IMPLEMENTACIÓ D'UN GESTOR DE CONTRASENYES *ONLINE*

Ramon Llunell Marí
Enginyeria Tècnica en Informàtica de Sistemes

Cristina Pérez Solà
14 de juny de 2013

"There are two kinds of cryptography in this world: cryptografy that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files. This book is about the latter"

Bruce Schneider, prefaci a Applied Cryptography

M'hagués agradat poder dir el mateix d'aquesta memòria, però em temo que aquesta és sobre el primer tipus.

RESUM

Actualment està creixent de manera exponencial el nombre de serveis al qual les persones poden accedir a través d'un navegador web i la majoria d'aquests serveis demana a l'usuari que s'autentiqui introduïnt un nom d'usuari i una contrasenya.

Per a les persones és cada cop més difícil recordar el creixent nombre de contrasenyes que necessiten per accedir a tots els serveis. Això fa que augmentin pràctiques poc segures com emmagatzemar les contrasenyes en clar, utilitzar sempre la mateixa, oblidar contrasenyes,...

Els gestors de contrasenyes són unes aplicacions que faciliten als usuaris gestionar les seves contrasenyes permetent-los desar-les en un entorn segur utilitzant tècniques criptogràfiques i recuperar-les quan les necessitin. Els gestors de contrasenyes *online* són aquells als que els usuaris poden accedir a través d'Internet mitjançant un navegador.

Aquest TFC proposa la implementació d'un gestor de contrasenyes *online* que ha de garantir unes funcionalitats mínimes de desar i recuperar contrasenyes alhora que ha de garantir la confidencialitat de les dades emmagatzemades al servidor.

Per desenvolupar aquest projecte es posaran en pràctica els coneixements adquirits en les diferents assignatures de l'Enginyeria Tècnica en Informàtica de Sistemes. En especial, es posaran en pràctica aspectes relacionats amb la seguretat de la informació i la criptografia, i aspectes relacionats amb la planificació i desenvolupament de projectes en el marc de l'enginyeria del programari.

ÍNDEX DE CONTINGUTS

ÍNDEX DE FIGURES

ÍNDEX DE TAULES

CAPÍTOL 1 - INTRODUCCIÓ

1.1. Justificació del TFC

En els darrers anys, l'increment exponencial de l'ús de les noves tecnologies ha generat un canvi molt significatiu en la relació dels usuaris amb la informació i amb Internet. Aquest canvi no és només tecnològic sinó també social i es basa en els següents aspectes fonamentals:

a) El gruix de la informació digital que els usuaris generen ha passat d'estar emmagatzemada als ordinadors dels usuaris a estar-ho a la xarxa

b) Hi ha una tendència cada cop més generalitzada a utilitzar programaris que no estan instal·lats als ordinadors dels usuaris i que són accessibles a través d'Internet (*cloud computing*)

c) La informació no només s'emmagatzema sinó que la compartició d'aquesta informació ha esdevingut una funcionalitat prioritària de la nova xarxa (xarxes socials, compartició d'arxius,...)

d) La mobilitat i la necessitat d'accedir a la informació des de diferents dispositius: PC, portàtil, tauletes, *smartphones*,...

Aquest escenari fa que els usuaris cada cop accedeixin a més aplicacions i serveis a Internet. Lògicament, tots aquests serveis requereixen l'autenticació dels usuaris. Malgrat que tecnològicament ja serien possibles mètodes d'autenticació basats en certificats digitals, en DNI electrònic o en sistemes de *single sign-on*, la poca implementació entre els usuaris fa que l'autenticació dels clients se segueixi fent, de manera molt majoritària, mitjançant l'ús de contrasenyes.

En una situació ideal, els usuaris utilitzarien contrasenyes fortes que serien diferents per cada servei i les recordarien. Però el fet que, en la pràctica, això és pràcticament impossible fa que es produeixin situacions que poden comprometre la seguretat:

- ➔ Utilitzar contrasenyes molt evidents per poder recordar-les i susceptibles de ser trencades per atacs de diccionari
- ➔ Utilitzar la mateixa contrasenya per a tots els serveis
- ➔ Guardar en clar les parelles usuari/contrasenya en un simple document de text sense protecció davant la impossibilitat de recordar-les totes

Aquestes situacions -que es donen més del que seria desitjable en el món real- han portat al desenvolupament d'unes aplicacions que tenen per objecte emmagatzemar les contrasenyes d'una

manera segura. D'aquesta manera, els usuaris poden utilitzar contrasenyes fortes, diferents per a cada aplicació o servei i no els cal guardar-les en clar per recordar-les.

Algunes aplicacions incorporen aquests gestors dins el seu programari. Típicament ho fan els navegadors ja que és mitjançant aquests que els usuaris accedeixen a la majoria de serveis a Internet. Aquests gestors de contrasenyes, però, són desats en local en l'ordinador de l'usuari i això comporta alguns problemes de seguretat. Un atacant que prengués el control de l'equip podria accedir a tots els serveis web de l'usuari[1].

Una possible solució seria que també el gestor de contrasenyes estigui disponible a través d'Internet. Això resoldria els inconvenients esmentats, però també plantejaria nous reptes de seguretat. Si no estigués ben dissenyat i es comprometés la seguretat del servidor del gestor podria comprometre's també l'accés de tots els usuaris als seus serveis.

Aquest TFC desenvolupa la implementació d'un gestor de contrasenyes *online* que intenta maximitzar els requisits de seguretat, però mantenint un equilibri amb la usabilitat per part dels usuaris. Considero que un sistema que fos tecnològicament segur -si això fora possible- però que no tingués en compte en aquest punt no compliria la seva missió. És a dir, s'ha dissenyat i implementat -amb les seves limitacions- un gestor de contrasenyes pel món real.

1.2. Objectius del TFC

L'objectiu del TFC serà implementar un gestor de contrasenyes *online* en un servidor que estigui disponible a través d'Internet i que serà accessible com a servei web utilitzant qualsevol navegador. Aquest gestor haurà de complir els requisits mínims que es demanen en l'enunciat del TFC:

Requisits funcionals mínims

a) donar d'alta clients en el servei de gestió de contrasenyes

b) que els clients puguin emmagatzemar parelles d'usuari/contrasenya

Requisits de seguretat mínims

a) assegurar que les contrasenyes no viatgen en clar per la xarxa

b) assegurar que en cap moment els administradors del gestor poden conèixer cap de les contrasenyes dels seus usuaris

Com veurem en capítols posteriors, en la fase d'anàlisi del projecte es van incorporar noves funcionalitats i requisits de seguretat al projecte. D'altra banda, per manca de temps es va descartar altres funcionalitats opcionals com la compartició de contrasenyes o la generació de contrasenyes fortes.

1 Per simplificar la gestió als usuaris, alguns navegadors xifren les contrasenyes desades basant-se en la contrasenya d'accés a l'equip (Internet Explorer [1][2], Chrome [3][4]). Altres navegadors permeten establir una contrasenya mestra diferent (Firefox [5]), fet que aporta més seguretat.

1.3. Metodologia

Per desenvolupar el projecte d'implementació del gestor de contrasenyes *online* s'ha seguit un cicle de vida clàssic del programari o cicle de vida en cascada. Aquesta memòria segueix aquest esquema i les seves etapes en els diferents capítols.

- **Anàlisi prèvia**: aquesta etapa ve determinada pel mateix enunciat del TFC que inclou la descripció, a grans trets, del projecte i del programari a implementar
- **Anàlisi de requisits**: en aquesta etapa es van desenvolupar d'una manera detallada els requisits de l'aplicació i, fruit d'aquest desenvolupament, es van triar els algorismes criptogràfics que serien utilitzats.
- **Disseny**: es va definir l'arquitectura de l'aplicació, les estructures de dades, les especificacions de les diferents funcions i les interfícies d'usuari.
- **Programació**: es va implementar el disseny desenvolupat utilitzant diferents tecnologies.
- **Etapa de prova**: es van especificar els jocs de proves que intentaven cobrir totes les possibles situacions a què es podia enfrontar el gestor de contrasenyes. Aquestes proves, van fer que es modifiquessin alguns punts del programari en què es van detectar funcionaments incorrectes.
- **Manteniment o explotació**: Per la naturalesa d'aquest projecte, el gestor de contrasenyes implementat no ha passat a la fase de producció. Per tant, no hi ha hagut fase de manteniment. Malgrat això, en el darrer capítol s'han apuntat algunes possibles situacions a què es podria enfrontar el programari en el futur.

1.4. Productes obtinguts

El producte final resultant d'aquest projecte és una aplicació web que permet als usuaris registrar-se per tal de gestionar les seves contrasenyes. Un cop registrats, els usuaris poden desar, recuperar, modificar i eliminar les dades d'accés als diferents serveis.

Aquesta aplicació es lliura estructurada en diferents tipus d'arxius:

- **HTML**: defineixen la interfície web amb la que interactuaran els usuaris
- **PHP**: inclouen les funcions de l'aplicació de la banda del servidor
- **JavaScript**: inclouen les funcions de l'aplicació de la banda del client
- **Dades**: taules i registres de la base de dades

1.5. Resum dels altres capítols

En el segon capítol es defineixen els gestors de contrasenyes i es classifiquen en diferents tipus, amb especial atenció als avantatges i desavantatges de cadascun d'ells.

El tercer capítol està dedicat a la fase d'anàlisi. En primer lloc, s'amplien i detallen els requisits que ha d'implementar l'aplicació. En segon lloc, s'entra en molt més detall en cadascun d'aquests requisits fent un especial esment a les implicacions que els requisits de seguretat tenen sobre les funcionalitats que ha d'oferir el gestor. Per estructurar millor la informació, el gestor de contrasenyes *online* es descomposa en dos mòduls: un modul de gestió d'usuaris i un mòdul de gestió de contrasenyes.

El capítol número quatre d'aquesta memòria està dedicat a la fase de disseny. Es dissenyen la arquitectura de l'aplicació, les estructures de dades i l'estructura del programari per tal que respongui als requisits de l'aplicació. Donada l'estructura dual de l'aplicació amb una part del programari executant-se en el client i l'altra en el servidor, es fa un especial èmfasi en aquesta relació i en els diagrames de flux que la modelen.

El cinquè capítol està dedicat a la fase de programació. En primer lloc, es presenten les tecnologies triades per a la implementació i es justifica la seva elecció. En segon, lloc es repassen alguns dels aspectes més importants de la programació dels dos mòduls, el d'usuaris i el de contrasenyes.

El sisè capítol presenta el producte final, l'aplicació en funcionament. Es repassen les diferents funcionalitats des del punt de vista de l'usuari.

Finalment, el darrer capítol d'aquesta memòria presenta algunes conclusions del TFC. En primer lloc, es fa una valoració del resultat obtingut parant especial atenció als punts de millora que haurien de contemplar-se en el futur. En aquest capítol també s'introdueixen alguns elements de discussió sobre el futur dels gestors de contrasenyes i sobre la seguretat i la confiança des del punt de vista dels usuaris.

CAPÍTOL 2 - ELS GESTORS DE CONTRASENYES

2.1. Què és un gestor de contrasenyes?

Veiem algunes descripcions que s'han donat dels gestors de contrasenyes:

- "*Un gestor de contrasenyes és un programa que s'utilitza per emmagatzemar una gran quantitat de parelles usuari/contrasenya. La base de dades on es guarda aquesta informació està xifrada mitjançant una única clau (contrasenya mestre o master password), de manera que l'usuari només hagi de memoritzar una clau per accedir a totes les demés.*" - Wikipèdia en castellà [6].
- *"Un gestor de contrasenyes és un programari que ajuda a l'usuari a organitzar contrasenyes i codis PIN. Típicament, el programari té una base de dades local o un arxiu que conté les contrasenyes encriptades per l'autenticació segura a ordinadors, xarxes, pàgines web i arxius amb dades d'aplicacions. Alguns gestors de contrasenyes també fan la funció d'omplir els formularis d'accés i omplen el nom d'usuari i la contrasenya de manera automàtica en els formularis."* - Wikipèdia en anglès [7].
- "*Un gestor de contrasenyes és un programa que s'utilitza per emmagatzemar contrasenyes. Ens permet recordar totes les contrasenyes, claus d'accés i noms d'usuari que necessitem per accedir a un compte o pàgina d'Internet. La informació s'emmagatzema xifrada i només es pot accedir-hi a través d'una clau.*" - INTECO [8]
- "*Un gestor de contrasenyes és un programari, o a vegades un servei, que ajuda a recordar contrasenyes i altra informació de login*" - PC Support [9].
- "*Utilitat local que permet a un usuari emmagatzemar noms d'usuari, contrasenyes i altres petits fragments d'informació sensible, com nombre de compte*" - NIST [10].
- "*Un gestor de contrasenyes és un programari que requereix a l'usuari recordar una sola contrasenya forta, que és utilitzada per desencriptar la base de dades del gestor de contrasenyes. Recordar una sola contrasenya mestra és molt més fàcil pels usuaris que poden, però, beneficiar-se de l'utilització de diferents contrasenyes per a cada servei online*" - Paolo Gasti i Kasper B. Rasmussen [11].
- "*El gestor de contrasenyes és una eina indispensable pels usuaris d'Internet. Automatitza completament el procés d'entrar les contrasenyes i altres dades als llocs web i estalvia a l'usuari el problema de crear i recordar múltiples contrasenyes. El programari crea contrasenyes extraordinàriament fortes i protegeix la informació de login de robatoris. Tota la*

> *informació confidencial és encriptada en una base de dades dedicada en l'ordinador de l'usuari*" - Kaspersky [12].

Si analitzem aquestes definicions dels gestors de contrasenyes en destaquen dos aspectes: a) són un programari que té per objecte emmagatzemar les contrasenyes d'accés a serveis web; b) els gestors de contrasenyes emmagatzemen les contrasenyes xifrades. Destacar que el NIST i PC Support defineixen els gestors només en base a la seva utilitat, però no fan cap esment de la seguretat implícita en aquest tipus d'utilitats.

Tot i la seva simplicitat, he considerat oportú afegir la definició de PC Support ja que és la única que menciona específicament els gestors de contrasenyes com un servei. Crec que és un detall important ja que -encara que no és el cas d'aquesta memòria- els gestors de contrasenyes es poden implementar com un servei dins una arquitectura orientada a serveis (SOA).

També cal esmentar que algunes definicions d'entitats que podem considerar autoritats en la matèria de la seguretat (NIST i Kaspersky) no consideren la possibilitat de gestors de contrasenyes *online* ja que els defineixen com a programari local instal·lat a l'ordinador de cada usuari.

La següent definició dels gestors de contrasenyes intenta incloure tots els punts que, al meu entendre, defineixen aquests:

> Els **gestors de contrasenyes** són un programari, aplicació o servei que permeten als usuaris emmagatzemar i utilitzar les seves contrasenyes de manera segura, de manera que a l'usuari només li calgui recordar una sola contrasenya amb la que s'encriptaran totes les altres. De manera opcional, els gestors de contrasenyes també poden implememementar altres funcionalitats que facilitin als usuaris la gestió de les seves dades d'accés a diferents serveis: generació de contrasenyes segures, accés directe als formularis d'autenticació dels serveis,...

2.2. Tipus de gestors de contrasenyes

Com hem vist, un gestor de contrasenyes és, bàsicament, un programari que emmagatzema de manera segura les contrasenyes dels usuaris. En funció d'on s'emmagatzemin les contrasenyes tenim dos grans tipus de gestors de contrasenyes: a) aquells en què les contrasenyes s'emmagatzemen en la banda de l'usuari; b) aquells en què les contrasenyes s'emmagatzemen en un servidor accessible des d'una xarxa de comunicacions (típicament, Internet).

2.2.1. Gestors de contrasenyes locals

En els gestors de contrasenyes locals, els arxius que desen les contrasenyes encriptades estan en l'equip o equips de l'usuari. En la majoria de casos, es tracta de funcionalitats que implementen els navegadors per facilitar als usuaris l'accés a serveis web que requereixen autenticació.

Encara que els gestors de contrasenyes locals ofereixen una gestió fàcil i intuïtiva de les contrasenyes pels usuaris, tenen alguns inconvenients relacionats amb la seguretat:

- la informació podria perdre's en cas d'avaria o pèrdua de l'ordinador de l'usuari

- ➔ en cas que la seguretat del sistema local fos compromesa, l'atacant podria suplantar l'identitat de l'usuari legítim
- ➔ aquests sistema obliga a mantenir un gestor de contrasenyes en cada dispositiu des del qual s'accedeix als serveis i, a més, no permet accedir-hi des de dispositius que no són els propis

Una variació d'aquest tipus, són aquells gestors que permeten la seva instal·lació en un dispositiu mòbil (el telèfon mòbil, un llapis USB,...) de manera que l'usuari els pugui utilitzar en qualsevol dispositiu que s'hi pugui connectar. Aquests tipus de gestors permeten utilitzar el gestor en qualsevol dispositiu de l'usuari i aporten més seguretat ja que, encara que aquest dispositiu es perdés, la seva utilització acostuma a estar lligada a l'autenticació prèvia amb PINs o contrasenyes. Però segueixen tenint l'inconvenient que en cas de pèrdua o avaria, l'usuari perdria l'accés als serveis.

2.2.2. Gestors de contrasenyes online

En el cas dels gestors de contrasenyes online, les contrasenyes encriptades es desen en un servidor al qual els usuaris es poden connectar mitjançant Internet. Entre els gestors online podem diferenciar dos tipus:

- ➔ Els que s'ofereixen com un servei web de manera que, no només l'emmagatzemament, sinó també tota la gestió de les contrasenyes -i altres serveis addicionals, si n'hi ha- es fa via web.
- ➔ Els que desen les contrasenyes encriptades a la xarxa, però tot el procés de gestió de contrasenyes es realitza a través d'una aplicació local.

Els gestors de contrasenyes *online* adrecen alguns dels problemes funcionals i de seguretat que afecten als gestors locals. L'usuari pot accedir des de qualsevol equip -propi o no- als seus serveis amb l'única condició que estigui connectat a Internet. La pèrdua de la màquina local no implica la pèrdua de les dades d'accés als serveis web. I un atacant que prengués el control de la màquina local no guanyaria accés als nostres serveis.

Però els gestors de contrasenyes *online* també impliquen altres riscs:

a) Encara que menor, també hi ha el risc de fallada en el servidor que desa les contrasenyes. El proveïdor pot minitmitzar aquest risc dissenyant un sistema de servidors de bases de dades redundants i una política conservadora de còpies de seguretat.

b) Cal confiar en el proveïdor. Un gestor de contrasenyes *online* podria ser un mètode per obtenir les dades d'accés dels usuaris. Alguns proveïdors han optat per fer públic el seu codi font perquè pugui ser revisat per terceres persones. Cal dir, en qualsevol cas, que el problema de la confiança no és exclusiu dels gestors de contrasenyes *online*.

c) Depenen molt de la fortalesa de la contrasenya que s'utilitza per encriptar la resta de contrasenyes. És a dir, de la fortalesa de la contrasenya que ha de recordar l'usuari [13][14]. Com evolucionarà el concepte de fortalesa de la contrasenya en el futur? El que ara es considera fort, pot ser una contasenya dèbil d'aquí a uns anys?

d) Estem parlant d'accés a serveis importants per als usuaris (correu electrònic, xarxes socials, comerç electrònic,...). Què passaria si, per exemple, l'empresa que administra el gestor de contrasenyes fa fallida? Crec que un bon gestor de contrasenyes *online* hauria de preveure el

seu desmantellament ordenat i sense pèrdua d'informació pels usuaris. No he trobat informació sobre aquest punt i em sembla una qüestió crítica a la que crec que s'ha donat poca importància.

2.3. Els gestors de contrasenyes online

Davant del creixement exponencial dels serveis que s'ofereixen via web i la necessitat per part dels usuaris de recordar cada cop més dades d'autenticació davant aquests serveis, en els darrers temps han augmentat els proveïdors que ofereixen el servei de gestor de contrasenyes *online*.

A continuació, llistem alguns dels més destacats i afegim entre parèntesi la data de la seva darrera actualització[2]:

- LastPass **** (18 de juny de 2012)
- KeePass Password Safe (4 d'octubre de 2012)
- clipperz KEEP IT TO YOURSELF! (4 de setembre de 2011)
- Password Safe (9 de novembre de 2012)

He inclòs la data de la darrera actualització perquè em sembla una dada molt reveladora. Com es discutirà en el darrer capítol, quan un usuari confia en un gestor de contrasenyes aquest pot esdevenir crític perquè pugui accedir als serveis web dels quals desa la contrasenya. Però en algunes ocasions els gestors de contrasenyes poden respondre a projectes empresarials poc definits o que poden suscitar dubtes importants respecte al seu futur. Aquest cas és encara més patent en el cas dels gestors gratuïts. Crec que això planteja una de les incògnites més importants sobre els gestors de contrasenyes *online* i sobre tots aquells serveis que suposen cedir informació crítica: està garantit el manteniment d'aquests serveis en tot cas? Han previst els proveïdors d'aquests serveis, en cas necessari, el possible tancament del servei i com minimitzar els efectes en els usuaris? Per exemple, imaginem que una empresa que gestiona un gestor de contrasenyes, per manca de recursos econòmics o per la inviabilitat del projecte, deixa de mantenir els seus servidors o el seu proveïdor de *hosting* els desconnecta. Si la informació és anònima, com es pot contactar amb els usuaris per donar-los un termini per recuperar les seves contrasenyes?

2.4. Principals atacs sobre l'autenticació per contrasenyes

2.4.1. Atacs de força bruta

Els atacs de força bruta -referit a un atac contra contrasenyes- consisteixen en intentar l'autenticació en el servei atacat aplicant un mètode d'assaig i error [16] fins a trobar una combinació

2 Segons l'article de Susan Brudno *Passwords, A Tangled Web* [15]

que permeti l'accés. És el tipus d'atac més costós i requereix una gran quantitat de recursos. Sovint s'utilitzen sistemes que realitzen els intents de manera automatitzada.

En sentit estricte, un atac de força bruta examina exhaustivament totes les combinacions de caràcters possible que poden formar la contrasenya per accedir al sistema.

Per impedir aquest tipus d'atacs, els serveis web poden prendre una sèrie de mesures fàcils d'implementar com utilitzar contrasenyes llargues i permetre un nombre màxim d'intents fallits abans de bloquejar el sistema durant un temps donat. Una altre mesura que dificulta els atacs de força bruta és utilitzar un sistema d'encriptació de contrasenyes que tingui un elevat cost computacional. En aquest cas, cada intent del atacant consumirà més temps fent més costós l'atac en termes de recursos temporals a dedicar. Alternativament, també es pot establir un petit retard en la resposta del servidor a l'intent d'autenticació per augmentar el temps necessari per l'atac [17].

2.4.2. Atacs de diccionari

Els atacs de diccionari exploten el fet que la majoria dels usuaris escullen les seves contrasenyes basant-se en la facilitat per recordar-la. Per això, utilitzen variacions de paraules amb sentit enlloc de combinacions aleatòries de caràcters. És a dir, la majoria de contrasenyes estan dins un domini molt més reduït que el que formen la totalitat de combinacions possibles [17].

D'aquesta manera, un atacant que utilitzi l'atac de diccionari actuarà igual que en un atac de força bruta, però optimitzarà els seus recusos limitant les combinacions que assaja a aquelles que formen part del diccionari triat. Com a exemple senzill, pensem en un atacant que limités a fer intents de connexió davant un servei provant combinacions de noms propis amb xifres. La probabilitat d'obtenir algun accés il·legítim és de varis ordre de magnitud superior a la probabilitat si prova totes les combinacions.

Per prevenir els atacs de diccionari es poden utilitzar bits de sal afegits en el xifratge de la contrasenya i utilitzar contrasenyes formades per diverses paraules (frases de pas) en lloc d'una sola paraula[18].

2.4.3. *Rainbow tables*[3]

Les taules rainbow són taules que inclouen les funcions hash per un conjunt de contrasenyes [19]. Les funcions hash són un mètode molt habitual d'emmagatzemar les contrasenyes dels usuaris ja que són funcions unidireccionals i, per tant, la contrasenya no pot derivar-se d'aquesta funció. Si un atacant guanya accés a aquests valors desats a la base de dades, les taules rainbow li poden permetre esbrinar la contrasenya en clar.

Aquest tipus d'atac explota, igual que en el cas dels atacs de diccionari, el fet que la majoria d'usuaris tria les seves contrasenyes utilitzant paraules amb sentit. D'aquesta manera, per ser efectiva, a una taula rainbow no li cal ser exhaustiva sinó que se centrarà en aquelles combinacions de contrasenyes que siguin més probables.

Per prevenir aquest tipus d'atac és important que el mètode amb què es xifrin les contrasenyes abans de desar-les utilitzi bits de sal. Aquests bits, al ser aleatoris, fa gairebé impossible l'atac ja que,

3 S'utilitza el terme en anglès ja que no he trobat el terme en català.

per una mateixa contrasenya, hi ha una gran quantitat de funcions hash possibles. Tantes com combinacions possibles dels bits de sal.

2.4.4. Atacs tipus *man-in-the-middle*

A diferència dels tipus atacs anteriors, els atacs del tipus *man-in-the-middle* [20] són atacs que pretenen interceptar la comunicació entre dos usuaris legítims. En el cas dels atacs contra contrasenyes, s'intercepten les comunicacions entre el client i el servidor en el moment de l'autenticació. Si un atacant intercepta la contrasenya amb la que l'usuari s'autentica, podrà posteriorment suplantar-lo accedint a la seva informació. Cal tenir en compte que el fet d'enviar la contrasenya xifrada no és una defensa suficient contra aquest atac si aquesta contrasenya xifrada (o funció *hash*) és la que autentica a l'usuari davant el servidor.

El mètode més eficient per evitar els atacs del tipus *man-in-the-middle* és la utilització de contrasenyes d'un sol ús. D'aquesta manera, si un atacant interceptés la contrasenya d'un usuari no podria utilitzar-la per suplantar-lo en accessos posteriors. Per prevenir aquest atac també cal autenticar al servidor mitjançant certificats d'infraestructura de clau pública (PKI). D'aquesta manera, el client té la certesa de comunicar-se amb el servidor utilitzant un transport SSL/TLS segur.

CAPÍTOL 3 - ANÀLISI

En aquest capítol es desenvolupa l'etapa d'anàlisi en què es van detallar els requisits funcionals i de seguretat que havia de complir el nostre gestor de contrasenyes. En primer lloc, a la fase d'anàlisi prèvia es defineixen les principals funcionalitats que han de definir el gestor, ampliant alguns dels requisits mínims de l'enunciat del TFC i descartant algunes funcionalitats opcionals. Posteriorment, durant la fase d'anàlisi de requisits es desenvolupen de manera molt més detallada totes les funcionalitats parant especial atenció als condicionants de seguretat que les afecten.

Ja en la fase d'anàlisi i per facilitar el desenvolupament del gestor de contrasenyes, les funcionalitats de l'aplicació s'estructuren en dos blocs: el **mòdul de gestió d'usuaris** i el **mòdul de gestió de contrasenyes**.

3.1. Anàlisi prèvia

En aquest punt del projecte es van definir les funcionalitats i requisits de seguretat que havien de configurar el gestor de contrasenyes. Es van ampliar els requisits mínims demanats per tal de dotar de major funcionalitat al gestor per a potencials usuaris. D'altra banda, es van descartar algunes funcionalitats opcionals per manca de temps deixant, però, oberta la possibilitat d'implementar-les en el futur.

3.1.1. Requisits funcionals

Els requisits funcionals que haurà d'oferir el gestor de contrasenyes són:

a) Ha de permetre el registre de nous usuaris

b) Ha d'autenticar els usuaris registrats que vulguin accedir-hi

c) Els usuaris han de poder desar les dades d'accés a serveis web

d) Els usuaris han de poder recuperar les dades d'accés prèviament desades

e) Els usuaris han de poder modificar qualsevol de les dades d'accés desades prèviament

f) Els usuaris han de poder eliminar les dades d'accés que hagin desat prèviament

g) Ha de permetre la baixa d'usuaris registrats i, en aquest cas, s'ha de garantir que tota la informació associada a un usuari és donada de baixa

h) No ha de permetre a usuaris no registrats accedir a cap informació

3.1.2. Requisits de seguretat

Els requisits de seguretat que ha de complir el gestor de contrasenyes són:

a) Qualsevol intercanvi d'informació entre els clients i el servidor s'ha de fer de manera xifrada

b) En cap moment els administradors del gestor de contrasenyes han de poder conèixer cap de les contrasenyes dels seus usuaris

c) S'ha de garantir l'autenticació del servidor mitjançant certificats digitals

d) S'ha de garantir l'autenticació dels usuaris mitjançant contrasenyes

e) S'han de gestionar les sessions de connexió al servidor per garantir-ne la seguretat

f) Per facilitar la usabilitat del gestor, l'usuari ha de tenir accés a totes les funcionalitats utilitzant només una sola contrasenya, que anomenarem **contrasenya mestra**

3.2. Anàlisi de requisits

Els requisits que hem detallat en el punt anterior defineixen l'aplicació que volem implementar. Però abans de passar a la fase de disseny cal que desenvolupem molt més aquests requisits. En la fase d'anàlisi prèvia hem descrit per separat les funcionalitats del gestor i els requisits de seguretat. El que farem a continuació serà desenvolupar-los conjuntament, detallant cada funcionalitat i veient com l'afecten els requisits de seguretat.

Classificarem les funcionalitats en dos grans blocs i, per facilitar la fase d'anàlisi i posteriors, estructurarem l'aplicació en dos mòduls: el mòdul de gestió d'usuaris i el mòdul de gestió de contrasenyes.

3.2.1. Mòdul de gestió d'usuaris

El mòdul de gestió d'usuaris incorpora les funcions que permetran a l'aplicació gestionar els usuaris. Així aquest mòdul serà l'encarregat d'autenticar als usuaris que vulguin accedir al gestor, així com de permetre el registre de nous usuaris i la baixa d'usuaris registrats.

D'acord amb els requisits de seguretat de l'aplicació, la gestió d'usuaris ve condicionada per dos aspectes:

a) L'autenticació d'usuaris s'ha de garantir mitjançant l'ús de contrasenyes, i aquestes no poden ser conegudes pels administradors del gestor

b) S'ha de garantir la seguretat de les sessions de connexió al servidor

3.2.1.1. Autenticació d'usuaris: requisits de seguretat

A continuació destallarem les característiques que ha de tenir el nostre mètode d'autenticació per garantir que compleix els requisits de seguretat definits i per garantir la màxima seguretat possible.

El servidor ha de desar les contrasenyes xifrades

Evidentment, les contrasenyes no podran guardar-se en clar a la base de dades i caldrà aplicar algun tipus de xifratge. Encara que el mètode més utilitzat per les aplicacions web és desar una funció *hash* unidireccional de la contrasenya de l'usuari, el mètode de xifratge que utilitzi el nostre gestor haurà de complir tres característiques addicionals:

a) El mètode de xifratge ha d'utilitzar bits de sal per tal d'evitar els atacs de diccionari

b) El mètode de xifratge ha de tenir un cost computacional elevat per tal de dificultar els atacs de força bruta augmentant els recursos necessaris per dur-lo a terme

c) El mètode de xifratge utilitzat ha de ser prou robust

Un mètode de xifratge que compleix tots aquests requisits i que utilitzarem en la nostra implementació és la xifra BlowFish. BlowFish utilitza bits de sal, té un cost computacional elevat ja que es calcula de manera iterativa que permet definir-ne el cost i, es considera prou robust a condició d'utilitzar una clau de 56 bytes [21].

El xifratge de la contrasenya s'ha de dur a terme en el client

Cap contrasenya de l'usuari, inclosa la d'accés al sistema, no ha d'arribar mai en clar al servidor, evitant d'aquesta manera que pugui ser coneguda. Cal tenir en compte que aquest requisit tindrà un impacte important en el disseny de l'arquitectura de l'aplicació ja que implica un pes important de la part de l'aplicació que s'executa en el client.

La cadena amb què s'autentica el client davant del servidor serà diferent en cada intent de connexió

Per tal de dotar de major seguretat al sistema d'autenticació de l'aplicació, s'ha valorat la utilització de contrasenyes d'un sol ús (one-time password, OTP) [22] i de sistemes d'autenticació basats en protocols de repte-resposta [23]. Aquestes dues tècniques, juntament amb l'autenticació de dos factors, s'estan consolidant per entorns que demanen un alt nivell de seguretat (banca electrònica, comerç electrònic, entorns corporatius,...).

Descartem els mètodes més sofisticats d'OTP que impliquen la distribució fora de línia de la contrasenya que ha d'introduïr el client a través de tokens sincronitzats amb el servidor, SMS o instal·lant generadors pseudoaleatoris en dispositius mòbils. Malgrat això, el nostre sistema d'autenticació ha de fer que la cadena amb la que el client s'autentica davant el servidor sigui diferent en cada cas. Aquest requisit permet evitar els atacs del tipus *man-in-the-middle*.

El factor que farà diferent aquesta cadena en cada intent de connexió serà que en el seu càlcul intervindrà un nombre aleatori que proveïrà el servidor. És a dir, el nostre sistema d'autenticació incorporarà una implementació simple del protocol de repte-resposta (*challenge-response authentication*) com es mostra a la figura 1.

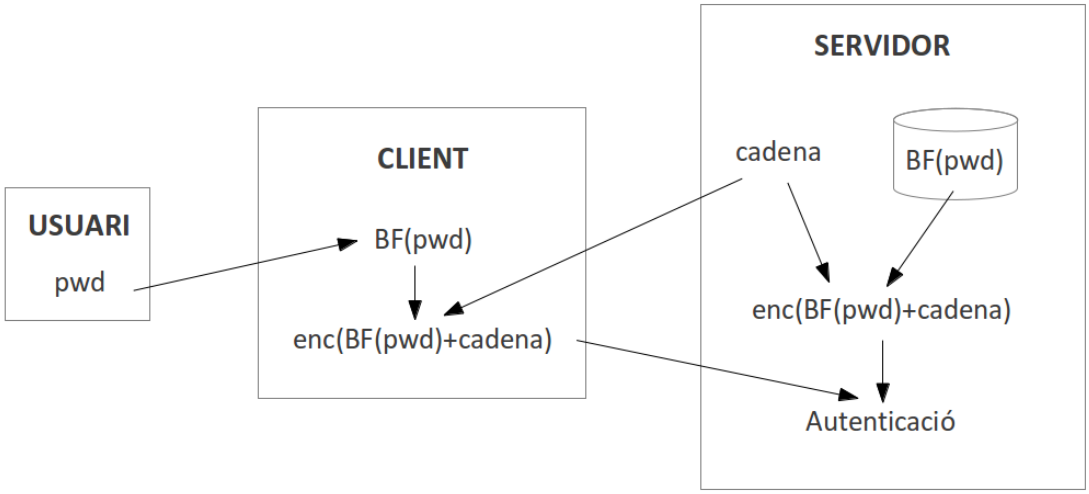

Figura 1 - Esquema del sistema d'autenticació

3.2.1.2. Seguretat de les sessions

L'aplicació ha de gestionar les sessions de connexió dels usuaris al servidor per tal de garantir la seguretat i evitar que la sessió pugui ser segrestada per un atacant. Malgrat això, cal destacar que encara que un atacant pogués accedir a la sessió i recuperar les dades dels usuaris desades a la base de dades, no tindria accés a les dades d'accés als serveis web ja que per desencriptar-les necessitaria saber la contrasenya mestre que en cap cas arriba al servidor.

El control de les sessions haurà de complir:

a) No s'ha de permetre que un usuari es connecti a l'aplicació mentre tingui una sessió oberta. És a dir, un usuari no pot accedir a l'aplicació des de dos dispositius alhora.

b) Si un usuari no interactua amb l'aplicació durant 5 minuts, el servidor ha de tancar la sessió i l'usuari haurà de tornar a autenticar-se.

c) El servidor rebutjarà qualsevol petició que no provingui de l'adreça IP des de la que l'usuari es va autenticar.

3.2.1.3. Validació d'usuaris

Un requisit complementari del gestor de contrasenyes és que el nom d'usuari amb que es registra a l'aplicació ha de ser una adreça de correu vàlida. Per tal de validar això, finalitzat el procés de registre, l'usuari ha d'activar el seu compte mitjançant un enllaç que se li envia per correu electrònic. Sense aquesta activació, l'usuari no podrà accedir al gestor de contrasenyes.

El motiu que justifica aquest requisit és que, si en el futur s'implementés la compartició de contrasenyes, els usuaris que vulguessin compartir una contrasenya amb altres usuaris els haurien d'identificar mitjançant el correu electrònic.

3.2.2. Mòdul de gestió de contrasenyes

El mòdul de gestió de contrasenyes és l'encarregat de gestionar les contrasenyes desades pels usuaris. Ha de permetre als usuaris emmagatzemar les dades d'accés a diferents serveis web, així com recuperar-les, modificar-les i eliminar-les.

La gestió de contrasenyes ha de complir els següents requisits:

a) Per cada servei el sistema ha de guardar el nom del servei, el nom d'usuari i la contrasenya. El conjunt d'aquests tres elements l'anomenarem dades d'accés.

b) Totes les dades desades han d'estar encriptades en el client. És a dir, a més de no tenir accés a la contrasenya en clar, els administradors del gestor tampoc no han de conèixer els noms d'usuari ni quins són els serveis dels quals l'usuari desa les dades en el gestor.

c) S'utilitzarà un mètode de xifratge simètric ja que no es produeix cap intercanvi d'informació entre el servidor i el client[4]

d) La clau pel xifratge de les dades d'accés abans de enviar-les al servidor ha de derivar-se de la contrasenya mestra de l'usuari

e) El procés de derivació de la clau haurà d'incloure la utilització de bits de sal per dificultar els possibles atacs contra contrasenyes

f) S'utilitzaran bits de sal diferents -i, per tant, claus diferents- pel xifratge de cada una de les claus d'accés

Encriptació de les dades d'accés

L'encriptació de les dades d'accés a un servei web utilitzarem AES amb una clau de 256 bits. Aquesta tria es considera la més adequada ja que AES és un estàndar pel xifratge simètric que es considera actualment prou segur [24].

Per generar la clau a partir de la contrasenya mestre de l'usuari, utilitzarem la funció de derivació de claus PBKDF2 [25] (Password-Based Key Derivation Function 2). Aquesta funció de derivació aplica iterativament diferents algorismes criptogràfics sobre la contrasenya original a la que s'afegeixen uns bits de sal addicionals.

3.2.3. Nivells de xifratge

L'aplicació preveu diferents nivells de xifratge de les dades a fi d'optimitzar la protecció d'aquestes i garantir la confidencialitat, l'autenticitat i la integritat de les comunicacions. Fins ara ens hem centrat en el xifratge de les dades a nivell d'aplicació, que es realitza de la banda del client. Però de manera complementària s'aplicaran mètodes de xifratge criptogràfic a altres nivells: la capa de transport i la base de dades.

La figura 2 il·lustra els tres tipus de xifratge que s'aplica a les dades.

4 Independentment del fet que tota la comunicació sigui xifrada utilitzant un protocol de transport SSL.

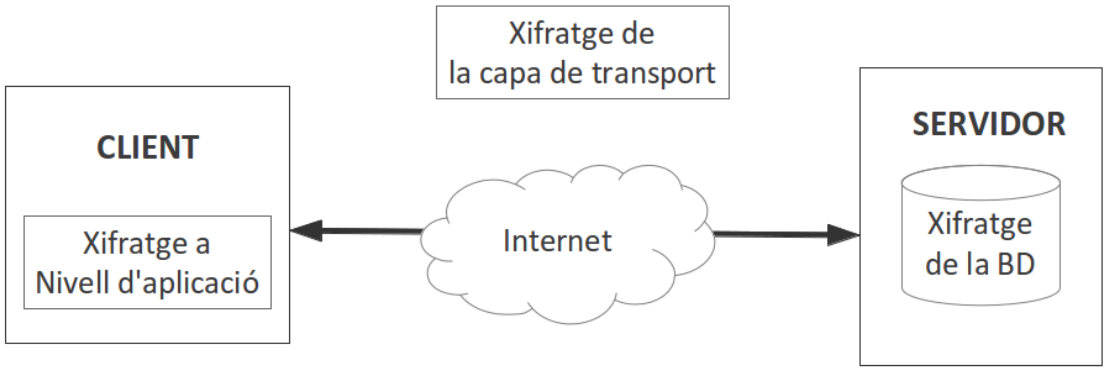

Figura 2 - Nivells de xifratge

3.2.3.1. Xifratge de les dades de transport

La connexió entre el servidor i el client es realitza a través d'Internet i, per garantir-ne la seguretat, totes les comunicacions han de ser xifrades a nivell de la capa de transport, utilitzant un protocol TLS/SSL (*Transport Layer Security/Secure Socket Layer).*

Encara que pràcticament totes les dades són xifrades i no s'envien mai xifrades a nivell d'aplicació, la utilització d'aquesta mesura de seguretat addicional és necessària per dos motius principals:

a) permet l'autenticació del servidor mitjançant certificats digitals

b) en el moment del registre el client envia la seva contrasenya d'accés al gestor encriptada amb BlowFish[5]. Un atacant que interceptés aquesta comunicació podria en el futur suplantar la identitat de l'usuari i guanyar accés a les seves dades[6], facilitant els atacs de criptoanàlisi[7].

3.2.3.2. Encriptació de la base de dades

Pràcticament totes les dades que es desen a la base de dades, i totes les que contenen informació dels usuaris es desen encriptades en la base de dades. Per tant, hem d'entendre l'encriptació de la base de dades com una mesura redundant de seguretat.

Aquesta mesura és d'utilitat per evitar que un possible atacant que hagi guanyat accés a la base de dades, pugui recuperar les dades encriptades tal com les ha enviat el client. D'aquesta manera, es dificulta el treball de criptoanàlisi de l'atacant que, abans de mirar de desencriptar les dades d'accés dels usuaris, ha de desencriptar la base de dades.

Aquest requisit de seguretat, obligarà a utilitzar un sistema de gestió de bases de dades (SGBD) que implementi aquesta funcionalitat. La majoria dels sistemes del mercat ofereixen aquesta possibilitat.

5 El client i el servidor han de compartir alguna informació per tal de dur a terme el procés d'autenticació i aquesta informació és la contrasenya mestra encriptada amb BlowFish. Aquest enviament només té lloc durant el procés de registre.

6 Recordem que encara que un atacant pogués accedir a les dades de l'usuari, per accedir en clar a les dades d'accés que aquest té emmagatzemades encara hauria de fer un atac de criptoanàlisi sobre la xifra AES amb que estan encriptades.

7 Si l'atacant té accés a les dades xifrades, podrà desar-les i provar atacs fora de línia.

3.2.4. Altres requisits addicionals de seguretat

3.2.4.1. Redundància de les bases de dades

La informació que els usuaris emmagatzemen al gestor de contrasenyes és crítica ja que sense accés a aquesta informació podrien no tenir accés a serveis web necessaris per a la seva vida personal o professional. Per tant, no només hem de garantir la seguretat de les dades, sinó que també hem de garantir la disponibilitat d'accés a aquestes.

Per aquest motiu, un requisit bàsic de l'arquitectura de l'aplicació ha de ser preveure la redundància de la base de dades amb, com a mínim, dos servidors localitzats en indrets diferents per què, en cas de caiguda d'un d'ells, l'altre pugui seguir donant servei als usuaris.

Addicionalment, s'ha d'establir una política conservadora de còpies de seguretat de tots dos servidors. La nostra aplicació haurà de preveure una còpia de seguretat complerta un cop per setmana i una còpia de seguretat incremental cada dia.

Aquesta funcionalitat no està implementada en el producte final que s'entrega amb aquesta memòria. Cal entendre-la com una recomanació alhora que aquest gestor hagués de passar a la fase de producció.

3.2.4.2. Sistemes de prevenció i detecció d'intrusions

Com a darrer requisit de seguretat, destacarem que caldrà protegir els servidors on s'instal·lin les aplicacions i les bases de dades amb els mecanismes de prevenció d'intrusions (tallafocs) i amb els mecanismes de detecció d'intrusions (IDS) addients. Aquests sistemes de protecció hauran de respondre a polítiques de seguretat que, per la naturalesa de les dades que protegeixen, hauran de ser molt restrictives i conservadores.

CAPÍTOL 4 - DISSENY

4.1. Arquitectura de l'aplicació

A la següent figura podem veure l'arquitectura de l'aplicació.

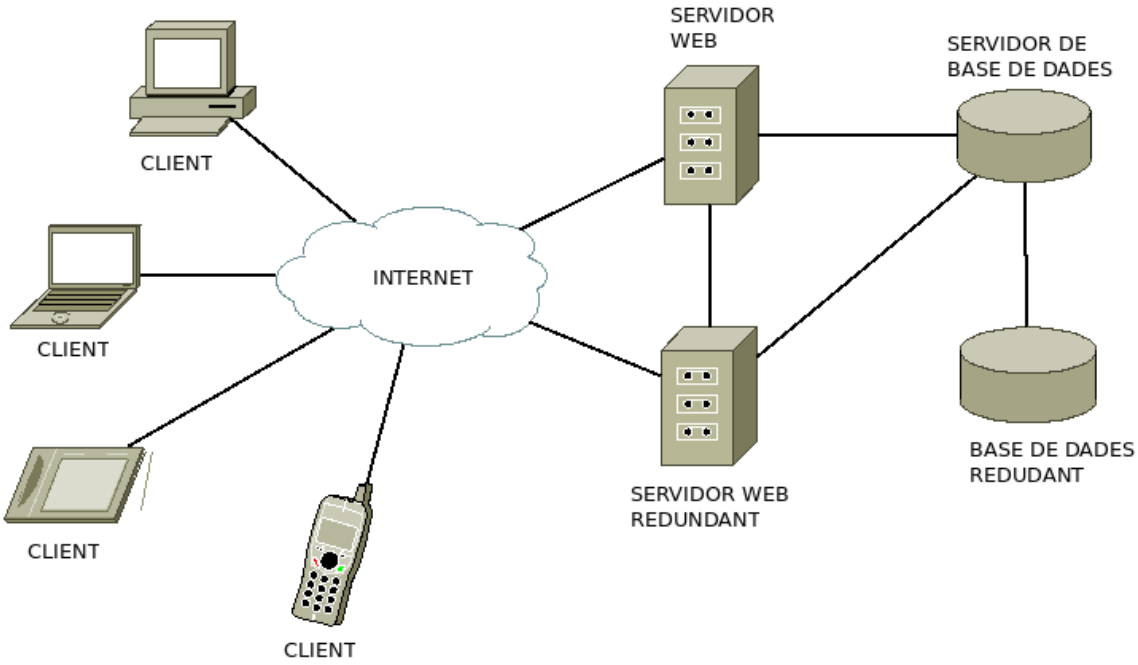

Figura 3 - Arquitectura de l'aplicació

L'arquitectura bàsica del gestor de contrasenyes serà relativament senzilla. Els elements imprescindibles per la seva implementació son un servidor web connectat a Internet i a un servidor de base de dades. Els clients es connectaran a aquest servidor a través d'Internet.

En la figura podem veure, però, que donat que garantir la disponibilitat del servei és un dels requisits imposats en l'implementació del gestor de contrasenyes, el servidor web ha d'estar recolzat per un altre servidor que, en cas de fallada del servidor web principal, pugui seguir oferint servei als clients.

Igualment, complint els requisits de seguretat també està duplicat el servidor de base de dades i la base de dades mateixa. Aquestes dues bases de dades s'han de mantenir sincronitzades per tal de garantir la continuïtat del servei en cas de caiguda d'un dels dos servidors de bases de dades.

Finalment, comentar que en l'arquitectura presentada una part molt important de l'execució de l'aplicació té lloc en els clients. Podem parlar per tant, d'una arquitectura amb clients pesats. El codi a

executar és descarregat en el client i executat en aquest. Com es pot veure en la figura, l'aplicació accepta qualsevol dispositiu que tingui la capacitat de connectar-se a Internet mitjançant un navegador web.

4.2. Casos d'ús

4.2.1. Diagrama de casos d'ús

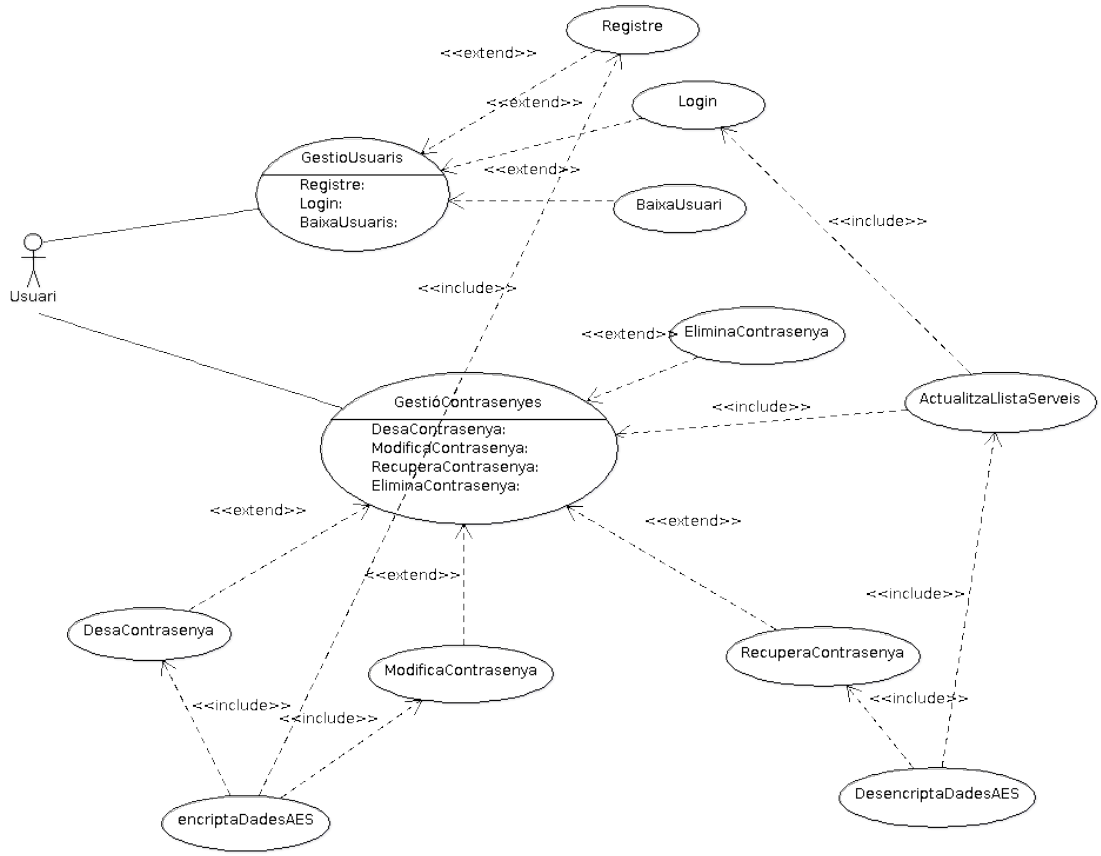

Figura 4 - Diagrama de casos d'ús

4.2.2. Descripció dels casos d'ús

4.2.2.1. Cas d'ús "GestioUsuaris"

- ➔ Resum de la funcionalitat: Aquest cas d'ús agrupa tots els casos d'ús que s'encarreguen de la gestió d'usuaris.
- ➔ Actor: Usuari
- ➔ *Extension points*: "Registre", "Login" i "BaixaUsuari".

4.2.2.2. Cas d'ús "Registre"

- ➔ Resum de la funcionalitat: Permet als usuaris registrar-se com a usuaris del gestor de contrasenyes

- Actor: Usuari
- Casos d'ús relacionats: encriptaDadesAES
- Precondició: Cap
- Postcondició: L'usuari està registrat en l'aplicació

4.2.2.3. Cas d'ús "Login"

- Resum de la funcionalitat: Permet als usuaris registrats al sistema autenticar-se per gestionar les seves dades d'accés
- Actor: Usuari
- Casos d'ús relacionats: "ActualitzaLlistaServeis"
- Precondició: L'usuari està correctament registrat en el sistema
- Postcondició: L'usuari pot iniciar una sessió en l'aplicació i gestionar les seves dades d'accés

4.2.2.4. Cas d'ús "BaixaUsuari"

- Resum de la funcionalitat: Permet als usuaris registrats donar-se de baixa de l'aplicació
- Actor: Usuari
- Casos d'ús relacionats: Cap
- Precondició: L'usuari està correctament registrat en l'aplicació i ha iniciat una sessió autenticant-se satisfactòriament
- Postcondició: Totes les dades corresponents a l'usuari s'han eliminat de la base de dades de l'aplicació

4.2.2.5. Cas d'ús "GestioContrasenyes"

- Resum de la funcionalitat: Aquest cas d'ús agrupa tots els casos d'ús que s'encarreguen de la gestió d'usuaris
- Actor: Usuari
- *Extension points*: "DesaContrasenya", "ModificaContrasenya", "RecuperaContrasenya" i "EliminaContrasenya"
- Casos d'ús relacionats: El cas d'ús "ActualitzaLlistaServeis" s'executa en finalitzar tots els casos d'ús que són extensions d'aquest.

4.2.2.6. Cas d'ús "DesaContrasenya"

- Resum de la funcionalitat: Permet a l'usuari desar unes dades d'accés a un servei web (nom del servei, nom d'usuari i contrasenya)
- Actor: Usuari
- Casos d'ús relacionats: "EncriptaDadesAES", "ActualitzaLlistaServeis"
- Precondició: L'usuari està correctament registrat en l'aplicació i ha iniciat una sessió autenticant-se satisfactòriament

- Postcondició: Les dades d'accés a un servei web s'han desat correctament i s'ha actualitzat la llista de serveis a la interfície d'usuaris

4.2.2.7. Cas d'ús "ModificaContrasenya"

- Resum de la funcionalitat: Permet a l'usuari modificar unes dades d'accés que prèviament ha desat en el gestor de contrasenyes
- Actor: Usuari
- Casos d'ús relacionats: "EncriptaDadesAES" i "ActualitzaLlistaServeis"
- Precondició: L'usuari està correctament registrat en l'aplicació, ha iniciat una sessió autenticant-se satisfactòriament i les dades d'accés que vol modificar estan desades en el gestor de contrasenyes
- Postcondició: Les modificacions s'han desat a la base de dades de l'aplicació i s'ha actualitzat la llista de serveis a la interfície d'usuari

4.2.2.8. Cas d'ús "RecuperaContrasenya"

- Resum de la funcionalitat: Permet a l'usuari recuperar les dades d'accés a un servei web
- Actor: Usuari
- Casos d'ús relacionats: "DesencriptaDadesAES" i "ActualitzaLlistaServeis"
- Precondició: L'usuari està correctament registrat en l'aplicació, ha iniciat una sessió autenticant-se satisfactòriament i les dades d'accés que vol recuperar estan desades en el gestor de contrasenyes
- Postcondició: Les dades d'accés es recuperen de la base de dades, es desencripten en el client i es mostren a l'usuari

4.2.2.9. Cas d'ús "EliminaContrasenya"

- Resum de la funcionalitat: Permet a l'usuari eliminar unes dades d'accés del gestor de contrasenyes
- Actor: Usuari
- Casos d'ús relacionats: "ActualitzaLlistaServeis"
- Precondició: L'usuari està correctament registrat en l'aplicació, ha iniciat una sessió autenticant-se satisfactòriament i les dades d'accés que vol eliminar estan desades en el gestor de contrasenyes
- Postcondició: Les dades d'accés del servei s'han eliminat de la base de dades i s'ha actualitzat la llista de serveis en la interfície d'usuari

4.2.2.10. Cas d'ús "ActualitzaLlistaServeis"

- Resum de la funcionalitat: Actualitza la llista de serveis que l'usuari té actualment desats al gestor de contrasenyes
- Actor: Usuari

- ➔ Casos d'ús relacionats: "DesencriptaDadesAES"
- ➔ Precondició: L'usuari està correctament registrat en l'aplicació, ha iniciat una sessió autenticant-se satisfactòriament
- ➔ Postcondició: En la interfície d'usuari es mostra la llista de tots els serveis que l'usuari té actualment desats en el gestor de contrasenyes o cap si no en té cap de desada

4.2.2.11. Casos d'ús "EncriptaDadesAES" i "DesencriptaDadesAES"

En sentit estricte, aquestes funcions no són casos d'ús ja que no són funcions que puguin ser engegades per cap actor -els usuaris en aquest cas- i només són invocades per altres funcions. Malgrat això, s'han inclòs en el diagrama perquè crec que aporten informació.

4.3. Diagrames d'activitats

A continuació veurem els diagrames d'activitats dels diferents processos de l'aplicació. Aquests diagrames complementen el diagrama de casos d'ús i especifiquen les funcions que han d'intervenir en cada un d'ells.

4.3.1. Mòdul de gestió d'usuaris

4.3.1.1. Procés de registre

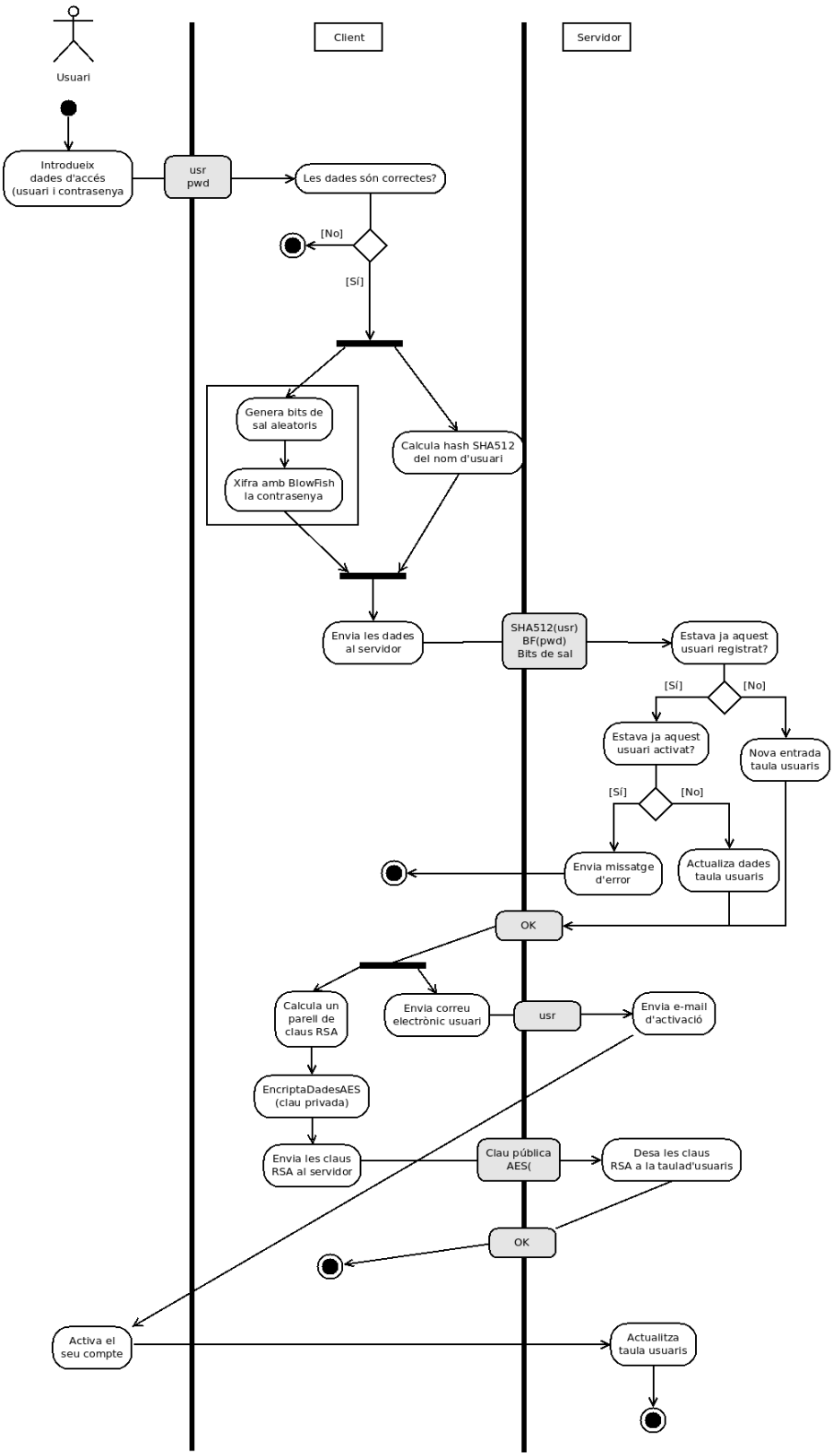

Figura 5 - Diagrama d'activitats del procés de registre

4.3.1.2. Procés d'autenticació

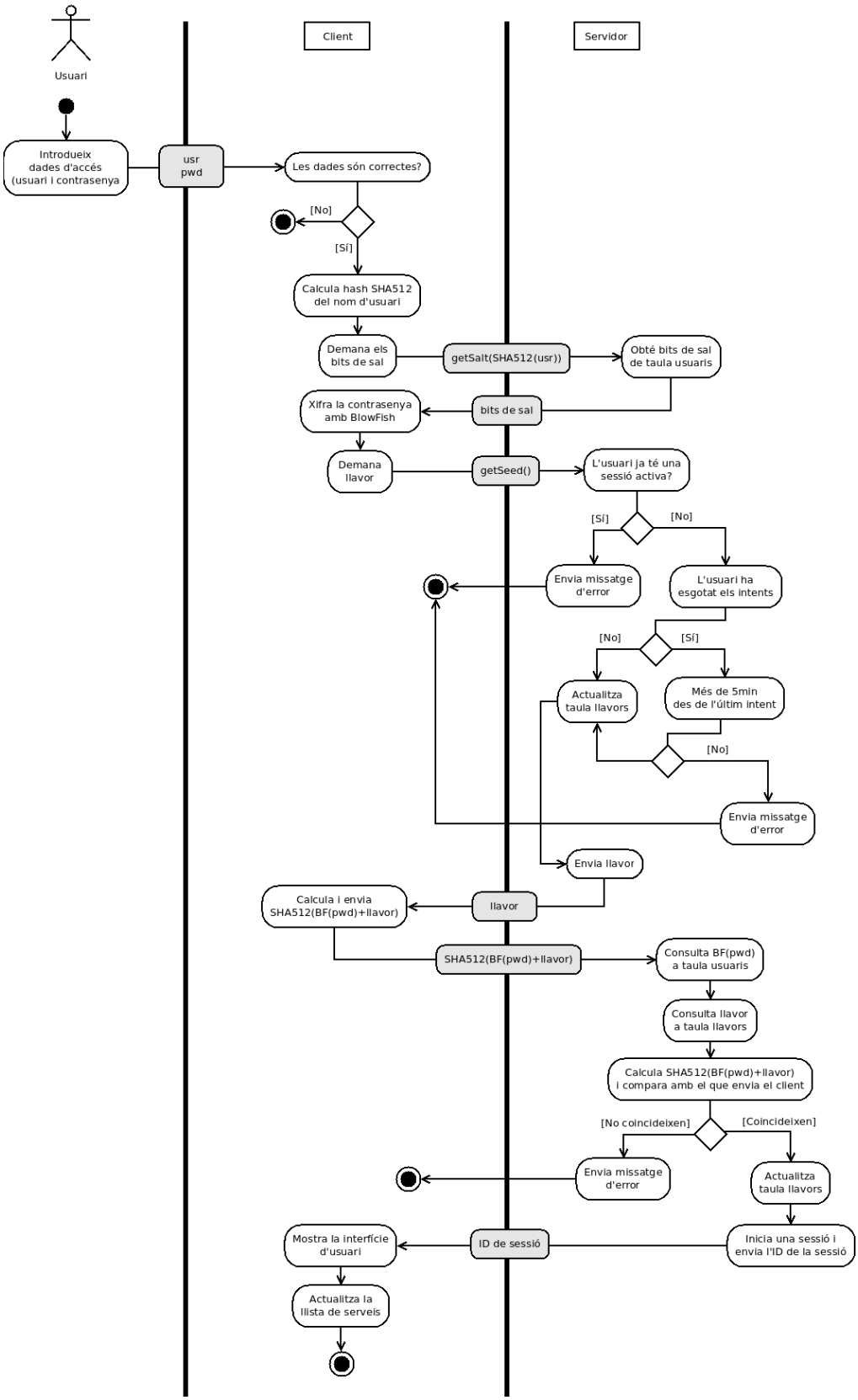

Figura 6 - Diagrama d'activitats del procés d'autenticació

4.3.1.3. Procés de baixa d'usuaris

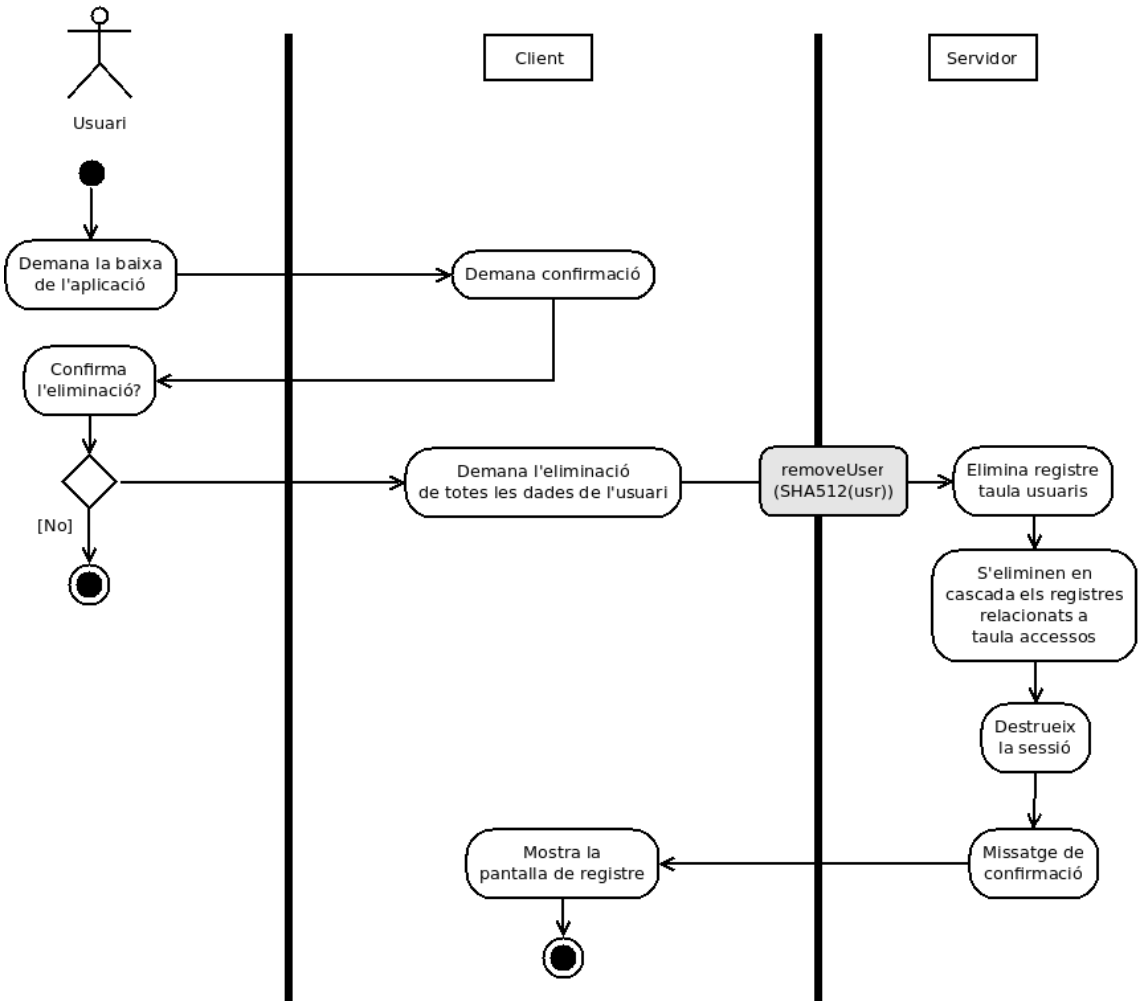

Figura 7 - Diagrama d'activitats del procés de baixa d'usuaris

4.3.2. Mòdul de gestió de contrasenyes

4.3.2.1. Desar unes dades d'accés

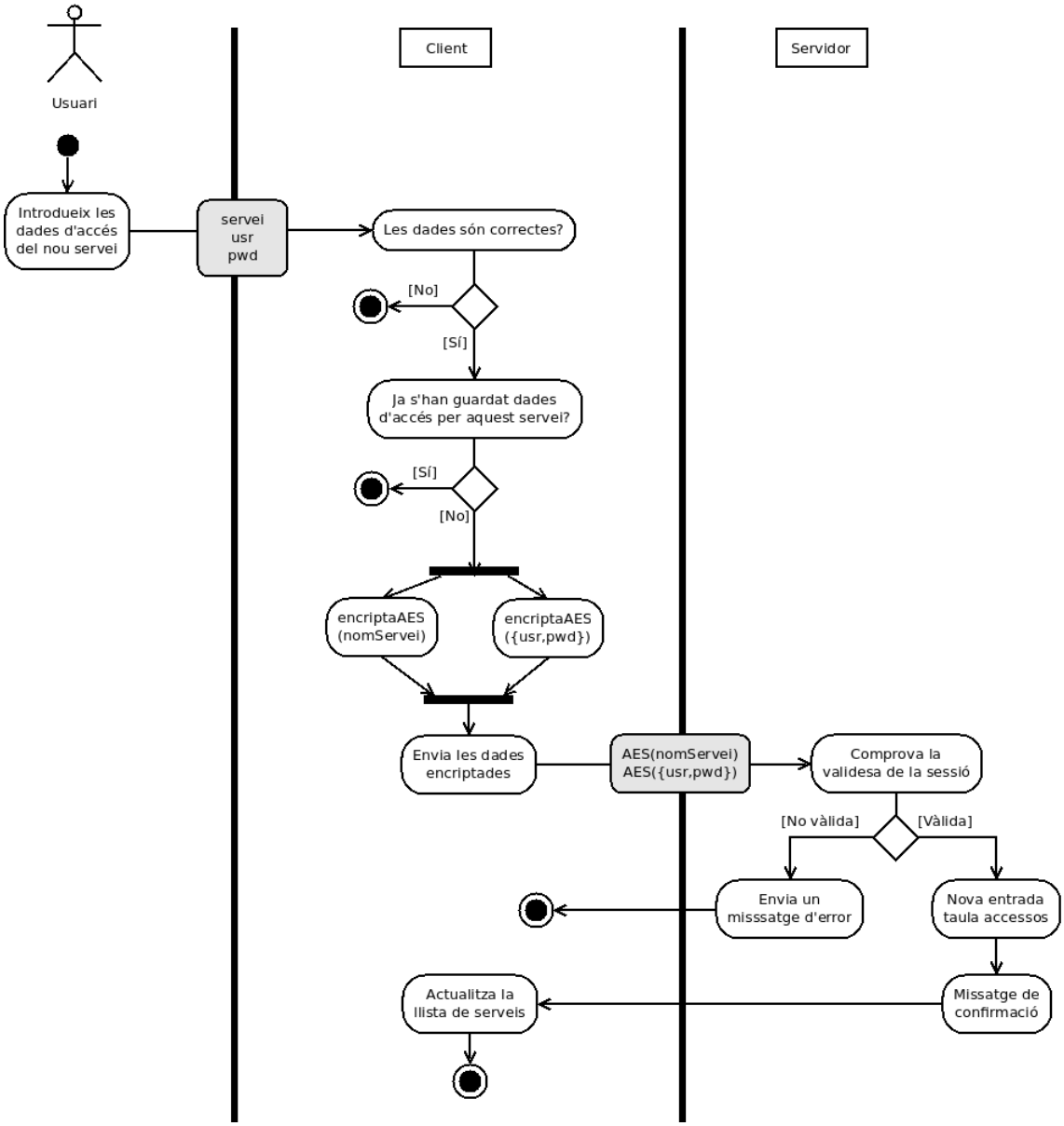

Figura 8 - Diagrama d'activitats del procés de desar una contrasenya

4.3.2.2. Recuperar unes dades d'accés

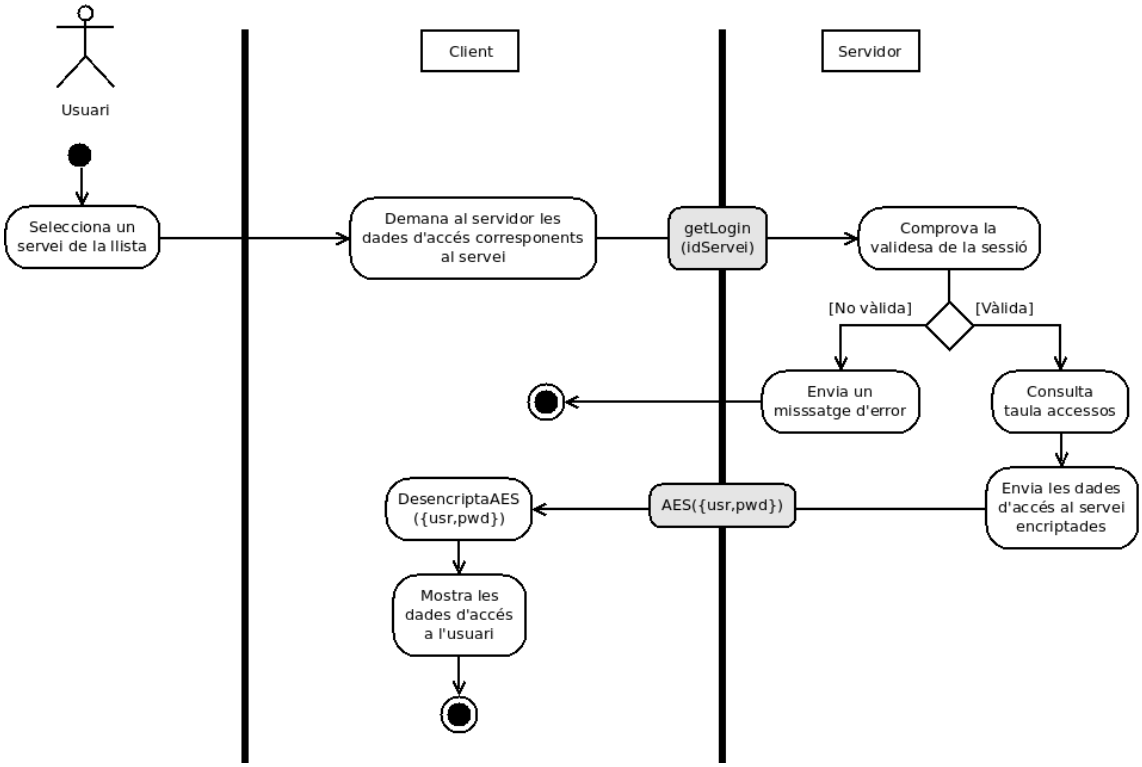

Figura 9 - Diagrama d'activitats del procés de recuperar una contrasenya

4.3.2.3. Modificar unes dades d'accés

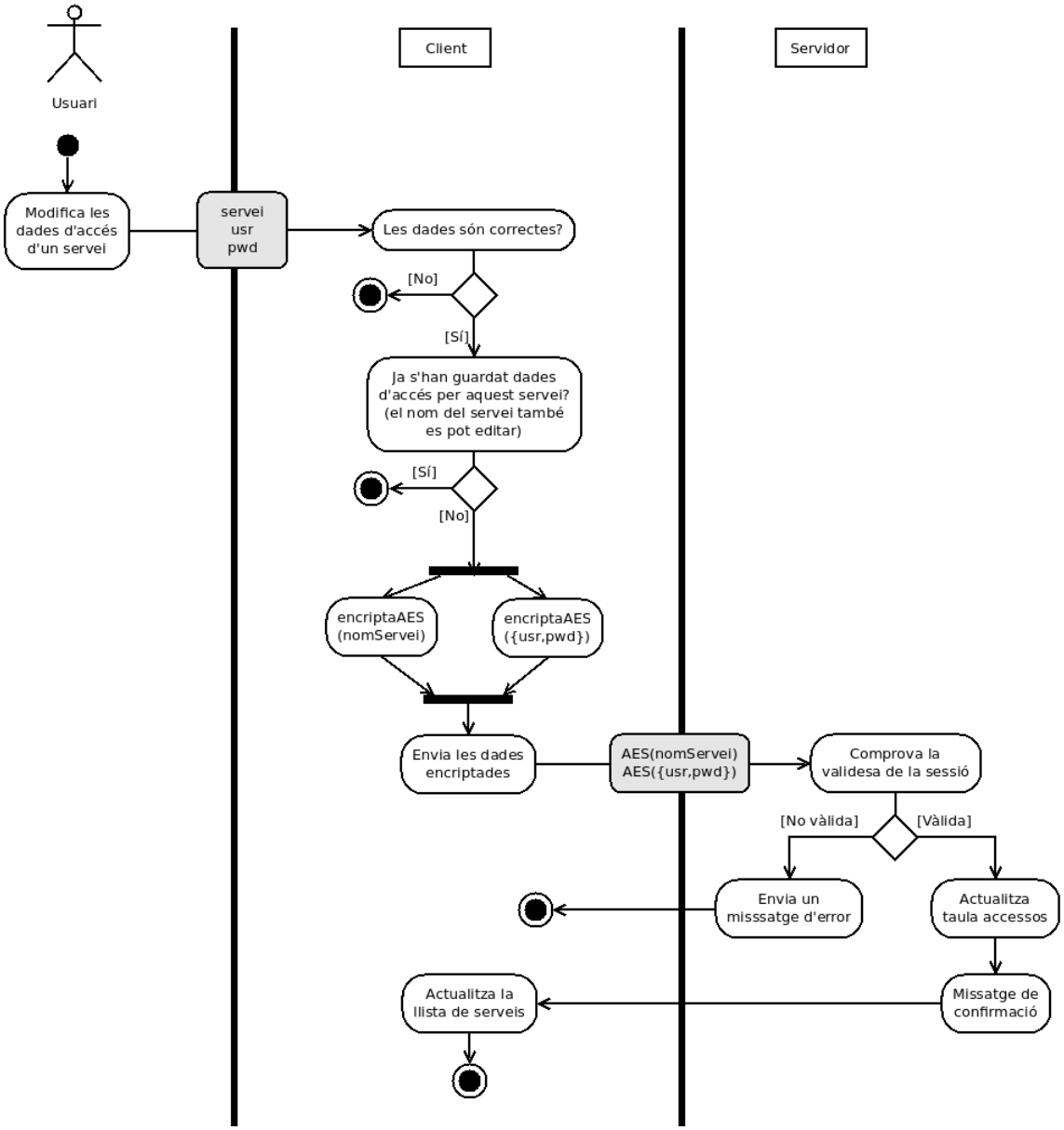

Figura 10 - Diagrama d'activitats del procés de modificar una contrasenya

4.3.2.4. Eliminar unes dades d'accés

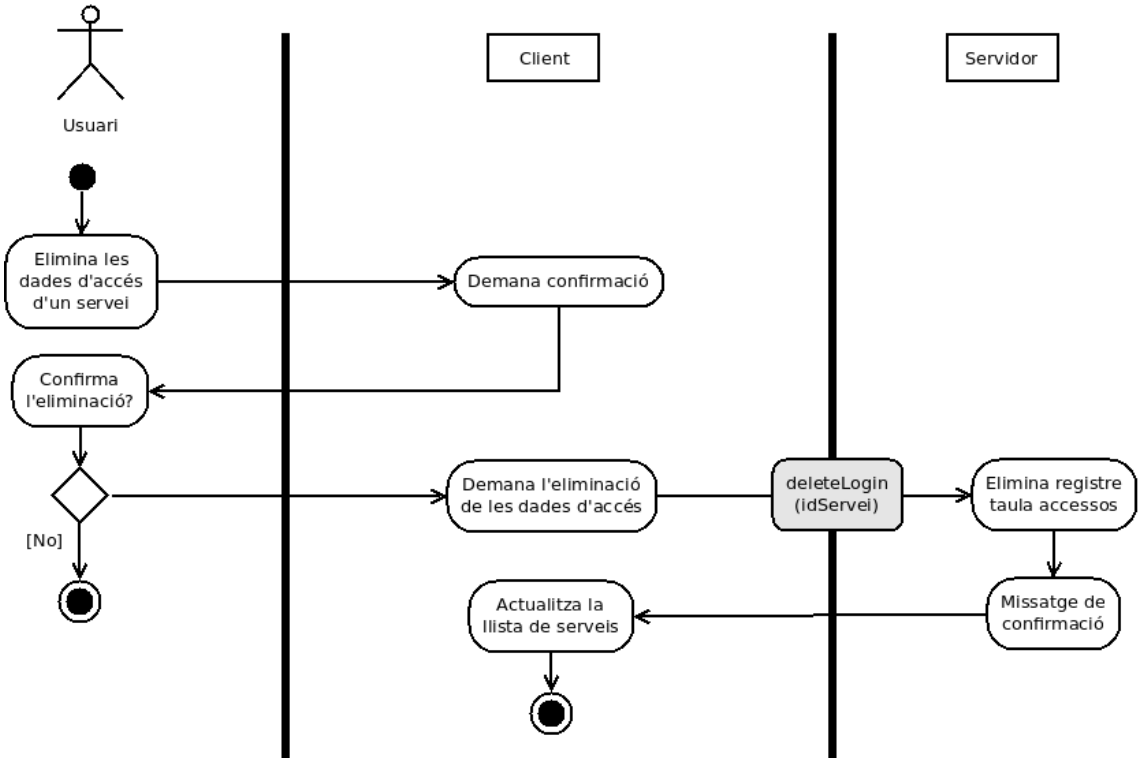

Figura 11 - Diagrama d'activitats del procés d'eliminar una contrasenya

4.4. Algorismes criptogràfics a utilitzar

En l'etapa d'anàlisi es van detallar els requisits de seguretat que havia de complir l'aplicació i es van apuntar alguns mètodes de xifratge que complien aquests requisits. A continuació veurem quins algorismes criptogràfics utilitzarem per protegir les dades dels usuaris.

4.4.1. Xifratge de la contrasenya mestra

En l'etapa de disseny vam veure la necessitat de xifrar la contrasenya mestra de l'usuari de manera que aquesta s'emmagatzemés xifrada a la base de dades d'una manera que ni tan sols els administradors de l'aplicació en tinguessin accés.

Els requisits que es van imposar al mètode de xifratge eren: a) havia d'utilitzar bits de sal, b) havia de tenir un cost computacional elevat, i c) havia de ser robust i resistent a atacs de criptoanàlisi.

Una xifra que compleix totes aquestes condicions és BlowFish, un algorisme publicat per Bruce Schneider el 1993 [21][26]. La millor manera d'utilitzar BlowFish per xifrar contrasenyes abans d'emmagatzemar-les en el servidor és utilitzant BCrypt [27].

BCrypt és una funció de derivació de claus que implementa l'algorisme BlowFish [28] i que s'ha utilitzat com equivalent de les funcions *hash*. La principal característica que aporta BCrypt és que introdueix un factor de cost que permet definir el cost computacional i incrementar-lo tant com vulguem. Per això, alguns autors recomanen aquest algorisme per desar contrasenyes [29][30].

Malgrat que es considera un bon algorisme per desar contrasenyes, el NIST recomana la utiltització de PBKDF2 per aquesta finalitat [31]. Però crec que el factor cost computacional justifica l'elecció de BCrypt i comparteixo les opinions de Thomas Pornin en una resposta molt ben documentada en el fòrum d'IT Security [31].

Finalment, mencionar també el desenvolupament d'un altre algorisme per encriptar contrasenyes[8] que és scrypt [32]. Segons els seus autors, millora les prestacions de BCrypt i de PBKDF2, però la poca documentació disponible fins a la data i el fet que les implementacions en JavaScript d'aquest algorisme encara semblen estar en fase experimental [33], m'ha fet desistir d'utilitzar scrypt per desar les contrasenyes mestres.

4.4.2. Encriptació de les dades d'accés

Per encriptar les dades d'accés a diferents serveis web que els usuaris guarden en el gestor de contrasenyes, els requisits que afecten a l'hora de triar un algorisme de xifratge són: a) ha de ser un mètode de xifratge simètric, b) s'ha de derivar la clau a partir de la contrasenya mestra, c) la funció de derivació de la clau ha d'utilitzar bits de sal, i d) ha de ser mètode robust contra el que no s'han d'haver descrit atacs criptoanalítics significatius.

El mètode de xifratge que he triat és AES (*Advanced Encription Standard*) [24]. No utilitzem BCrypt com en el cas de la contrasenya mestra perquè el cost computacional elevat no és un requisit. Per contra, si el cost computacional fos elevat podria alentir el funcionament de l'aplicació. Utilitzarem un vector d'inicialització i el xifratge es durà a terme utilitzant el mode CBC de xifratge de bloc.

Conjuntament amb AES, hem de definir una funció que ens permeti derivar les claus de 128 bits que utilitzarem en cada xifratge amb AES. En aquest cas, triem PBKDF2 [25] ja que és una funció que utilitza bits de sal i és una de les funcions de derivació de claus més utilitzada sense que s'hagin descrit incidències significatives. Addicionalment, PBKDF2 permet definir el nombre d'iteracions a partir de les que es derivarà la clau reforçant-ne la seguretat. En el nostre cas, establirem la xifra de 1.000 iteracions per derivar la clau a partir de la contrasenya mestra de l'usuari.

8 En puritat, tant PKBDF2 com BCrypt i scrypt són funcions de derivació de claus. Però degut a la seva utilització per desar de manera segura contrasenyes, ens referim a aquests mètodes com algorismes de xifratge. També és usual, per la seva funció, referir-s'hi com a funcions *hash*.

4.5. Estructura de les dades

4.5.1. Estructura de la base de dades: diagrama relacional

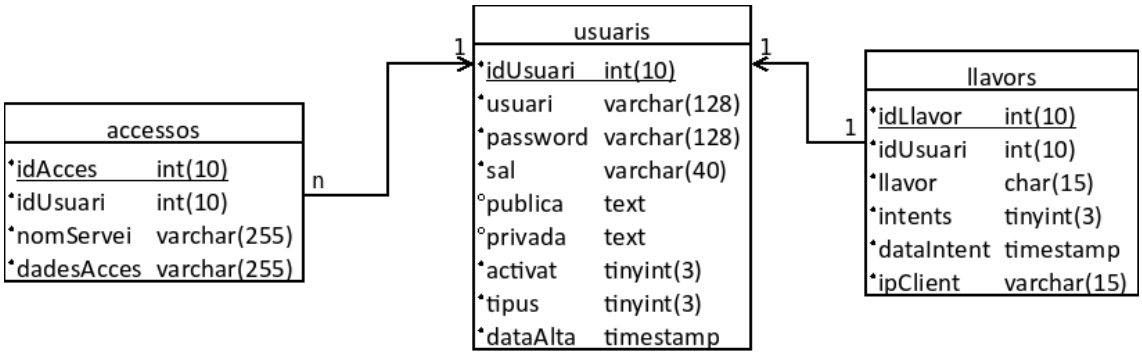

Figura 12 - Diagrama relacional de la base de dades

4.5.2. Estructura de la base de dades: descripció de les taules

A continuació es descriuen els continguts de les diferents taules, amb especial èmfasi en les estructures que inclouen les dades encriptades. Cal tenir en compte que les dades es presenten sense aplicar el xifratge a nivell de la base de dades. Malgrat això, en la majoria dels casos, es tracta d'informació que ja ha estat xifrada a nivell d'aplicació en el client.

4.5.2.1. Taula "usuaris"

La taula "usuaris" guarda la informació d'accés als usuaris al gestor de contrasenyes *online*. A continuació, mostrem un registre d'aquesta taula que servirà d'exemple per comentar el tipus d'informació que emmagatzema.

Camp	Valors
idUsuari	109
usuari	1dde020278ec206e519ea5b0ecdcda48d6057c554db768c7f75d356b42b1798a45e615e1c6fd4cb63c6ef4c4204aeb30b404307ea0a3fc7b45d3c013cfc3e9db
password	$2a$05$2xRdY83VxLXrQBUzXrbyqOsHLCaqk4HB0iFjBXVjOQDSn2kgbcnca
sal	$2a$05$2xRdY83VxLXrQBUzXrbyqO
publica	{\"n\":\"5743b208b04d0f2a62e6950bf74e334785a373441e421cb823cb929fd2369ca0dc16d1eb2c4fbf9d6b8d3d097b3af7982bf122920588207317e6f1132ac8df75b2accc04710d164d5bb782d610768444abfecd8b72592af0f8bc289bd7361a3e00345aed3a3f3a787e1ee931684fd2bbccbb838ad447dc5afc5afdcc3be732dd\",\"e\":\"10001\"}
privada	{\"ct\":\"LBEy+4s5cOmQ14wXcdeFGvCmntSw6FqpJBYsmMoXAlbO6kAJRklRq/hdvpMO7hGpUtEIljOCDZXvbX6KXkHtUoQ3m5q5PL2h4qBMOp+SX2BTEyQRsEkIMdlK/q3KVo6nlZENuH0572+SZpWcsk8Cmpq2cazU3f9HYb+wtc87aoZuWErgvqrxcL8f1zpDCdHKV6pQRWktal+ZwmgTMvFbnuCFAl6KDY+ZuiifcK5cDmA3nPiLlA1Hyb9OjYwyGsOwQot8

	YEZKh2khiuxWlI4itD9qWy84sJxnc45jVxm1HQeoy5QNxE0s4t/T9LHKt1f6jkG22+0xC0Y HtXNK30IASp8ewaQqPG70LfUc5FQMJ6sYw9RLY+vIYIIXTTZ7KpjZ6vg/fDcSQmtlcAIyuE 3d9+BR9WUWq3b4kRSf2ZtK3+ +X3uluiseB/QdsapNlXjEaZlUe5VYwVz0S5UgKVACiLwb8xWJZxl+s8aZvc13nNGQurRr 6tg5J+WXFXCU51Y8AX8f6z/hdPIeHy13hkgO8hhMKPbG9JzubtmfumxO/04Yiz6+wEgf t2j5ljILK/QnVpTiS/Tmfx2ldJg5HwvBSJJDXTaBOnR1npKS1phL/A6ZrJM+XXwsWzflmenq Vg3lyMDBXif2Dw265XimSQGPnecWhN25Ua1IWDaHUrdlnYPgUiziYEEjUCxKGLo2qU3 aq7UqZl58ZEo14lSve0HNgcA==\",\"iv\":\"f316476e634d0267bd10bc2a37aa5a64\",\ "sal\":\"155274fafdfd06b4e3441eac91111b49\"}
activat	1
tipus	2
dataAlta	2013-05-02 13:09:53

Taula 1 - Exemple de registre de la taula d'usuaris

- ➔ El camp "idUsuari" és la clau principal de la taula.
- ➔ El valor del camp "usuari" correspon a la funció *hash* SHA-512 aplicada al nom d'usuari introduït per l'usuari que, necessàriament, ha de ser una adreça de correu electrònic vàlida.
- ➔ El camp "password" emmagatzema la contrasenya mestra de l'usuari xifrada amb BlowFish. Els valors tenen el format següent que analitzem a partir del valor del registre d'exemple:
  - **$2a** - Indica el tipus de xifra aplicat (BlowFish en aquest cas)
  - **$05** - Indica el nombre d'iteracions aplicades. S'ha considerat que el cost computacional que suposen 5 iteracions són suficients per complir els requisits de seguretat imposats al xifratge de la contrasenya mestra
  - **$2xRdY83VxLXrQBUzXrbyqO** - Són els bits de sal utilitzats pel xifratge
  - **sHLCaqk4HB0iFjBXVjOQDSn2kgbcnca** – És pròpiament la contrasenya xifrada
- ➔ El camp "sal" emmagatzema els bits de sal utilitzats pel xifratge de la contrasenya mestra. Aquests bits de sal estan incorporats en el text xifrat, però s'ha decidit emmagatzemar-los en un camp independent per facilitar la simplicitat de l'aplicació.
- ➔ La clau pública[9] s'emmagatzema sense encriptar (a nivell d'aplicació) i amb el mateix format que l'envia el client. La clau pública està emmagatzemada com una cadena que representa un objecte JSON[10] que el client podrà interpretar en rebre'l.

9 La clau pública i privada que es presenten en el diagrama de registre, en el diagrama relacional de la base de dades i que es descriuen aquí, es van incloure en el disseny i la programació ja que inicialment estava previst utilitzar-les per una funcionalitat de compartició de contrasenyes. Encara que la implementació d'aquesta funcionalitat s'ha descartat, mantenim la generació i emmagatzematge d'aquestes claus de criptografia asimètrica per què estan incorporades al producte resultant que acompanya aquesta memòria i per facilitar una posterior implementació de la compartició de contrasenyes En el darrer capítol es discuteixen les possibles funcionalitats que es podrien afegir i es proposa com s'haurien d'utilitzar aquestes claus per la compartició de contrasenyes.

10 Més endavant veurem com el format d'objecte JSON (*JavaScript Object Notation*) s'ha triat com a format d'intercanvi d'informació entre el client i el servidor.

- La clau privada es desa encriptada. Està encriptada amb xifratge AES amb una clau generada a partir de la contrasenya mestra del client. Es desa com a text un objecte JSON que inclou el text xifrat, el vector d'inicialització i els bits de sal utilitzats per derivar la clau.
- El camp "activat" indica si l'usuari ha activat el seu compte (1) o no (0).
- El camps "tipus" indica el tipus d'usuari. Aquest camp no s'utilitza en l'estat actual del desenvolupament. El seu valor per defecte és 2 que equival al tipus d'usuari normal.

4.5.2.2. Taula "accessos"

La taula "accessos" emmagatzema les dades d'accés als serveis web que els usuaris desen al gestor de contrasenyes i que es reben del client ja encriptades. La informació que el client envia xifrada amb AES s'emmagatzema com una cadena que representa un objecte JSON amb els següents atributs:

a) **ct**: el text xifrat

b) **iv**: el vector d'inicialització amb que s'han xifrat les dades

c) **sal**: els bits de sal utilitzats per derivar la clau a partir de la contrasenya mestra

La taula 2 ens mostra un exemple dels registres d'aquesta taula.

Camp	Valors
idAcces	47
idUsuari	109
nomServei	{"ct":"puwcl36sNRG6KheuxP3QGg==","iv":"eaff33526fb2448bd59d5783caa00197","sal":"21e73bc6317faf49ec8271b2c7908821"}
dadesAcces	{"ct":"mVMK0zf8SBm6Os4llQCkcsYTwPx+QJL12x4lH14uDLt4plbncyrFFqTMsFJcnV82","iv":"f9c975ed005cedac0a3ae69a4b8c32fc","sal":"bd0e30e455b08582887d89d8b1e2559a"}

Taula 2 - Exemple de registre de la taula d'accessos

- El camp "idAcces" és la clau principal de la taula.
- El camp "idUsuari" és una clau forana de la taula "usuaris" i identifica l'usuari que ha desat les dades d'accés del registre.
- El camp "nomServei" és el nom que l'usuari ha introduït per identificar el servei web i s'emmagatzema encriptat amb AES.
- El camp "dadesAcces" és una cadena que representa el nom d'usuari i la contrasenya d'accés al servei web i que s'emmagatzema encriptada amb AES.

4.5.2.3. Taula "llavors"

La taula "llavors" és una taula que guarda informació relevant per l'autenticació dels usuaris. A diferència dels registres de les altres dues taules -que contenen registres més persistents-, els valors de la taula de llavors es generen i actualitzen durant els intents d'autenticació i s'esborren quan l'usuari tanca la sessió. Podríem dir que és una taula auxiliar del sistema d'autenticació i no manté cap registre de l'activitat dels usuaris.

La taula 3 ens mostra un exemple de registre d'aquesta taula.

Camp	Valors
idLlavor	15
idUsuari	109
llavor	Bg7XK9ZuxrzMfku
intents	5
dataIntent	2013-05-14 18:12:39
ipClient	127.0.0.1

Taula 3 - Exemple de registre de la taula de llavors

- El camp "idLlavor" és la clau principal de la taula i identifica de manera única a cada llavor.
- Els valors del camp "llavor" són cadenes aleatòries que genera el servidor per cada intent d'autenticació de l'usuari. La taula de llavors s'actualitza per cada intent d'autenticació de manera que l'usuari sempre s'autentica davant del servidor amb una contrasenya diferent.
- El camp "intents" comptabilitza els intents d'autenticació que fa un usuari. El màxim nombre d'intents són 3. Mentre l'usuari té oberta una sessió al gestor de contrasenyes, aquest camp pren valor 5 (com en el cas de l'exemple).
- El camp "dataIntent" és del tipus timestamp i pren per defecte el valor CURRENT_TIMESTAMP en el moment de crear-se un registre durant l'intent d'autenticació. Les actualitzacions del registre en els posteriors intents comporten l'actualització del camp amb CURRENT_TIMESTAMP. El sistema d'autenticació estarà bloquejat durant 5 minuts des del darrer intent per un usuari que hagi consumit tres intents sense èxit. Aquest camp s'utilitza per saber el moment del darrer intent.
- El camp "ipClient" emmagatzema l'adreça IP des de la que s'ha efectuat l'intent d'autenticació. Aquest camp s'utilitzarà durant la sessió activa per comprovar que les peticions del client provenen de la mateixa adreça IP.

4.5.3. Estructura de les dades encriptades

Com hem vist anteriorment, el gestor de contrasenyes utilitzarà xifratge simètric (o de clau compartida) tant per protegir la contrasenya mestra dels usuaris com per protegir les dades d'accés a serveis web que desen en la nostra aplicació[11]. La raó d'utilitzar xifratge simètric és que, malgrat que la informació s'emmagatzemi en una base de dades en el servidor, en realitat no hi ha cap canvi d'informació entre el client i el servidor. És a dir, l'únic que fa el client és xifrar les dades per què ningú no hi pugui accedir mentre estan desades, i el client és l'únic que ha de desxifrar-les.

I si no hi ha intercanvi d'informació, no hi ha d'haver intercanvi de claus. I si no hi ha intercanvi de claus, no és necessari recòrrer a la criptografia de clau pública. Només introduiríem la criptografia

11 Evidentment, això no inclou el xifratge a nivell de capa de transport que sí utilitza criptografia asimètrica ja que, en aquest cas, el que es protegeix és un intercanvi d'informació entre el client i el servidor.

de clau pública en el supòsit d'implementar la compartició de contrasenyes ja que, en aquest cas, sí hi hauria intercanvi d'informació entre diferents usuaris que implicaria un intercanvi de claus.

En el xifrage de clau compartida, d'acord amb la suposició de Kerckhoffs, tota la seguretat recau en la clau (en aquest cas, en la contrasenya mestra). Més en aquest cas en el que tots els algorismes de xifratge són públics ja que els algorismes que els calculen els pot descarregar del servidor qualsevol potencial client en els arxius que s'executen de la banda del servidor.

Ara bé, per xifrar amb els algorismes que utilitzarem -BlowFish i AES- necessitarem utilitzar algunes dades addicionals que ajuden a fer aquests xifratges més segurs i que no cal que siguin secretes. Donat que l'aplicació de la banda del client no és persistent i, a més, ha de poder accedir-hi des de qualsevol dispositiu connectat a Internet, aquesta informació necessària s'ha d'emmagatzemar a la base de dades per tal que un client correctament autenticat la pugui recuperar en qualsevol moment.

Això condicionarà la manera com hem de desar les dades xifrades. A continuació, veurem com s'emmagatzemen les dades pels dos sistemes de xifratge.

4.5.3.1. Xifratge de la contrasenya amb BCrypt

En el cas del xifratge amb BCrypt s'utilitzen uns bits de sal aleatoris que fan més segur l'encriptament, fent-lo més resistents a atacs de diccionari i de *rainbow tables*. Encara que en el cas de l'encriptació de la contrasenya mestre, el client en cap cas necessita desencriptar-la[12], sí que necessitarà recuperar els bits de sal per poder tornar a encriptar novament la contrasenya mestra per autenticar-se davant del servidor.

La següent figura mostra com els bits de sal s'emmagatzemen a la base de dades juntament amb el text xifrat.

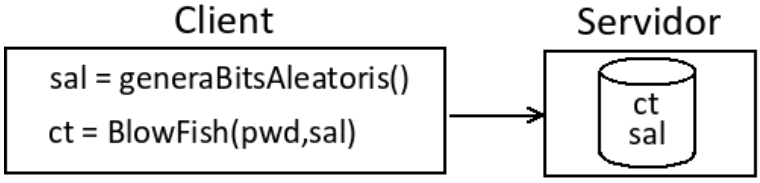

Figura 13 - Xifratge de la contrasenya amb BlowFish

En el moment d'autenticar-se, el client pot fer-ho des de qualsevol dispositiu ja que els bits de sal necessaris per poder fer-ho estan desats a la base de dades. La figura 14 mostra aquest procés.

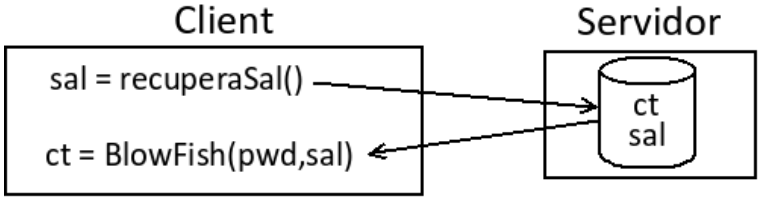

Figura 14 - Recuperar la sal per xifrar amb BlowFish

4.5.3.2. Estructura de les dades xifrades amb AES

12 No tindria cap sentit, ja que precisament la contrasenya mestra és la única cosa que ha de saber en tot moment

En el cas del xifratge AES *(Advanced Encription Standard),* també utilitzarem bits de sal per derivar la clau a partir de la contrasenya mestra utilitzant PBKDF2 (*Password-Based Key Derivation Function 2*). Recordem que no podem utilitzar directament la contrasenya com a clau ja que la clau AES ha de tenir 128 bits. AES és un xifratge de bloc que utilitza vector d'inicialització.

Tant els bits de sal com el vector d'inicialització seran necessaris per tal de desencriptar les dades desades i hauran d'emmagatzemar-se en el servidor. A diferència del cas anterior en què la sal s'emmagatzemava en un camp independentment, les dades xifrades amb AES es desaran en una estructura conjunta[13] que inclourà el text xifrat, els bits de sal i el vector d'inicialització.

El motiu d'emmagatzemar-ho tot junt és que -a diferència de la contrasenya mestra xifrada- aquestes dades el client les desarà i recuperarà com un tot ja que tot el procés d'encriptació i desencriptació tindrà lloc en el client.

Les figures 15 i 16 il·lustren els processos i les estructures de dades resultants.

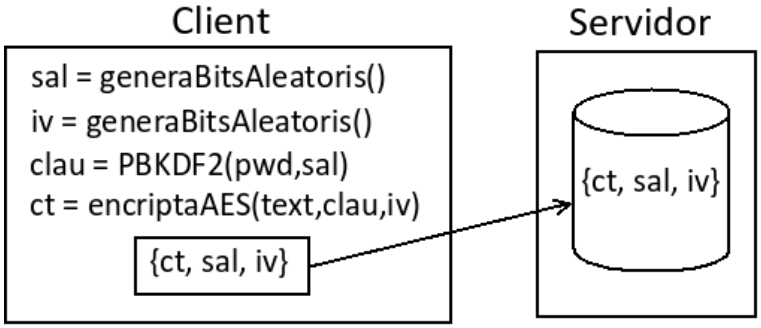

Figura 15 - Dades xifrades amb AES

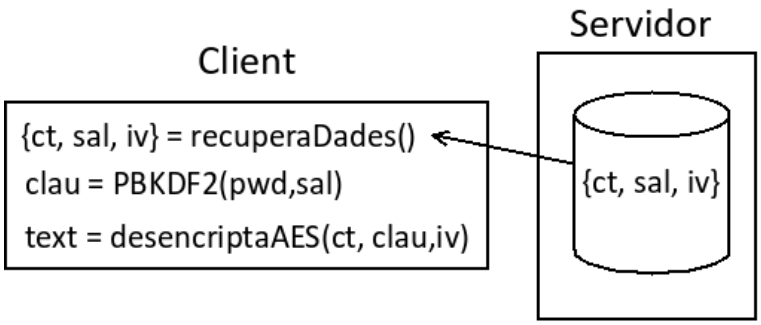

Figura 16 - Dades desxifrades amb AES

13 Més endavant veurem que aquesta estructura és un objecte JSON.

CAPÍTOL 5 - PROGRAMACIÓ

5.1. Tecnologies utilitzades

5.1.1. Servidor web SSL: Apache HTTP Server

Donat que el gestor de contrasenyes s'ofereix com un servei web, cal que l'aplicació estigui instal·lada en un servidor web. El servidor escollit ha estat *Apache HTTP Server* desenvolupat per l'*Apache Software Foundation*. En l'actualitat *Apache* és el servidor de pàgines web més utilitzat [34] i es distribueix d'acord amb l'*Apache License 2.0* que permet usar-lo lliurement respectant el copyright [35]. Encara que inicialment va ser desenvolupat per plataformes Linux/Unix disposa de versions pels sistemes operatius més utilitzat avui en dia [36].

L'hem triat per tractar-se de programari lliure, perquè tot el desenvolupament del projecte s'ha dut a terme en entorns Linux i perquè compleix els requisits que demana la nostra aplicació:

a) inclou el mòdul mod_ssl que suporta *Secure Socket Layer* i *Transport Layer Security*

b) inclou un mòdul de processament per interpretar el llenguatge PHP que és el llenguatge amb que s'ha programat l'aplicació de la banda de l'aplicació

c) suporta l'autenticació amb certificats digitals

5.1.2. Certificats digitals: OpenSSL

Per generar el certificat que autenticarà el servidor davant dels usuaris en connexions SSL/TLS segures, s'ha utilitzat OpenSSL. Un cop generat, el certificat s'ha instal·lat en el servidor i s'ha configurat el servidor perquè l'accés al gestor de contrasenyes només pugui efectuar-se en connexions SSL/TLS. En cas que un client intenti connectar-se en una connexió no segura, el servidor web redirigirà la seva petició al servidor segur.

Els procediments en què s'ha utilitzat OpenSSL no queden reflectits en el producte final i són els següents:

a) Generació d'una clau privada RSA

b) Generació d'un *Certificate Signing Request* (CSR) amb la clau privada anterior

c) L'autoritat de certificació signa el CSR[14]

14 S'ha utilitzat un certificat de l'autoritat de certificació que es va distribuïr als alumnes de l'assignatura *Seguretat de xarxes de computadors* per a la realització de la pràctica uns semestres enrera.

El certificat de l'autoritat de certificació (CA) s'ha instal·lat en els navegadors dels clients que s'han utilitzat per fer les proves. En cas que l'aplicació hagués de passar a una fase de producció, el CSR hauria de signar-se per una CA reconeguda com a tal.

Entre els arxius inclosos en el producte final lliurat s'inclouen els certificats obtinguts.

5.1.3. Llenguatge de servidor: PHP

Com a llenguatge de programació de la banda del servidor s'ha utilitzat PHP (*PHP: Hypertext Preprocessor*), llenguatge àmpliament utilitzat en programació web. Actualment és desenvolupat per *The PHP Group* i es distribueix sota la *PHP License* [37] que permet el seu ús lliure.

Com la majoria de llenguatges de programació, PHP suporta dues característiques que el fan indicat per desenvolupar l'aplicació:

a) suporta la connexió a bases de dades d'una manera senzilla

b) suporta els mètodes de xifratge que és necessari executar de la banda del client: SHA-512 i BlowFish

5.1.4. Llenguatge de client: JavaScript

Com a llenguatge de programació de la banda del client s'ha triat *JavaScript*. És un llenguatge interpretat i està especialment dissenyat per ser executat per navegadors. És un llenguatge de client molt utilitzat per aplicacions web ja que s'integra fàcilment amb els elements que formen part de les pàgines.

Actualment, no hi ha massa alternatives a JavaScript com a llenguatge de la banda de client. En qualsevol cas, s'ha de tenir en compte que, al ser un llenguatge interpretat pel navegador, pot haver-hi lleugeres diferències en les funcionalitats que suporten alguns navegadors.

A nivell funcional, la gran difusió de JavaScript fa que hi hagi una gran quantitat de llibreries que cobreixen les més variades funcions i utilitats. Entre aquestes llibreries n'hi ha que implementen els mètodes criptogràfics requerits per l'implementació del gestor de contrasenyes.

Un aspecte que crec que cal destacar respecte a aquestes llibreries que implementen els mètodes criptogràfics és que utilitzar-les suposa confiar en els seus desenvolupadors encara que sovint aquests són programadors independents que formen part de la "comunitat". La gran complexitat que té la programació d'algorismes criptogràfics fa difícil valorar la ideoneïtat o no de la implementació.

5.1.4.1. Llibreries criptogràfiques JavaScript

Destacarem només les llibreries que s'han utilitzat amb finalitats criptogràfiques:

a) CryptoJS 3.1 [38] – Complerta llibreria d'algorismes criptogràfics implementats en JavaScript de manera clara i amb una documentació molt completa. D'aquesta llibreria s'ha utilitzat els següents algorismes: SHA512, PBKDF2 i AES.

b) jsBCrypt [39] – Implementació de bcrypt en JavaScript.

c) jsbn [40] – Implementació en JavaScript d'algorismes de clau pública. Entre els algorismes que inclou la llibreria, hem utilitzat els algorismes relacionats amb claus RSA[15].

d) passchk_fast [41] – Llibreria que permet valorar la fortalesa d'una contrasenya i calcular-ne l'entropia. Encara que el càlcul de la entropia es fa tenint en compte les característiques de l'idioma anglès, en general funciona prou bé, sempre que tinguem en compte les limitacions d'aquest tipus d'utilitats.

5.1.5. Aplicació web: HTML/AJAX

Lógicament, per les pàgines web s'ha utilitzat HTML (*HyperText Markup Language*) i cal destacar la interacció entre aquest i JavaScript com a llenguatges interpretats pel navegador a l'hora de presentar la informació i les interfícies als usuaris.

Per les característiques del gestor de contrasenyes s'ha utilitzat la tècnica AJAX (*Asynchronous JavaScript and XML*) [42] per simplificar l'estructura i per augmentar la rapidesa de resposta que ja està penalitzada en alguns casos pels temps de processament dels algorismes criptogràfics.

AJAX permet que el client faci peticions HTTP per intercanviar informació amb el servidor en segon pla i sense necessitat de recarregar la pàgina cada cop.

La característica del gestor de contrasenyes que fa especialment indicada aquesta tècnica per implementar-ho té a veure amb que es tracta d'una aplicació on l'intercanvi d'informació entre el client i el servidor és constant. Però, en canvi, els volums d'informació no són molt elevats ja que són, principalment, peticions de consulta o modificació de la base de dades.

Malgrat el seu nom -*Asynchronous JavaScript and XML*- cal destacar que les peticions HTTP (*HTTPRequest*) poden no ser asíncrones i que l'intercanvi d'informació no necessàriament s'ha de fer utilitzant dades XML ja que el client pot rebre la resposta del servidor en text pla. El gestor de contrasenyes fa ús, sobretot, de peticions síncrones i, enlloc d'intercanviar informació en format XML, utilitza el format JSON que veurem a continuació.

5.1.6. Format d'intercanvi d'informació: JSON

JSON (*JavaScript Object Notation*) és un format lleuger d'intercanvi d'informació [43] que s'està imposant com alternativa a XML per la seva senzillesa i la facilitat d'interpretació per part dels llenguatges, especialment JavaScript.

El principal motiu per triar JSON com a format d'intercanvi de dades entre el servidor i el client ha estat que permet representar d'una manera senzilla i representable com a cadena de caràcters les estructures de dades que s'han detallat en la secció 4.4.3.2 per intercanviar amb el servidor dades xifrades al client amb AES.

5.1.7. SGBD: MySQL / phpMyAdmin

Com a servidor de base de dades s'ha triat MySQL que és un sistema de gestió de bases de dades relacionals de codi obert.

15 Recordem que en el procés de registre es generen claus RSA per una possible funcionalitat de compartició de contrasenyes que, finalment, no s'ha incorporat a l'aplicació.

L'administració de la base de dades s'ha fet des de myPHPAdmin (biblio) que és una eina de programari lliure escrita en PHP per gestionar MySQL. Aquesta eina proveu una interfície gràfica per gestionar bases de dades, taules, camps, relacions, usuaris, permisos,... que facilita la interacció amb la base de dades.

CAPÍTOL 6 - PRODUCTE FINAL: INTERFÍCIES D'USUARI

En aquest capítol repassarem el producte final i les interfícies d'usuari. Veurem l'aplicació des del punt de vista de l'usuari.

Quan un usuari accedeix a la pàgina d'inici del gestor de contrasenyes se li donen dues opcions: a) autenticar-se si ja està registrat o b) registrar-se si encara no ho està.

Figura 17 - Pantalla d'inici de l'aplicació

És interessant fixar-se en què l'adreça de la pàgina web és "https://..." per comprovar que la connexió és segura i que el client ha acceptat el certificat amb què el servidor s'ha autenticat.

A continuació es mostraran les interfícies de les diferents funcionalitats i com l'aplicació interactua amb l'usuari mantenint-lo informat en tot moment de cada procés.

6.1. Processos de gestió d'usuaris

6.1.1. Registre d'usuaris

Quan un usuari vol registrar-se en el gestor de contrasenyes, es troba amb un formulari que li demana que introdueixi un nom d'usuari i una contrasenya, i també li demana que confirmi la contrasenya per seguretat. Sota el formulari se li comunica que quan introdueixi la seva contrasenya el sistema l'informarà de les característiques d'aquesta que la fan més o menys segura.

Figura 18 - Formulari de registre

Si l'usuari vol registrar-se sense introduïr un nom d'usuari, el sistema l'informarà que no és possible i esborrarà els camps del formulari.

Figura 19 - Error de registre: falta l'usuari

Si l'usuari introdueix un nom d'usuari que no sigui una adreça de correu electrònica vàlida, el sistema l'informarà i esborrarà els camps del formulari.

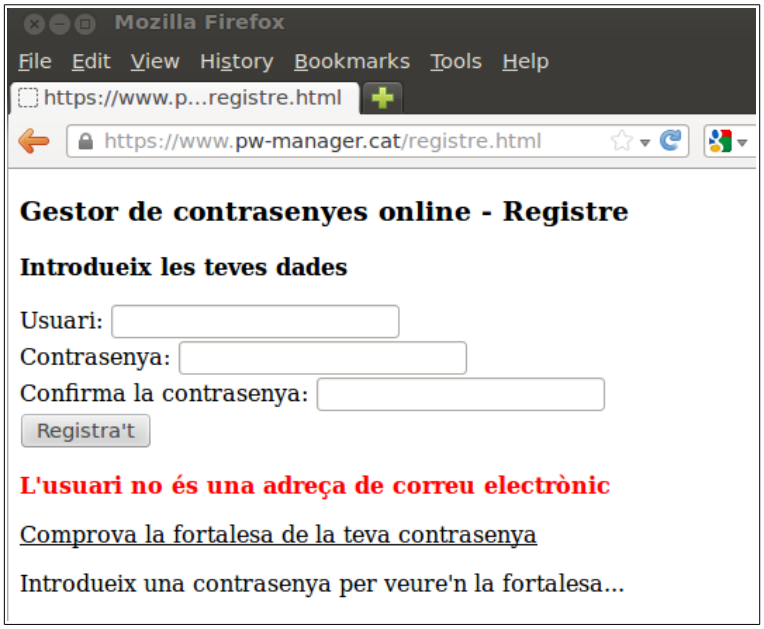

Figura 20 - Error de registre: falta contrasenya

Si les dues contrasenyes introduïdes per l'usuari no coincideixen, el sistema informarà a l'usuari i esborrarà els camps del formulari.

Figura 21 - Error de registre: les contrasenyes no coincideixen

A mesura que l'usuari omple el camp "contrasenya", el sistema mostra les característiques de la contrasenya. Indica el nombre de caràcters, una estimació de la seva fortalesa, una estimació dels bits d'entropia que hi ha entre els caràcters consecutius i el tamany del joc de caràcters utilitzats[16]. A continuació veurem dos exemples de com el sistema valora les contrasenyes a mesura que es van introduïnt.

Figura 22 - Exemple de valoració de contrasenya

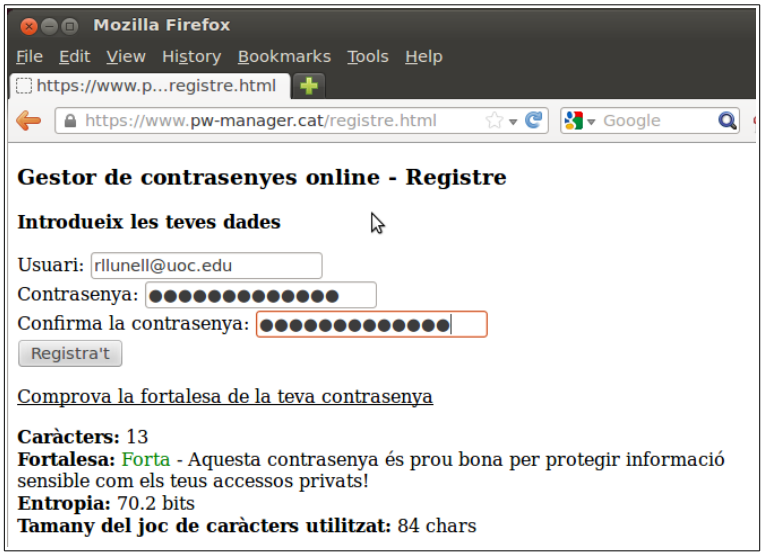

Figura 23 - Exemple de valoració de contrasenya

16 El joc de caràcters serà més gran quan més tipus diferents de símbols inclogui la contrasenya (nombres, minúscules, majúscules, símbols,...)

Si l'usuari vol registrar-se introduïnt una contrasenya que tingui menys de 10 caràcters, el sistema l'informa que la contrasenya ha de tenir, com a mínim, 10 caràcters i borra els camps del formulari.

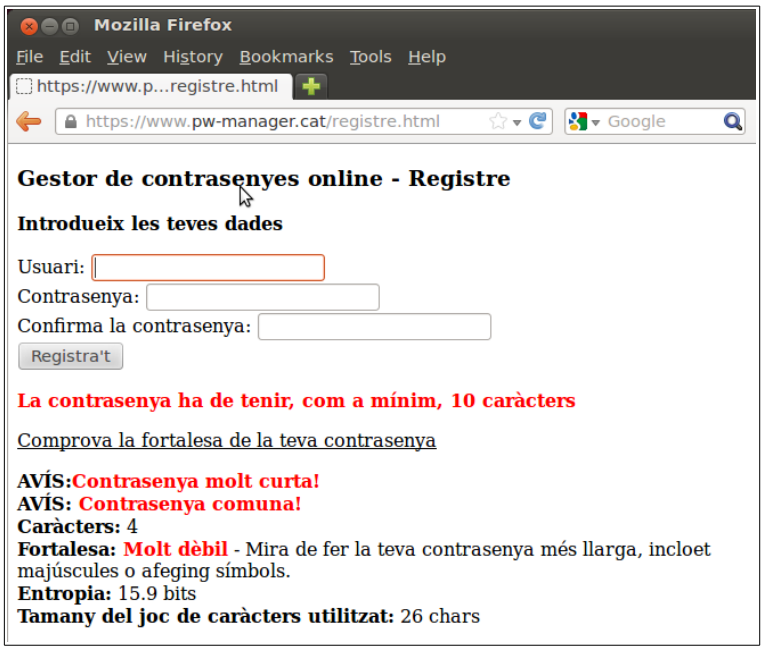

Figura 24 - Error de registre: contrasenya curta

Si l'usuari vol registrar-se amb una contrasenya que tingui menys de 40 bits d'entropia, el sistema l'informarà que no és possible i borrarà els camps del formulari.

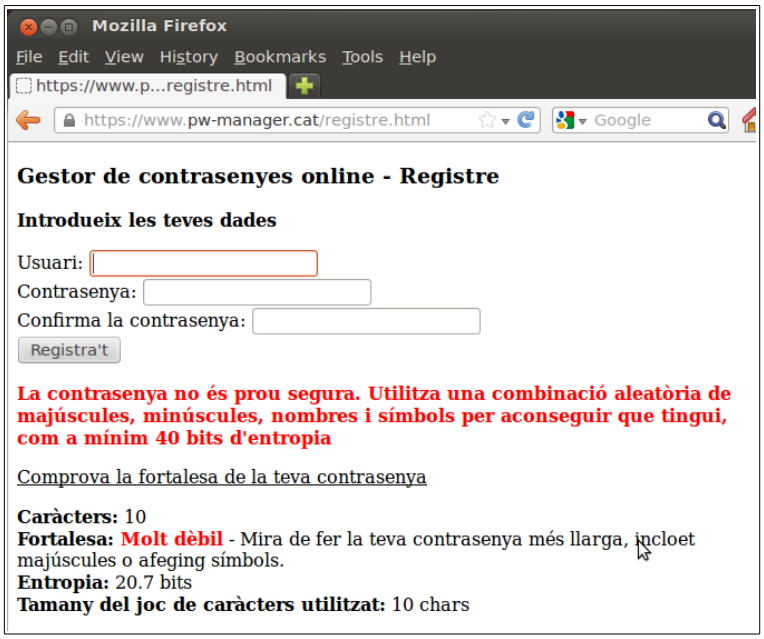

Figura 25 - Error de registre: contrasenya amb poca entropia

Si ja hi ha un usuari registrat amb aquest nom d'usuari, el sistema informa a l'usuari i borra els camps del formulari.

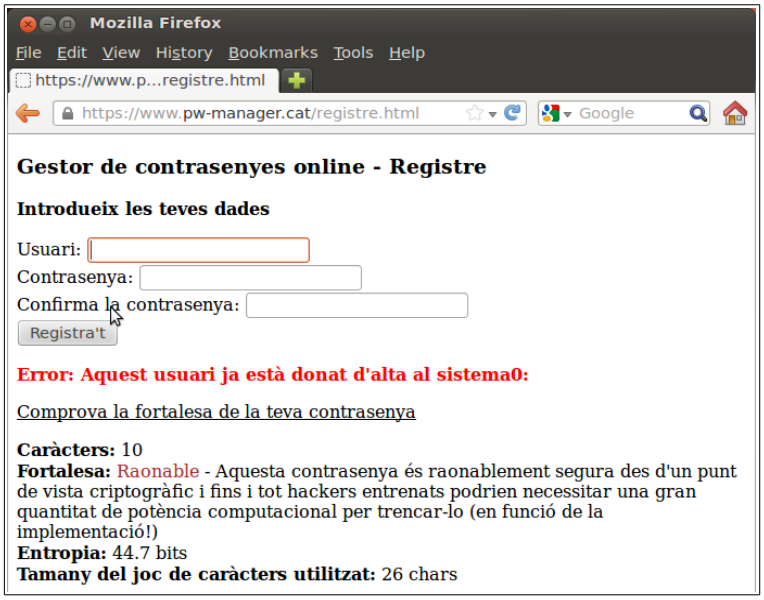

Figura 26 - Error de registre: usuari ja registrat

Si no es dóna cap de les anteriors circumstàncies i tant el nom d'usuari com la contrasenya són correctes, el sistema informa que les dades d'accés a l'aplicació s'han desat correctament i que s'està enviant un correu electrònic de validació.

Figura 27 - Dades guardades en el procés de registre

Per finalitzar el procés, el sistema informa a l'usuari que el procés ha finalitzat satisfactòriament i que ha d'activar el seu compte abans de poder accedir a l'aplicació.

Figura 28 - Registre finalitzat

L'usuari rep un correu electrònic amb un enllaç al que ha d'accedir per activar el seu usuari en l'aplicació.

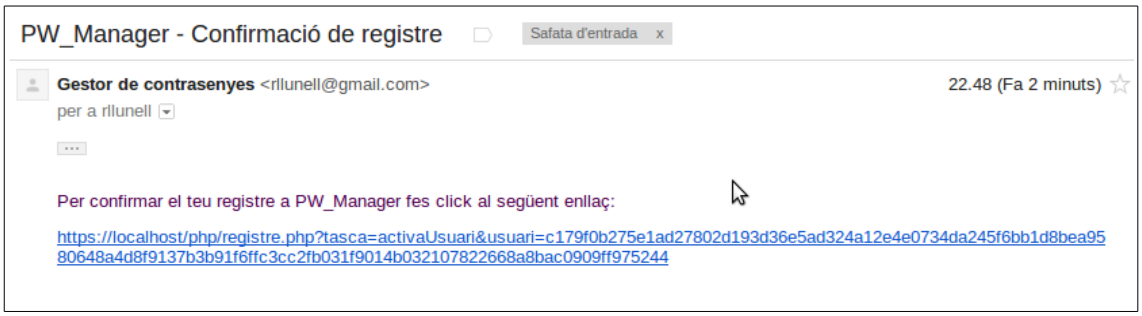

Figura 29 - E-mail d'activació d'usuaris

Un cop activat l'usuari, el procés de registre finalitza completament i l'usuari ja pot autenticar-se per accedir al gestor de contrasenyes.

6.1.2. Autenticació d'usuaris

Quan els usuaris volen accedir a l'aplicació per gestionar les seves contrasenyes, primer han d'autenticar-se per iniciar la sessió. Per fer-ho, accedeixen a la pàgina d'inici de l'aplicació que els dóna la oportunitat d'autenticar-se.

Figura 30 - Formulari d'autenticació al gestor de contrasenyes

Si l'usuari vol autenticar-se sense introduïr un nom d'usuari, el sistema l'informa i retorna al formulari.

Figura 31 - Error de login: falta nom d'usuari

Si el nom d'usuari introduït no és una adreça de correu vàlida, el sistema informa a l'usuari i retorna al formulari.

Figura 32 - Error de login: correu electrònic no vàlid

Si l'usuari vol autenticar-se sense introduïr la seva contrasenya, el sistema l'informa i retorna al formulari.

Figura 33 - Error de login: falta la contrasenya

Si el nom d'usuari introduït en el formulari d'autenticació no està registrat a l'aplicació, el sistema informa a l'usuari i retorna al formulari.

Figura 34 - Error de login: usuari no registrat

Si l'usuari és correcte, però la contrasenya no és correcta, l'aplicació informa a l'usuari i retorna al formulari.

Figura 35 - Error de login: contrasenya incorrecta

Si el nom d'usuari i la contrasenya són correctes i corresponen a un usuari registrat a l'aplicació, però l'usuari encara no ha activat el seu compte. El sistema informa a l'usuari i retorna al formulari.

Figura 36 - Error de login: usuari no activat

Si no es donen cap de les anteriors circumstàncies, l'usuari està registrat i activat i la contrasenya és verificada, l'usuari pot iniciar una sessió en el gestor de contrasenyes i accedeix a la interfície principal de gestió de contrasenyes.

Figura 37 - Sessió iniciada

Com veiem, en la columna de l'esquerra es mostren els serveis pels quals l'usuari ja ha desat les dades d'accés en el gestor de contrasenyes.

6.1.3. Baixa d'usuaris

Si un usuari que té una sessió iniciada vol donar-se de baixa de l'aplicació, ho pot fer en qualsevol moment amb l'opció "Suprimir el meu compte" (veure figura 37). Si ho fa, el sistema l'avisarà que es perdrà tota la seva informació i li demanarà confirmació per continuar.

Figura 38 - Avís de confirmació per suprimir el compte

Si l'usuari confirma l'acció, s'esborraran totes les dades relacionades amb aquest usuari de la base de dades del gestor de contrasenyes. Es tancarà la sessió i es mostrarà el formulari d'autenticació. Si l'usuari vol tornar a accedir a l'aplicació, no podrà fer-ho.

Figura 39 - L'usuari eliminat ja no pot accedir a l'aplicació

6.2. Processos de la gestió de contrasenyes

Un cop l'usuari ha iniciat una sessió a l'aplicació, se li mostra la interfície inicial de l'aplicació mostrada a la figura 37. A partir d'aquesta pantalla, podrà gestionar les seves contrasenyes.

6.2.1. Desar contrasenyes

Si l'usuari selecciona l'opció "Desa un password", el sistema li mostrarà el formulari per introduïr les dades d'accés a un servei web per desar-les en el gestor de contrasenyes.

Figura 40 - Formulari d'introducció d'unes dades d'accés

El formulari demana a l'usuari el nom del servei, el nom d'usuari, la contrasenya i una confirmació de la contrasenya. Les opcions que té el client són desar les dades d'accés o cancel·lar

l'operació[17]. De manera similar al procés de registre, abans que l'usuari pugui desar les dades, el sistema comprovarà que les dades siguin correctes.

Si l'usuari vol guardar les dades sense haver introduït un nom pel servei, el sistema l'informarà i tornarà al formulari.

Figura 41 - Error al desar l'accés – Falta el nom del servei

Si l'usuari vol desar les dades sense haver introduït un nom d'usuari per aquestes dades d'accés, el sistema l'informarà i retornarà al formulari.

Figura 42 - Error al desar l'accés – Falta el nom d'usuari

17 En la figura 40 es mostra també l'opció "Comparteix", però només a nivell d'interfície gràfica ja que la funcionalitat no està implementada.

Si l'usuari vol desar les dades sense haver introduït una contrasenya, el sistema l'informarà i retornarà al formulari.

Figura 43 - Error al desar l'accés – Falta la contrasenya

Si la contrasenya i la seva confirmació no són iguals, en seleccionar "Desa" el sistema informarà a l'usuari i retornarà al formulari.

Figura 44 - Error al desar l'accés – Les contrasenyes no coincideixen

Si tots els camps són correctes, però el sistema considera que la contrasenya és massa senzilla i que pot ser comuna, es mostra un avís a l'usuari i se li demana confirmació per continuar. A diferència del procés de registre, només s'informa a l'usuari però no se li impedeix continuar amb el procés.

Figura 45 - Avís de contrasenya comuna

Si la contrasenya introduïda per l'usuari té menys de 40 bits d'entropia, es mostra un avís a l'usuari i se li demana confirmació per continuar. Igual que en el cas anterior, només s'informa a l'usuari però no se l'impedeix continuar amb el procés.

Figura 46 - Avís de poca entropia

Si totes les dades són correctes, les dades d'accés del servei són desades en el gestor de contrasenyes. La llista de serveis s'actualitza amb el nou servei desat.

Figura 47 - Dades d'accés desades i llista actualitzada

6.2.2. Recuperar contrasenyes

Recuperar una contrasenya és molt senzill per a l'usuari. Només cal que, a la llista de serveis de l'esquerra, faci clic al nom del servei pel que vol recuperar les dades.

Figura 48 - Recupera unes dades d'accés

6.2.3. Modificar contrasenyes

Un cop hem recuperat unes dades d'accés d'un servei com es mostra a la figura 48, una de les opcions que tenim és "Modifica" que ens permet editar els camps que formen aquestes dades d'accés. Si seleccionem aquesta opció, se'ns mostrarà el mateix formulari que s'utilitzava per desar unes noves dades d'accés, però amb la diferència que ara el formulari conté les dades actuals.

Figura 49 - Modificació d'unes dades d'accés

Les restriccions són les mateixes que hem vist en el cas del procés de desar unes noves dades d'accés. A la figura 50 podem veure com, no només la contrasenya, sinó tots els camps poden editar-se.

Figura 50 - Modificació de les dades d'accés d'un servei

Un cop les dades han estat correctament desades, la llista de serveis s'actualitza i si recuperem les dades del servei que acabem de modificar, veiem que ja es mostren modificades.

Figura 51 - Dades d'accés modificades

6.2.4. Eliminar contrasenyes

Si un cop hem recuperat unes dades d'accés (veure figura 48) seleccionem l'opció "Eliminar", l'aplicació ens avisarà que es perdran les dades i ens demanarà que confirmem si volem continuar.

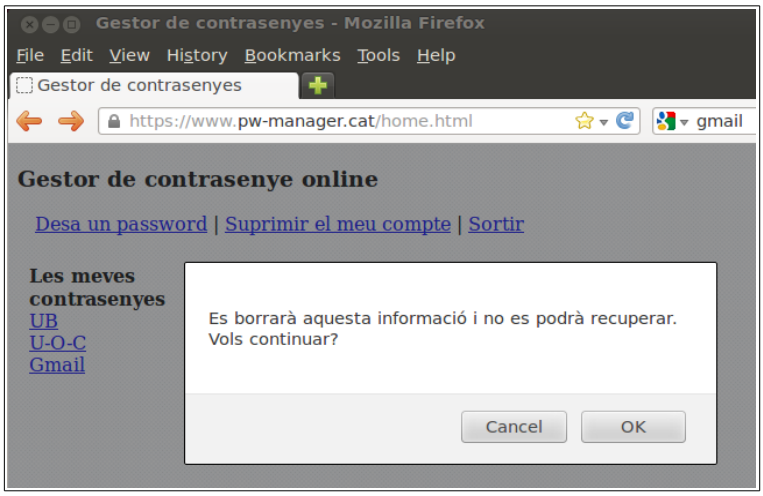

Figura 52 - Avís abans d'eliminar unes dades d'accés

Si l'usuari confirma l'eliminació, les dades d'accés s'eliminaran i s'actualitzarà la llista de serveis.

Figura 53 - Dades d'accés eliminades i llista actualitzada

CAPÍTOL 7 - CONCLUSIONS

7.1. Valoració global del resultat obtingut

La valoració final del TFC i de la implementació d'un gestor de contrasenyes *online* és moderadament satisfactòria. D'una banda, s'ha assolit l'objectiu de lliurar un producte final plenament operatiu i funcional. Però d'altra banda, considero que és un desenvolupament amb molts punts de millora possible. Tan a nivell funcional, com a nivell de disseny de les interfícies d'usuari que són especialment pobres.

En qualsevol cas, valoro molt positivament el fet d'haver pogut desenvolupar tot un projecte des de les primeres etapes de definició prèvia fins al lliurament del producte final. Ha estat una oportunitat per posar en pràctica molts coneixemens adquirits en el decurs dels estudis d'Enginyeria Tècnica en Informàtica de Sistemes. Una oportunitat que m'ha servit especialment per valorar la importància de les fases d'anàlisi i disseny i, en general, de la importància de seguir un procediment definit com a millor estratègia per assolir l'objectiu.

Encara que, per la seva naturalesa didàctica, aquest projecte no ha de tenir continuïtat, al finalitzar el mateix es plantegen dubtes i qüestions de futur que miraré de resumir en els següents apartats.

7.2. Possibles noves funcionalitats

És evident que el gestor de contrasenyes que s'ha implementat es limita a cobrir les necessitats mínimes i bàsiques per poder-se considerar un gestor de contrasenyes. Hi ha moltes funcionalitats que podrien afegir-se si hagués de passar a una fase de producció.

- La **generació de contrasenyes** que siguin prou fortes i segures per proposar-les als usuaris crec que és una de les mancances més importants de l'aplicació. Fins i tot, seria interessant plantejar-se que els usuaris tinguessin l'opció d'acceptar-la directament. És a dir, que el sistema s'encarregués de generar la contrasenya i de desar-la sense que l'usuari li calgués ni tan sols conèixer-la.
- L'**accés directe als serveis web** des del gestor de contrasenyes. Un cop més, sense que l'usuari hagi de conèixer ni teclejar les dades d'accés. Per això, haurien de desar-se les URL del serveis i desenvolupar-se la complimentació automàtica de formularis.
- La **compartició de contrasenyes** és un dels punts de millora possible i es va plantejar en les primeres etapes d'anàlisi. Posteriorment, es va descartar per manca de temps per dedicar al

projecte. Aquesta funcionalitat suposaria implementar algorismes criptogràfics de clau pública[18].

7.3. Manteniment del programari i adaptació a nous requisits

Com tot programari, si el gestor de contrasenyes hagués de passar a estar operatiu, hauria d'haver-hi un manteniment. Tant del propi programari, com de l'entorn (servidors, tallafocs, IDS,...) que és tant o més important a nivell de seguretat.

Però en aquest punt em vull centrar en dos aspectes específics dels gestors de contrasenyes: preveure la finalització del servei i l'adaptació a nous requisits de seguretat.

7.3.1. Finalització del servei

Crec que en el decurs de la memòria, ha quedat ja clar que els gestors de contrasenyes són elements crítics per tal que els seus usuaris puguin desenvolupar les seves tasques quotidianes i professionals amb normalitat. És per això que és tan important garantir-ne la seguretat i la disponibilitat.

Però aquesta importància fa que s'hagi de preveure des de la seva posada en marxa què passaria si els administradors del gestor de contrasenyes, en un moment donat, no poden mantenir l'aplicació i no poden garantir-ne la continuïtat. Arriba aquest moment, s'ha de definir un pla d'actuació que garanteixi que cap usuari perd cap informació o accés relevant.

Aquest punt considero que és especialment important tenir-lo en compte quan no es té absolutament cap informació de l'usuari. No podem contactar amb els usuaris per donar-los un marge de temps per recuperar les seves dades o per facilitar-los la migració a un altre servei similar.

No he trobat informació sobre aquest punt en cap dels gestors de contrasenyes que funcionen actualment. Tampoc no tinc clar com s'hauria de preparar aquest pla. Evidentment, no és un problema fàcil de solucionar, si és que la solució existeix.

7.3.2. Adaptació a nous requisits de seguretat

El concepte de seguretat en criptografia és variable amb el temps. L'avenç de les tècniques de criptoanàlisi, l'augment exponencial de la capacitat computacional dels ordinadors, una organització cada cop més "professional" i coordinada dels atacs contra informació sensible dels usuaris són només alguns dels elements que mostren que el gestor de contrasenyes haurà d'estar avaluant constantment la seguretat que ofereixen els algorismes que implementa.

Però, un cop més, això no és tan fàcil. El gestor de contrasenyes només emmagatzema informació xifrada a la que no tenim accés i que només pot ser desencriptada amb el concurs de l'usuari i la seva contrasenya mestre.

18 De fet, aquests algorismes ja s'han començat a implementar i tots els usuaris disposen, després del seu registre a l'aplicació, d'un parell de claus RSA. El fet que es contemples la compartició de contrasenyes en les fases inicials del projecte també explica per què el nom d'usuari per autenticar-se en l'aplicació havia de ser una adreça de correu electrònic validada. Un usuari indicaria amb quins usuaris volia compartir una contrasenya indicant-ne el correu electrònic.

Si, per exemple, d'aquí a uns anys es desenvolupessin atacs de criptoanàlisi que poguessin trencar amb èxit l'algorisme AES, nosaltres ens plantejaríem actualitzar el nostre algorisme de xifratge i utilitzar-ne un altre que oferís la robustesa necessària. Però, què fer amb tota la informació desada fins aquell moment? Necessàriament hauria de romandre encriptada amb AES ja que només pot desencriptar-se en el client utilitzant la contrasenya mestra de l'usuari.

L'actualització no es podria fer en una sola fase. S'haurien de programar *scripts* de la banda del client que desencriptessin les dades amb AES i, després, les encriptessin amb el nou algorisme. Però això només es podria fer quan els usuaris accedissin a l'aplicació. I hauríem de mantenir els dos algorismes i el procés de migració fins que tots els usuaris l'haguessin complertat. I això potser no acabaria de passar mai.

7.4. El futur dels gestors de contrasenyes *online*

Els gestors de contrasenyes *online* ofereixen una gran ajuda a tots els usuaris d'Internet que accedeixen cada cop a més serveis web i per aspectes cada cop més rellevants (banca electrònica, comerç electrònic, sanitat *online*,...). A més, suposen un avantatge important respecte als gestors de contrasenyes locals desats als ordinadors dels usuaris en un món on la mobilitat i la integració de dispositius és cada cop més important.

Vol dir això que els gestors de contrasenyes jugaran un paper important en el futur de la seguretat d'Internet? En la meva opinió, no. Veiem per què.

- Els gestors de contrasenyes no són una alternativa a un escenari que cada cop demana més contrasenyes als usuaris. Només són una manera de gestionar aquest escenari. I no crec que aquest escenari en què un usuari pot arribar a tenir desenes -potser centenars en el futur- de contrasenyes diferents sigui sostenible.
- Hi ha alternatives tecnològiques que suposarien una estandardització dels processos d'autenticació amb tots els avantatges que això suposa. N'hi ha que utilitzen contrasenyes (tècniques de *single sign-on*) i d'altres que utilitzarien la infraestructura de clau pública (no crec que sigui agosarat pensar en una mena de DNI electrònic a nivell global). Aquestes tecnologies tenen l'avantatge que suposa l'estandardització.
- En darrera instància, els mètodes d'autenticació depenen dels desenvolupadors dels diferents serveis. Si el desenvolupament dels mètodes que hem vist a l'apartat anterior ofereixen als programadors mètodes fiables i reutilitzables per autenticar als usuaris és probable que els acabin utilitzant.
- Ni tal sols actualment, quan els usuaris tenen serioses dificultats per gestionar les seves contrasenyes, es pot dir que els gestors de contrasenyes *online* siguin molt populars. En part, pel desconeixement del gran públic, però també perquè hi ha un problema de confiança.

En definitiva, estaria d'acord amb el final de la següent frase que es pot llegir en l'apartat de la Wikipèdia en anglès dedicat als gestors de contrasenyes: "L'ús d'un gestor de contrasenyes basat en la web és una alternativa a les tècniques de *single sign-on* com OpenID o Windows Life ID, o poden servir com un recurs provisional pendent de l'adopció d'un mètode millor"[44]

7.5. Confiança *vs* seguretat

Per acabar aquesta memòria voldria fer una reflexió sobre la relació entre dos conceptes que sovint estan relacionats però que no són el mateix: la seguretat i la confiança.

En el marc de la criptografia i de les tecnologies de la informació, la seguretat és un concepte objectiu i, fins i tot, quantificable. Pel contrari, la confiança és un factor subjectiu que depèn de la percepció de les persones. Els dos termes estan relacionats ja que, en general, la seguretat d'un sistema acostuma a ser directament proporcional a la confiança que genera.

Un dels punts centrals del projecte proposat com a TFC era la distinció entre la part de l'aplicació que s'executava en el client i la que s'executava en el servidor. D'aquesta manera, simplificant-ho, es considerava que la informació en clar estaria segura si no arribava mai en clar al servidor i només era gestionada de la banda del client.

Però, si ho mirem des del punt de vista de l'usuari, té sentit aquesta diferenciació? En última instància els arxius que s'executen a la banda del client es descarreguen del servidor. Si no confiem en el servidor, per què executar una aplicació que ell ens envia? És cert que la major part de l'aplicació s'executa en el client, però des del punt de vista de l'usuari quina diferència hi hauria si s'executés de la banda del servidor?

En el cas del gestor de contrasenyes la informació en clar s'encripta en el client i això la fa més segura, però ningú no repassa totes les funcions JavaScript per assegurar-se que l'aplicació no ens estigui enganyant. La seguretat de l'aplicació, com s'ha dissenyat, la fa més segura. Però aquest és el nostre punt de vista com a programadors. Des del punt de vista de l'usuari, en última instància el paràmetre clau és la confiança. Utilitzarà el gestor de contrasenyes si confia que aquest és segur, independentment que ho sigui o no.

BIBLIOGRAFIA

1: Michael Pietroforte, *Saved Internet Explorer Passwords*, 2010, http://4sysops.com/archives/saved-internet-explorer-passwords/

2: securityxploded.com, *Exposing de Password Secrets of Intenet Explorer*, , http://securityxploded.com/iepasswordsecrets.php

3: Google Inc., *Manage your website passwords*, 2012, https://support.google.com/chrome/answer/95606?hl=en

4: Jason Faulkner, *How Secure are Your Saved Chrome Browser Passwords?*, 2011, http://www.howtogeek.com/70146/how-secure-are-your-saved-chrome-browser-passwords/

5: Mozilla Support, *Password manager - Remember, delete and change passwords in Firefox*, [Consultada juny 2013], http://support.mozilla.org/ca/kb/password-manager-remember-delete-change-passwords

6: Wikipedia, *Gestor de contraseñas*, [Actualitzada març 2013], http://es.wikipedia.org/wiki/Gestor_de_contrase%C3%B1as

7: Wikipedia, *Password Manager*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Password_manager

8: Oficina de seguridad del internauta (INTECO), *Contraseñas seguras*, [Consultada juny 2013], http://www.osi.es/es/protegete/protegete-en-internet/contrase%C3%B1as-seguras

9: Tim Fisher (PC Support), *Password Manager*, [Consultada juny 2013], http://pcsupport.about.com/od/termsp/g/password-manager.htm

10: Karen Scarfone i Murugiah Souppaya, *Guide to Enterprise Password Management (Draft)*, abril, 2009, National Institute of Standards and Technology, http://csrc.nist.gov/publications/drafts/800-118/draft-sp800-118.pdf

11: Paolo Gasti i Kasper B. Rasmussen, *On The Security of Password Manager Database Formats*, 2012, Computer Science Department, University of California, http://link.springer.com/content/pdf/10.1007/978-3-642-33167-1_44.pdf

12: Kaspersky Lab, *What is Password Manager and how to create password*, [Consultat juny 2013], Online course, http://utils.kaspersky.com/special/pure_2/38_pure_password_manager_en.pdf

13: Melanie Pinola, *LastPass Users May Have to Change the Last Password They'll Ever Need*, maig 2011, http://lifehacker.com/5798874/lastpass-users-may-have-to-change-the-last-password-theyll-ever-need

14: Melanie Pinola, *Which Password Manager is The Most Secure*, setembre 2012, http://lifehacker.com/5944969/which-password-manager-is-the-most-secure

15: Susan Brudno, *Passwords, A Tangled Web*, gener 2013, Information Today,

16: Cory Janssen, *Brute Force Attack*, [Consultat juny 2013], http://www.techopedia.com/definition/18091/brute-force-attack

17: Benny Pinkas i Tomas Sander, *Securing Passwords Against Dictionary Attacks*, 2002, Proceedings of the 9th ACM Conference on Computer and Communications Security, http://www.pinkas.net/PAPERS/pwdweb.pdf

18: Margaret Rouse, *Definition: Dicctionari attack*, 2005, http://searchsecurity.techtarget.com/definition/dictionary-attack

19: Wikipedia, *Rainbow table*, [Actualitzat maig 2013], http://en.wikipedia.org/wiki/Rainbow_table

20: Wikipedia, *Man-in-the-middle*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Man-in-the-middle_attack

21: Wikipedia, *Blowfish (cipher)*, [Actualitzada maig 2013], http://en.wikipedia.org/wiki/Blowfish_%28cipher%29

22: Wikipedia, *One-time password*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/One-time_password

23: Wikipedia, *Challenge-response authenticacion*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Challenge%E2%80%93response_authentication

24: Wikipedia, *Advanced Encryption Standard*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

25: Wikipedia, *PBKDF2*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Pbkdf2

26: EverPassword, *Blowfish Encryptor*, [Consultada juny 2013], http://www.everpassword.com/blowfish-encryptor

27: Wikipedia, *Bcrypt*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Bcrypt

28: Johnny Selley i Philip Stolarczyk, *bcrypt*, [Consultada juny 2013], http://bcrypt.sourceforge.net/

29: Coda Hale, *How To Safely Store A Password*, gener 2010, http://codahale.com/how-to-safely-store-a-password/

30: Niels Provos and David Mazières, *A Future-Adaptable Password Scheme*, 1999, USENIX Annual Technical Conference, http://static.usenix.org/events/usenix99/provos.html

31: Thomas Pornin, *Forum IT Security: Resposta a Do any security experts recommend bcrypt for password storage?*, agost 2011, http://security.stackexchange.com/questions/4781/do-any-security-experts-recommend-bcrypt-for-password-storage

32: Colin Percival, *The scrypt key derivation function*, [Consultada juny 2013], http://www.tarsnap.com/scrypt.html

33: cheongwy [usuari GitHub], *node-scrypt-js*, [Consultada juny 2013], https://github.com/cheongwy/node-scrypt-js

34: Netcraft, *January 2013 Web Server Survey*, gener 2013, http://news.netcraft.com/archives/2013/01/07/january-2013-web-server-survey-2.html

35: Apache Software Foundation, *Apache License, Version 2.0*, gener 2004, http://www.apache.org/licenses/LICENSE-2.0.html

36: Wikipedia, *Apache HTTP Server*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Apache_HTTP_Server

37: The PHP Group, *PHP Licensing*, [Actualitzada juny 2013], http://www.php.net/license/

38: Jeff.Mott.OR [Usuari de code.google.com], *crypto-js - JavaScript implementations of standard and secure cryptographic algorithms*, [Actualitzada 2013], http://code.google.com/p/crypto-js/

39: nevins.bartolomeo@gmail.com [Usuari de code.google.com], *javascript-bcrypt*, [Consultada juny 2013], http://code.google.com/p/javascript-bcrypt/

40: Tom Wu, *RSA and ECC in JavaScript*, [Actualitzada setembre 2009], http://www-cs-students.stanford.edu/~tjw/jsbn/

41: Scott Donnelly, *passchk_fast - A JavaScript strength utility based on passchk*, [Consultada juny 2013], http://scott.donnel.ly/passchk_fast-a-javascript-password-strength-utility-based-on-passchk/

42: Wikipedia, *Ajax (programming)*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Ajax_%28programming%29

43: www.json.org, *Introducing JSON*, [Consulada juny 2013], http://www.json.org/

44: Wikipedia, *Online password manager*, [Actualitzada juny 2013], http://en.wikipedia.org/wiki/Password_manager#Online_password_manager