resum criptogràfic emmagatzemat.

```
public boolean targetedPublicKeyAttackProtection(){
    boolean correct = true;

    for (int i = 0; i < this.numUsers; i++){


            String aux = this.shares[i].toString();
            String auxHash = this.
               computeDigestFromString(aux);

            if (!auxHash.equals(this.commitments[i]))
               return false;
    }

    return correct;
}
```

Apèndix D

Persistència de dades

D.1 Persistència amb el patró DAO

Per a la persistència de dades s'ha escollit utilitzar el patró de disseny DAO (Data Access Objects). Aplicat a la tecnologia java/JDBC es tracta de tenir un conjunt d'objectes que permeten realitzar les accions CRUD(Create, Read, Update i Delete). D'aquesta manera els objectes DAO fan de façana per a l'accés a les dades i ajuden a desacoblar el controlador de la persistència de dades concreta (en aquest cas l'accés a la base de dades MySQL per mitjà de JDBC).
Concretament s'ha escollit utilitzar un conjunt d'objectes JavaBean per emmagatzemar les dades i un conjunt de factories amb les operacions CRUD. Cada factoria implementa les operacions create(), read(), update() i delete() utilitzen com a paràmetres col·leccions de JavaBeans del tipus concret lligat a la factoria. Per exemple la factoria UserFactory implementa els mètodes CRUD utilitzant o retornant vectors de instàncies de la classe UserBean.
A cada factoria, els mètodes CRUD s'han creat per duplicat. Hi ha els mètodes CRUD normals que creen una nova connexió a la base de dades i els mètodes CRUD que re-utilitzen una connexió. Aquests últims permeten realitzar modificacions a la base de dades en forma de transacció. Es necessari utilitzar transaccions en les accions de creació, validació i esborrament de grups, ja que aquestes accions influeixen en les taules de usuaris (en més de una fila) i grups. Cal a més a més bloquejar l'accés a les taules quan s'estan realitzant aquestes operacions de creació, validació i esborrament de grups, ja que sinó es pot donar lloc a condicions de carrera. Per

exemple dos fils d'execució que han de incrementar el valor del número de usuaris que han acceptat el grup, primer llegeixen aquest valor i després el modifiquen (sumant-li 1). Sense el bloqueig de la taula corresponent els dos fils d'execució podrien llegir el valor abans de que s'actualitzés i per tant al modificar-lo els dos hi escriurien el mateix valor.

D.2 Creació de la base de dades

Les sentències SQL per a crear la base de dades i les seves respectives taules es mostren a continuació:

```
CREATE DATABASE uup;

USE uup;

GRANT ALL ON uup.* TO 'nodecentral'@'localhost'
   IDENTIFIED BY 'uup3847970';

CREATE TABLE groups (id INTEGER AUTO_INCREMENT PRIMARY
   KEY, numusers INTEGER DEFAULT 0, available BOOLEAN
   DEFAULT 1, numaccepted INTEGER DEFAULT 0, valid
   BOOLEAN DEFAULT 0, p TEXT, g TEXT);

CREATE TABLE users (id INTEGER AUTO_INCREMENT PRIMARY
   KEY, session VARCHAR(32) NOT NULL, groupnum INTEGER
   DEFAULT -1, usergroupnum INTEGER DEFAULT -1,
   groupAccepted INTEGER DEFAULT -1);

CREATE TABLE messages (id INTEGER AUTO_INCREMENT PRIMARY
    KEY, contents TEXT, sourceiduser INTEGER, destiduser
    INTEGER);

CREATE TABLE parameters (id INTEGER AUTO_INCREMENT
   PRIMARY KEY, p TEXT);
```

Apèndix E

Dades de test

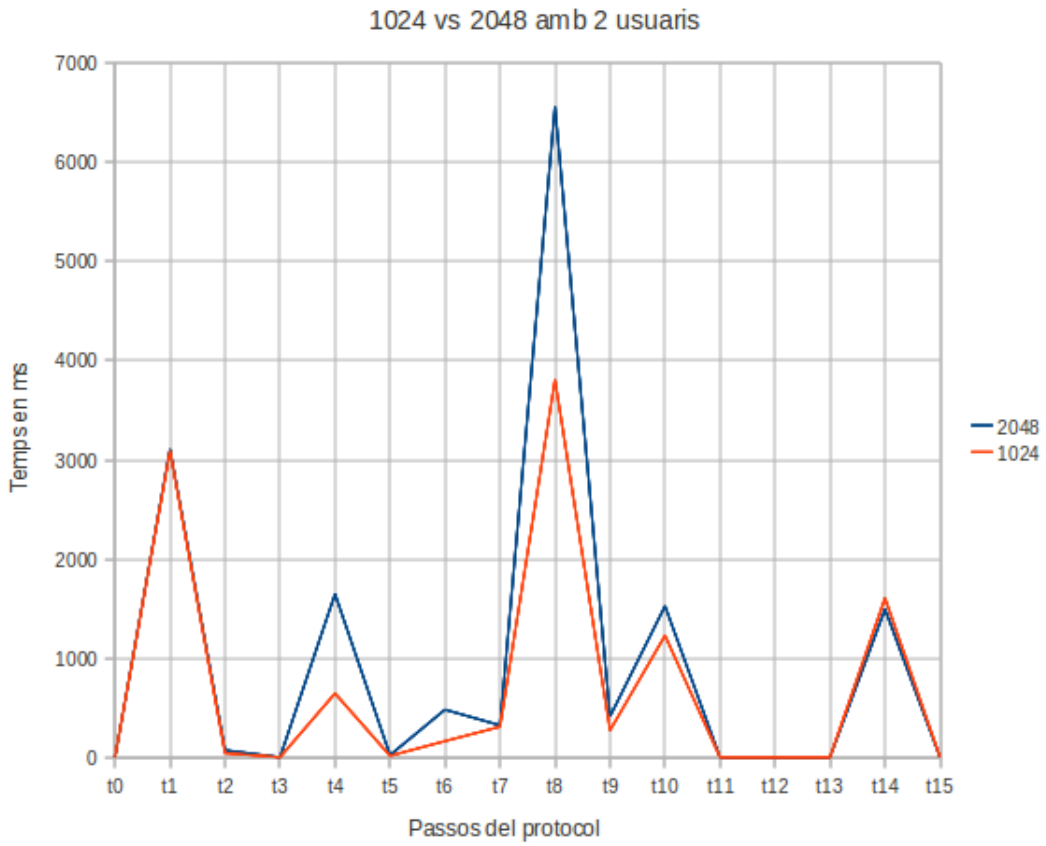

Figura E.1: Temps dels passos del protocol per execucions amb $n = 2$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari a la mateixa xarxa local que el node central i l'altre usuari des de Internet

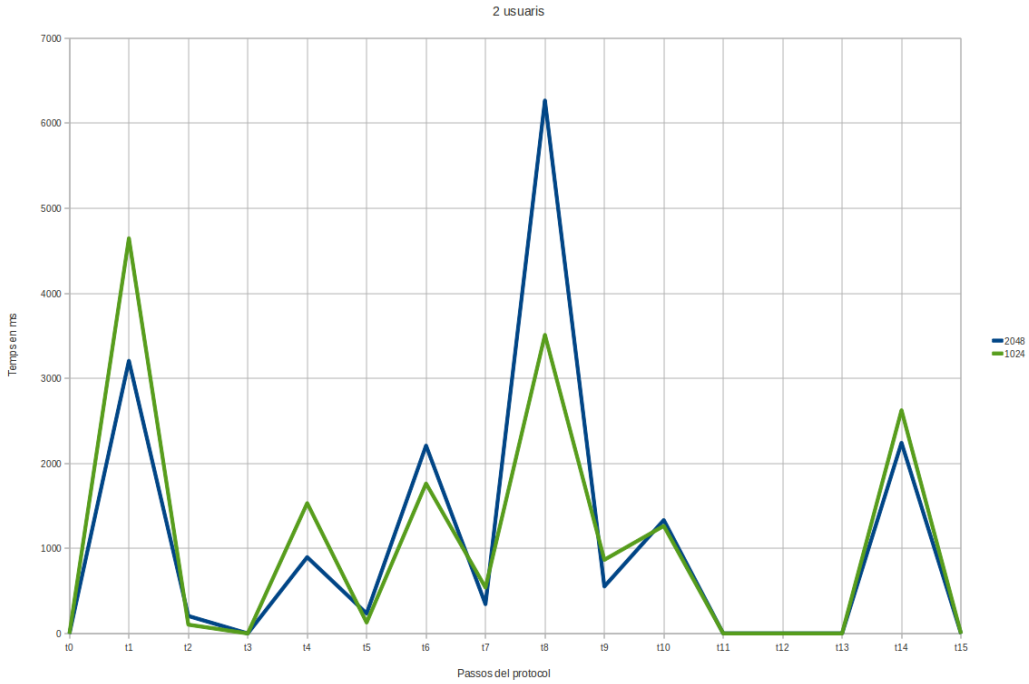

Figura E.2: Temps dels passos del protocol per execucions amb $n = 2$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari a la mateixa xarxa local que el node central i l'altre usuari des de Internet

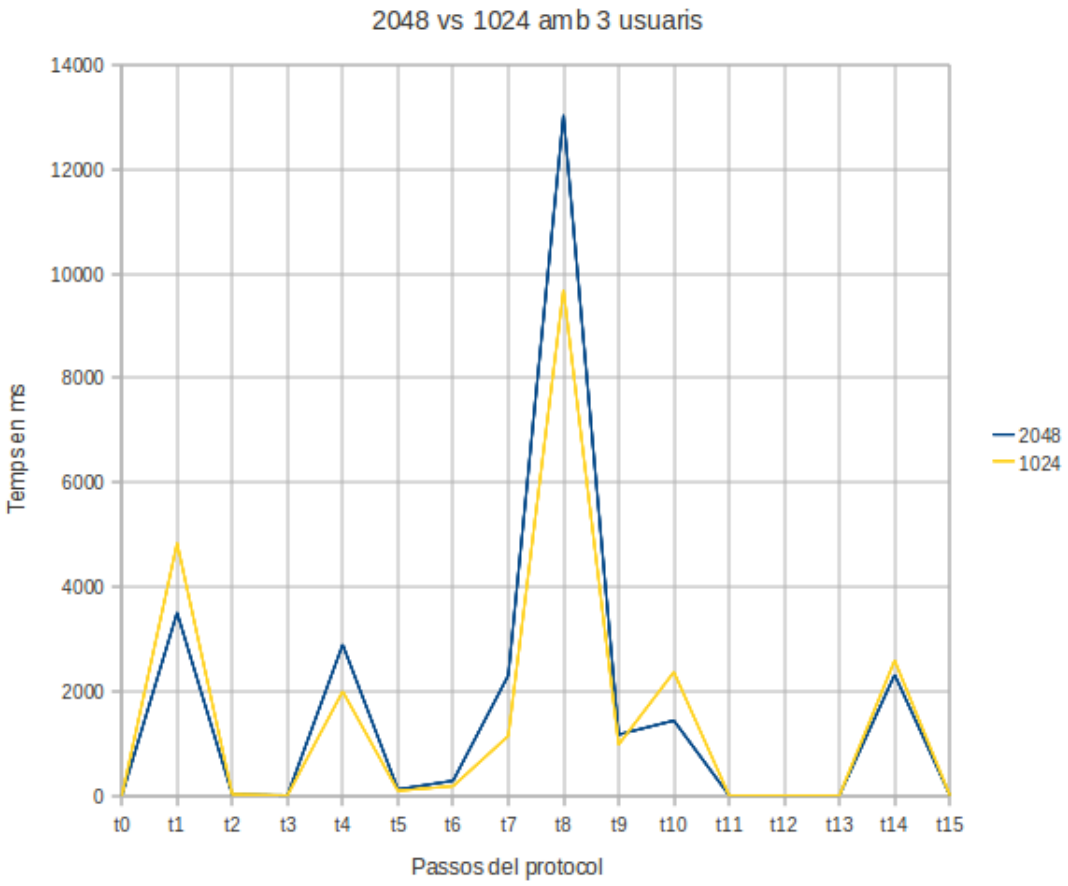

Figura E.3: Temps dels passos del protocol per execucions amb $n = 3$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Internet i la resta a la mateixa xarxa local que el node central

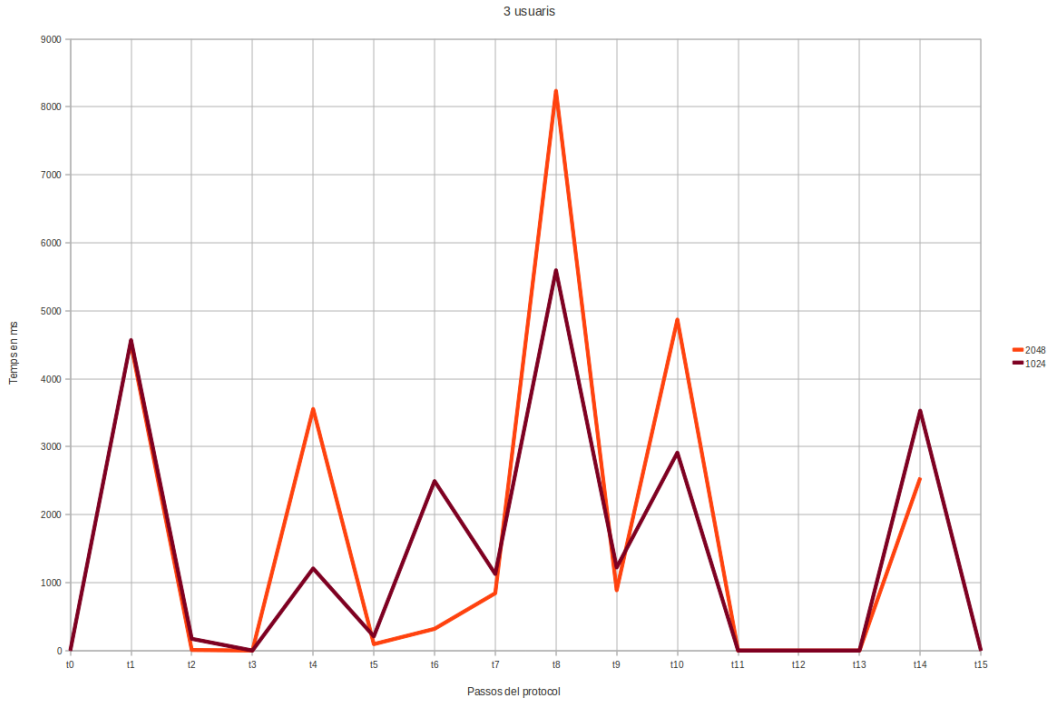

Figura E.4: Temps dels passos del protocol per execucions amb $n = 3$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Internet i la resta a la mateixa xarxa local que el node central

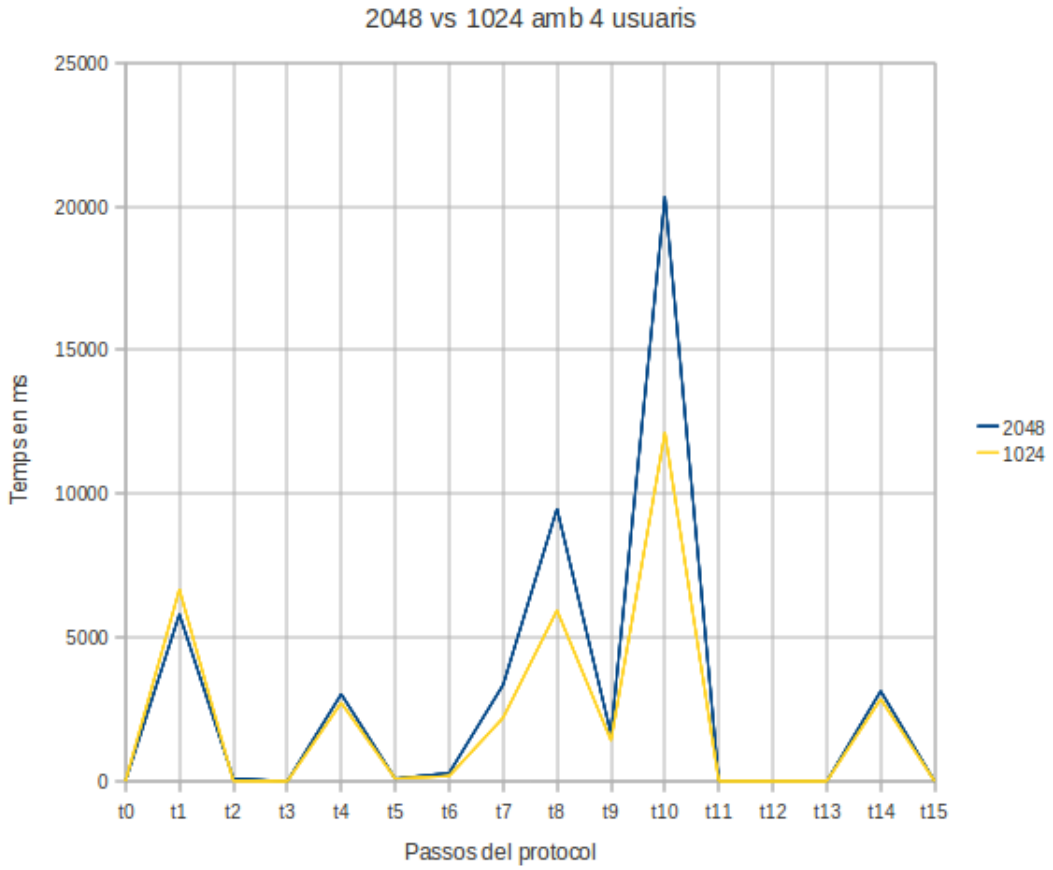

Figura E.5: Temps dels passos del protocol per execucions amb $n = 4$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Internet i la resta a la mateixa xarxa local que el node central

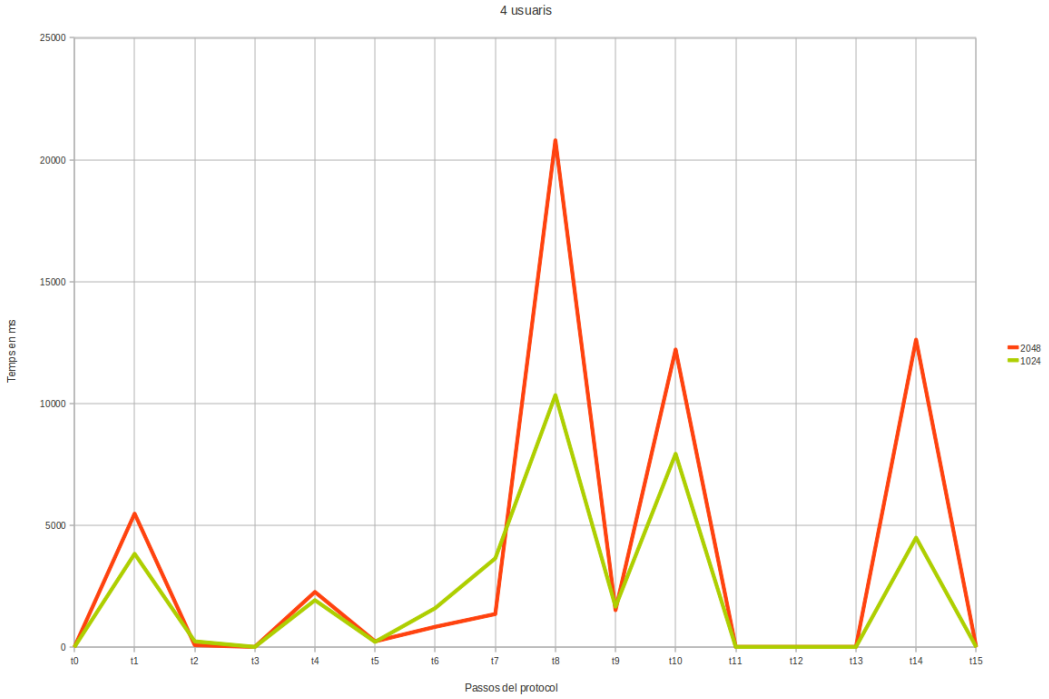

Figura E.6: Temps dels passos del protocol per execucions amb $n = 4$ usuaris amb longitud de clau 1024 i 2048 bits, un usuari des de Internet i la resta a la mateixa xarxa local que el node central

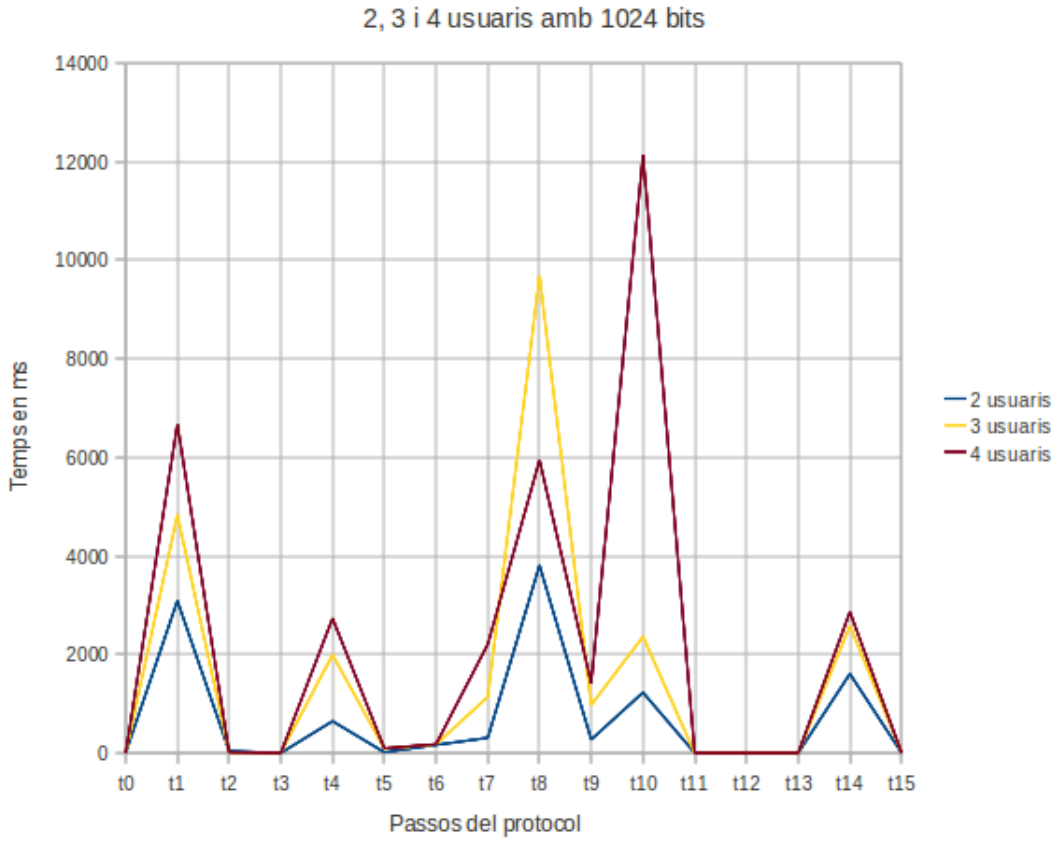

Figura E.7: Comparativa dels temps parcials del protocol amb diferent nombre d'usuaris per a una longitud de clau $l = 1024$

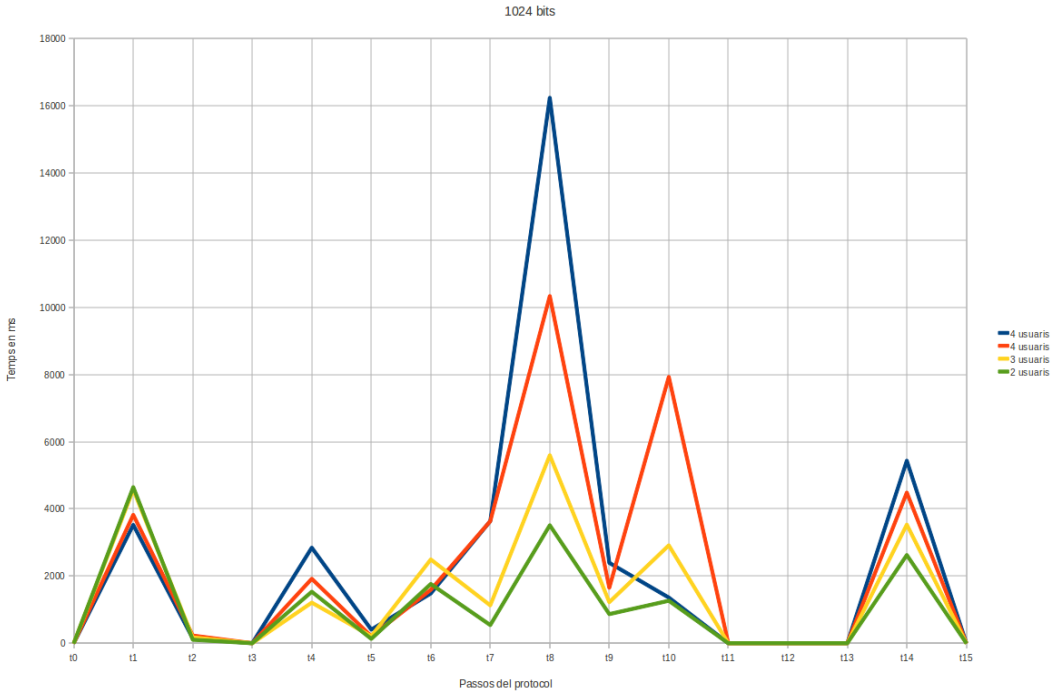

Figura E.8: Comparativa dels temps parcials del protocol amb diferent nombre d'usuaris per a una longitud de clau $l = 1024$

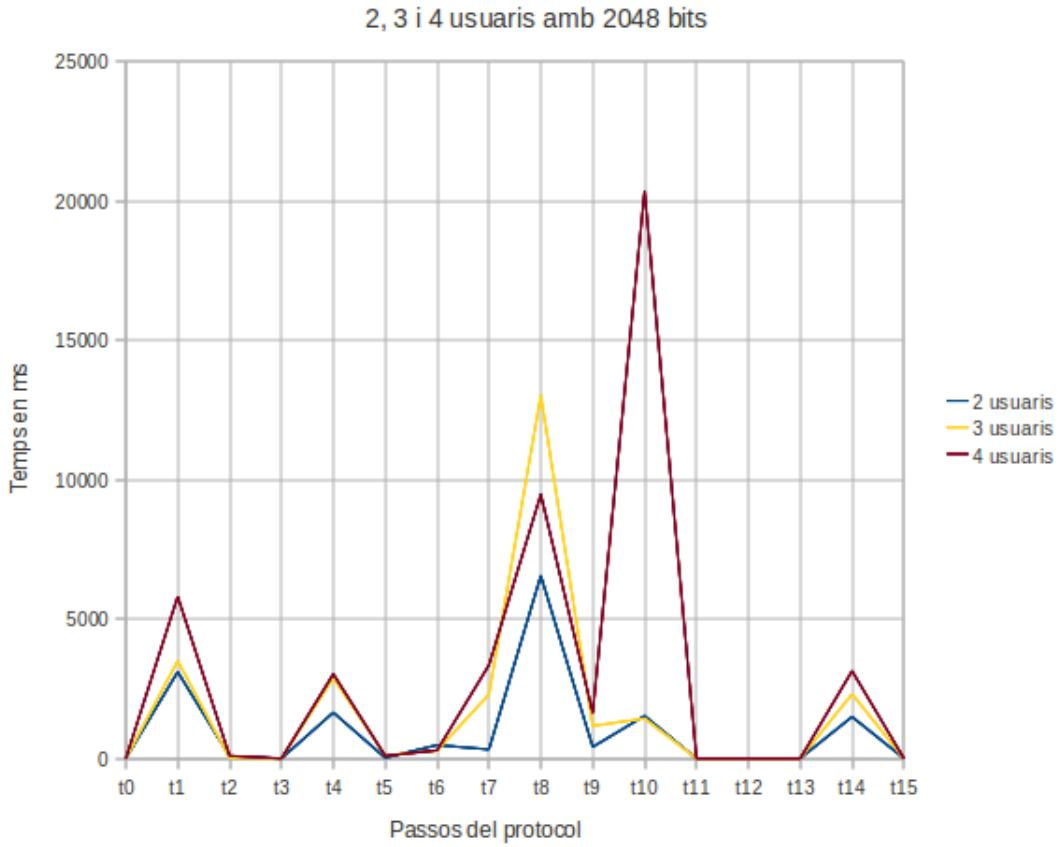

Figura E.9: Comparativa dels temps parcials del protocol amb diferent nombre d'usuaris per a una longitud de clau $l = 2048$

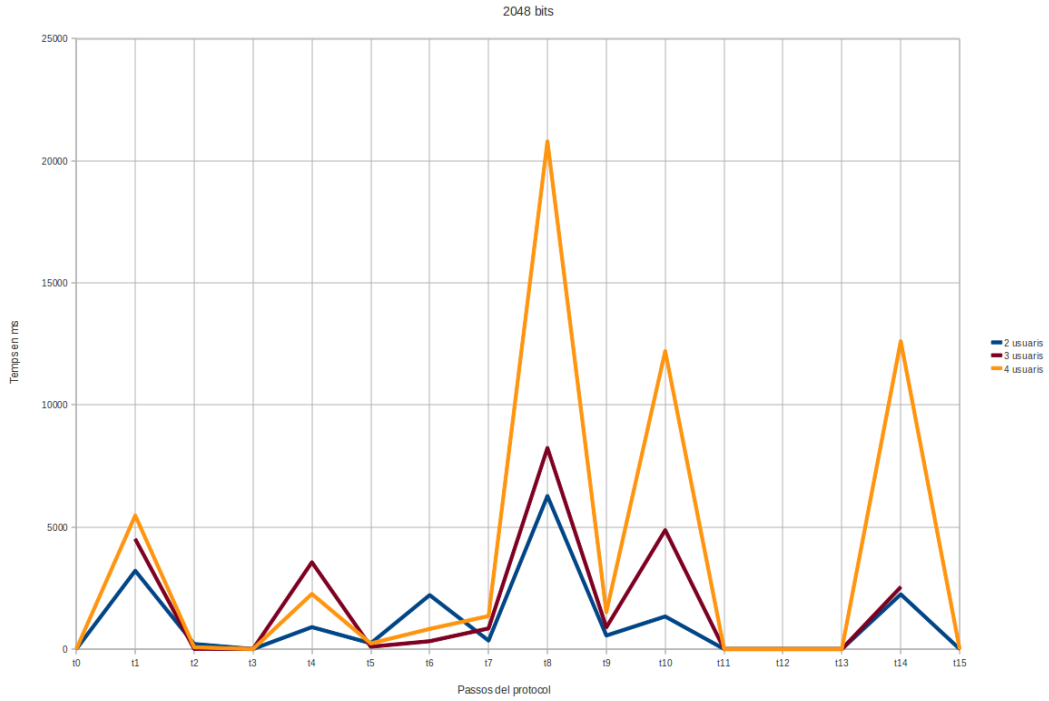

Figura E.10: Comparativa dels temps parcials del protocol amb diferent nombre d'usuaris per a una longitud de clau $l = 2048$

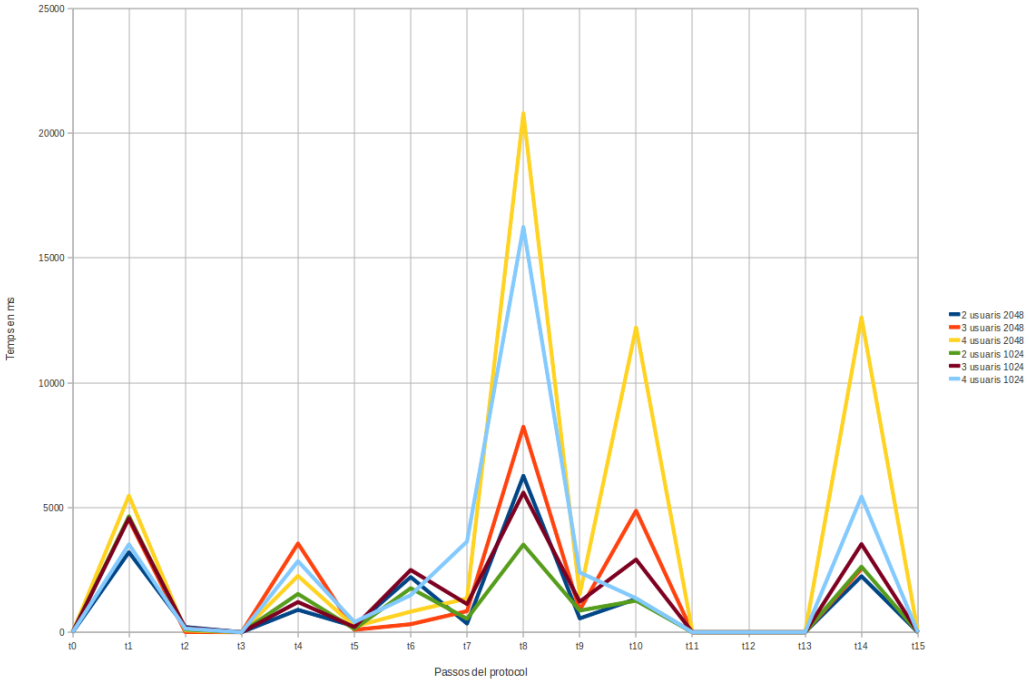

Figura E.11: Comparativa dels temps parcials del protocol amb $n = 2$, 3 i 4 usuaris per a una longitud de clau $l = 2048$ i 1024

Bibliografia

[1] Webview.loaddata() forgets to encode some crucial non-url-safe chars. http://code.google.com/p/android/issues/detail?id=1733, 2009.

[2] Xml encryption syntax and processing version 1.1, w3c working draft 18 october 2012. http://www.w3.org/TR/2012/WD-xmlenc-core1-20121018/, 2012.

[3] Xml signature syntax and processing version 1.1, w3c working draft 18 october 2012. http://www.w3.org/TR/2012/WD-xmldsig-core1-20121018/, 2012.

[4] M. Abe. Mix-networks on permutation networks. *Advances in Cryptology – Asiacrypt'92, Lecture Notes in Computer Science*, 1716:258–273, 1999.

[5] M. Barbaro and T. Zeller. A face is exposed for AOL Searcher No. 4417749. *The New York Times*, 2006.

[6] G. Brassard, C. Crépéau, and J. Robert. All-or-nothing disclosure of secrets. *Advances in Cryptology – Crypto'86, Lecture Notes in Computer Science*, 263:234–238, 1986.

[7] M. Brownlow. Smartphone statistics and market share. http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm, September 2012.

[8] J. Castellà-Roca, A. Viejo, and J. Herrera-Joancomartí. Preserving user's privacy in web search engines. *Computer Communications*, 32(13-14):1541–1551, 2009.

[9] R. Chow and P. Golle. Faking contextual data for fun, profit, and privacy. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society – WPES'09*, page 105–108, 2009.

[10] Y. Desmedt and Y. Frankel. Threshold cryptosystems. *Advances in Cryptology – CRYPTO'89. Lecture Notes in Computer Science*, 335:307–315, 1990.

[11] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *Proceedings of the 13th conference on USENIX Security Symposium*, page 21–21, 2004.

[12] J. Domingo-Ferrer, A. Solanas, and J. Castella'-Roca. h(k)-private informa- tion retrieval from privacy-uncooperative queryable databases. *Journal of Online Information Review*, 33, no.4:1468–4527, 2009.

[13] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.

[14] S. Hansell. Increasingly, Internet's Data Trail Leads to Court. *The New York Times*, 2006.

[15] S. Hansell. Yahoo Issues Takedown Notice for Spying Price List. *Wired*, December 2009.

[16] D. C. Howe and H. Nissenbaum. TrackMeNot. `http://cs.nyu.edu/trackmenot`, 2011.

[17] S. T. Peddinti and N. Saxena. On the privacy of web search based on query obfuscation: a case study of trackmenot. In *Proceedings of the 10th international conference on Privacy enhancing technologies – PETS'10*, page 19–37, 2010.

[18] M. Reiter and A. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security,*, 1, no.1:66–92, 1998.

[19] C. Romero-Tris, A. Viejo, and J. Castellà-Roca. Improving query delay in private web search. In *3PGCIC*, pages 200–206, 2011.

[20] F. Saint-Jean, A. Johnson, D. Boneh, and J. Feigenbaum. Private web search. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society – WPES'07*, page 84–90, 2007.

[21] D. L. Shinder. Pros and cons of jailbreaking or rooting your smartphone. http://www.techrepublic.com/blog/smartphones/pros-and-cons-of-jailbreaking-or-rooting-your-smartphone/1460, August 2010.

[22] N. Summers. Walking the Cyberbeat. *Newsweek*, May 2009.

[23] A. Viejo and J. Castella‘-Roca. Using social networks to distort users' profiles generated by web search engines. *Computer Networks*, 54, no.9:1343–1357, 2010.

[24] E. W. Y, Lindell. Private web search with malicious adversaries. In *Proceedings of the 10th international conference on Privacy enhancing technologies - PETS'10*, pages 220–235, 2010.