
This is the **published version** of the bachelor thesis:

Peix Iglesias, Jan. *Disseny i implementació d'un sistema d'emmagatzematge de fitxers amb xifrat d'extrem a extrem*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334440>

under the terms of the  license.

Disseny i implementació d'un sistema d'emmagatzematge de fitxers amb xifrat d'extrem a extrem

Jan Peix

29 de juny de 2026

Resum– Aquest treball presenta el disseny i implementació d'una funcionalitat de *Personal Vault* basada en xifrat de punta a punta (E2EE) integrada en un sistema de gestió de fitxers *self-hosted*. L'objectiu és garantir la confidencialitat de les dades mitjançant un model *zero-knowledge*, on el servidor no té accés al contingut dels fitxers ni a les claus criptogràfiques dels usuaris. El sistema combina criptografia simètrica i asimètrica amb gestió de claus al costat del client, realitzant totes les operacions criptogràfiques de forma transparent. Addicionalment, s'ha integrat TLS per assegurar les comunicacions entre client i servidor. El treball valida la viabilitat d'integrar E2EE en sistemes web mantenint funcionalitat, seguretat i rendiment.

Paraules clau– E2EE, TLS, *zero-knowledge*, criptografia, emmagatzematge segur, seguretat web.

Abstract– This work presents the design and implementation of a *Personal Vault* feature based on end-to-end encryption (E2EE), integrated into a self-hosted file management system. The objective is to ensure data confidentiality through a zero-knowledge model, where the server has no access to file contents or users' cryptographic keys. The system combines symmetric and asymmetric cryptography with client-side key management, performing all cryptographic operations transparently. Additionally, TLS has been integrated to secure client-server communications. The work demonstrates the feasibility of integrating E2EE into web-based systems while maintaining functionality, security, and performance.

Keywords: E2EE, TLS, zero-knowledge, cryptography, secure storage, web security.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

EN ELS DARRERS ANYS, els serveis d'emmagatzematge al núvol s'han convertit en eines essencials tant en entorns personals com professionals, permetent l'emmagatzematge, sincronització i compartició eficient d'informació. Plataformes com Google Drive, Dropbox o OneDrive faciliten l'accés remot i la col·laboració, però impliquen que les dades dels usuaris resideixen en infraestructures de tercers, amb les conseqüents implicacions en termes de dependència i privacitat.

Tot i la implementació generalitzada de mecanismes de xifrat en trànsit i en repòs, en la majoria de casos el pro-

veïdor manté el control de les claus criptogràfiques. Aquesta arquitectura pot permetre l'accés al contingut dels fitxers en determinades circumstàncies, com ara incidents de seguretat o requeriments legals, fet que posa de manifest la necessitat de models de confiança mínima.

El xifrat d'extrem a extrem (E2EE) aborda aquesta limitació garantint que únicament l'usuari final tingui accés al contingut en clar de les seves dades. En aquest model, les operacions de xifrat i desxifrat es realitzen íntegrament al costat del client, mentre que el servidor actua exclusivament com a repositori de dades xifrades dins d'una arquitectura *zero-knowledge*, eliminant qualsevol capacitat d'accés al contingut original.

A partir d'aquest paradigma, es proposa l'evolució d'un sistema existent de gestió de fitxers cap a un Personal Vault basat en E2EE i criptografia híbrida, amb comunicacions protegides mitjançant el protocol *Transport Layer Security* (TLS). Es presenta el disseny i la implementació de la solu-

- E-mail de contacte: 1671277@uab.cat
- Menció: Tecnologies de la Informació
- Treball tutoritzat per: Rubén Martínez Vidal (DEIC)
- Curs 2025/26

ció, així com una avaluació del rendiment i de les propietats de seguretat del sistema.

2 ESTAT DE L'ART

2.1 Introducció al núvol i sistemes d'emmagatzematge

El *cloud computing* ha modificat profundament la gestió de dades, permetent l'accés remot a recursos d'emmagatzematge i processament a través d'Internet [1]. En aquest context, serveis com Google Drive, Dropbox i OneDrive s'han establert com a solucions àmpliament utilitzades per a l'emmagatzematge i la sincronització de fitxers entre dispositius.

Aquests sistemes es basen en una arquitectura centralitzada, on l'usuari delega la custòdia de les seves dades a un proveïdor extern, que gestiona la infraestructura i els serveis associats. Això aporta avantatges clars en termes de disponibilitat, escalabilitat i facilitat d'ús, però introdueix una dependència directa del proveïdor en aspectes relacionats amb la privacitat i el control de la informació.

2.2 Seguretat en serveis d'emmagatzematge al núvol

Els principals proveïdors implementen mecanismes de seguretat estàndard com el xifrat en trànsit mitjançant TLS i el xifrat en repòs amb algoritmes simètrics com AES. Aquestes tècniques protegeixen les dades davant accessos externs no autoritzats i atacs durant la transmissió o l'emmagatzematge físic.

No obstant això, en la majoria de casos el proveïdor gestiona les claus criptogràfiques, fet que implica que el sistema no és completament opac per a l'entitat que el proporciona. Això manté un model de confiança centralitzada, on la confidencialitat depèn tant de la criptografia com de la gestió interna del servei.

2.3 E2EE i models zero-knowledge

El model E2EE representa un canvi de paradigma respecte als sistemes tradicionals d'emmagatzematge al núvol. En aquest enfocament, el xifrat i el desxifrat es realitzen exclusivament al dispositiu del client, de manera que el servidor només emmagatzema dades xifrades sense accés al contingut en clar ni a les claus necessàries per interpretar-lo.

Aquest model es coneix com a arquitectura *zero-knowledge*, ja que el proveïdor no disposa de la informació necessària per accedir al contingut emmagatzemat. Això trasllada completament el control criptogràfic a l'usuari i elimina la dependència del servidor en termes de confidencialitat.

Un dels exemples més representatius d'aquest model és MEGA.nz. La plataforma implementa un esquema de xifrat del costat del client en què cada fitxer es xifra localment mitjançant claus simètriques generades aleatòriament abans de ser enviades al servidor. Aquestes claus es protegeixen mitjançant mecanismes de derivació de claus basats en la contrasenya de l'usuari, garantint que el proveïdor no tingui accés al material criptogràfic necessari per al desxifrat [2].

Altres plataformes d'emmagatzematge al núvol, com Proton Drive, també adopten arquitectures *zero-knowledge*. A diferència dels models tradicionals, Proton Drive aplica xifrat E2EE no només al contingut dels fitxers, sinó també als noms de fitxers i carpetes, de manera que el proveïdor no pot conèixer la informació emmagatzemada ni la seva estructura organitzativa. Addicionalment, el sistema incorpora signatures digitals i mecanismes de verificació d'integritat que permeten detectar modificacions no autoritzades del contingut o de la jerarquia de directoris [3].

El model *zero-knowledge* també s'ha aplicat amb èxit en altres tipus de serveis. Un exemple destacat és Bitwarden, un gestor de contrasenyes de codi obert que utilitza E2EE i una arquitectura *zero-knowledge* per protegir les credencials dels usuaris. Segons la seva documentació de seguretat [4], totes les claus criptogràfiques es generen i gestionen localment al dispositiu del client, i el proveïdor no té accés ni a les claus ni a les dades desxifrades. A més, Bitwarden complementa aquest model amb auditories externes periòdiques i la publicació oberta del seu codi font.

Malgrat els avantatges que ofereix en termes de confidencialitat i control per part de l'usuari, el model *zero-knowledge* trasllada tota la responsabilitat de la gestió criptogràfica al client. En conseqüència, la pèrdua de les credencials o dels mecanismes de recuperació pot provocar la pèrdua de l'accés a les dades emmagatzemades.

3 SISTEMA BASE I PUNT DE PARTIDA

Aquest treball parteix d'un sistema web de gestió de fitxers desenvolupat prèviament [5], dissenyat com una aplicació de gestió de fitxers *self-hosted*, similar en funcionalitats a serveis com Google Drive. El sistema segueix una arquitectura client-servidor desplegada mitjançant Docker i Docker Compose, amb un front-end desenvolupat amb Svelte i un back-end basat en FastAPI. Les metadades dels fitxers i carpetes s'emmagatzemen a PostgreSQL, mentre que el contingut dels fitxers es conserva al sistema de fitxers del servidor. La interfície principal del sistema es mostra a l'Annex A.1.

Tot i disposar de mecanismes d'autenticació i control d'accés, el sistema presenta diverses limitacions en termes de seguretat. La comunicació entre client i servidor es realitza mitjançant HTTP, sense protecció criptogràfica durant la transmissió, i els fitxers s'emmagatzemen al servidor en format no xifrat, mantenint la capacitat d'accés al seu contingut.

Aquestes limitacions constitueixen el punt de partida del present treball. L'objectiu és ampliar el sistema mitjançant la incorporació de comunicacions segures basades en TLS i un mecanisme opcional E2EE, que permet combinar l'emmagatzematge tradicional amb un model orientat a la privacitat basat en principis *zero-knowledge*.

4 OBJECTIUS

4.1 Objectiu general

L'objectiu principal del treball és dissenyar, implementar i avaluar una funcionalitat d'emmagatzematge segur basada en E2EE sobre un sistema *self-hosted* de gestió de fitxers,

mantenint la compatibilitat amb el model tradicional d'emmagatzematge i habilitant un mode opcional de protecció avançada de dades.

4.2 Objectius específics

- Implementar una funcionalitat *Personal Vault* basada en xifratge del costat del client.
- Protegir el contingut dels fitxers i metadades sensibles abans de la seva transmissió al servidor.
- Dissenyar un esquema de gestió de claus basat en criptografia híbrida.
- Integrar comunicacions segures mitjançant TLS.
- Analitzar el model de seguretat resultant en comparació amb l'arquitectura original.
- Avaluar l'impacte del xifratge en el rendiment de les operacions de pujada i descàrrega.

5 DESENVOLUPAMENT

5.1 Planificació i seguiment del desenvolupament

La planificació del desenvolupament es va basar en la metodologia Kanban, amb un enfocament iteratiu i incremental. El treball es va organitzar en *epics* corresponents a les diferents fases del sistema, dels quals es van derivar tasques concretes.

Cada tasca va seguir un flux definit, iniciant-se en l'estat *Selected for Development*, passant a *In Progress* durant la implementació i entrant en fase de *Review* mitjançant *pull request*. Un cop validada la implementació i superades les proves corresponents, les tasques es van integrar a la branca principal i es van marcar com a *Done*.

5.2 Tecnologia utilitzada

El desenvolupament del sistema parteix del sistema base existent, mantenint la seva arquitectura i pila tecnològica, i incorpora noves capes de seguretat orientades a la protecció de les dades mitjançant E2EE i comunicacions segures.

Per a la protecció de les comunicacions entre client i servidor s'ha utilitzat TLS, habilitat mitjançant Nginx com a *reverse proxy*.

La funcionalitat E2EE s'ha implementat mitjançant la Web Crypto API [6] del navegador, que proporciona primitives criptogràfiques natives per a la generació de claus, xifrat i desxifrat de dades sense dependències externes.

5.3 Disseny criptogràfic i selecció de paràmetres

El sistema implementa un esquema de criptografia híbrida en què el xifratge simètric protegeix el contingut dels fitxers i el xifratge asimètric s'utilitza exclusivament per a la gestió de claus. El disseny ha estat condicionat per les primitives disponibles a la Web Crypto API, que defineix un conjunt limitat però estandarditzat d'algorismes criptogràfics (vegeu Annex A.3, Figura 34).

5.3.1 Xifratge simètric: AES-GCM

Per al xifratge del contingut dels fitxers s'ha utilitzat AES-GCM amb claus de 256 bits [7]. Aquest esquema AEAD proporciona confidencialitat i autenticació simultànies, simplificant el disseny criptogràfic i evitant la composició de mecanismes addicionals.

En comparació amb AES-CBC, AES-GCM ofereix millor eficiència computacional gràcies a la seva capacitat de paral·lelització i redueix la sensibilitat a errors d'implementació associats a esquemes de padding [8]. Aquestes característiques, juntament amb la seva adopció en protocols i estàndards moderns, justifiquen la seva selecció.

El sistema garanteix la generació d'un vector d'inicialització (IV) únic per operació, d'acord amb el requisit establert a NIST SP 800-38D [7]. La reutilització d'un IV sota la mateixa clau compromet les propietats de seguretat de l'esquema, afectant tant la confidencialitat com la integritat del sistema.

5.3.2 Xifratge asimètric: RSA-OAEP

Per a la protecció de claus simètriques s'ha utilitzat RSA-OAEP amb claus de 4096 bits i SHA-512 [9]. Aquest esquema s'utilitza exclusivament per a operacions de *key wrapping*, de manera que el rendiment no constitueix un factor crític.

L'elecció de claus de 4096 bits prioritza un marge de seguretat elevat davant d'avenços futurs en capacitat computacional, protegint les claus que assegurin el contingut dels fitxers.

Tot i que la Web Crypto API proporciona primitives basades en criptografia de corba el·líptica, aquestes requereixen esquemes addicionals per a *key wrapping*. En aquest sistema s'ha optat per RSA per simplicitat d'integració i flux de gestió de claus, ja que el seu ús es limita a fases puntuals del cicle de vida de les claus.

5.3.3 Derivació de claus: PBKDF2

Per a la derivació de claus a partir de contrasenyes s'ha utilitzat PBKDF2 amb SHA-512, un salt aleatori de 16 bytes i 100.000 iteracions [10]. Aquesta elecció ve determinada per les primitives disponibles a la Web Crypto API, on PBKDF2 és l'únic mecanisme compatible amb derivació de claus des de contrasenyes.

El nombre d'iteracions s'ha fixat seguint les recomanacions del NIST [11], com a compromís entre seguretat i rendiment en entorns d'ús general. Aquest valor també considera limitacions de dispositius de gamma baixa i mitjana, on un cost computacional elevat podria afectar el rendiment del sistema. Aquesta decisió és coherent amb sistemes E2EE com MEGA [13].

5.3.4 Clau de recuperació

La clau de recuperació es genera al client mitjançant un CSPRNG amb una longitud de 32 bytes (256 bits) i es codifica en Base64 per a la seva persistència. Posteriorment, s'importa com a clau simètrica AES-GCM i s'utilitza exclusivament per al xifratge de la clau privada RSA de l'usuari, permetent la recuperació del compte en cas de pèrdua de la contrasenya.

5.4 Implementació del sistema

5.4.1 Model de persistència i esquema de dades

El sistema va estendre el model de dades original per suportar el mode E2EE, introduint un esquema de persistència separat per a les entitats xifrades.

Tal com es mostra a la Figura 1, es van definir tres noves entitats: *UserKey*, *EncryptedFile* i *EncryptedFolder*, destinades a la gestió de claus i metadades del sistema E2EE.

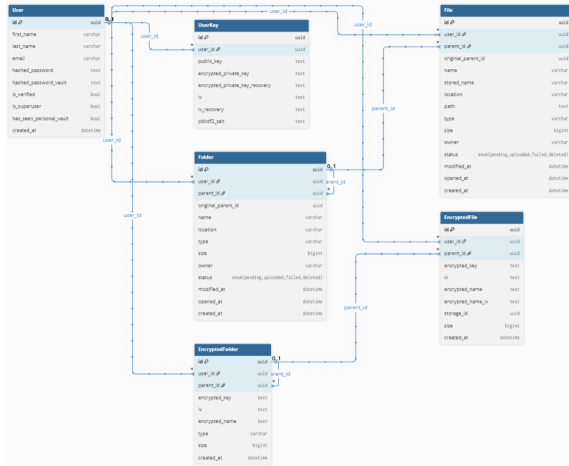


Fig. 1: Diagrama Entitat-Relació del sistema amb integració del model base i les entitats del sistema E2EE.

L'entitat *UserKey* va emmagatzemar el material criptogràfic de l'usuari, incloent claus públiques, claus privades protegides i paràmetres de derivació i recuperació de claus.

Les entitats *EncryptedFile* i *EncryptedFolder* van substituir les entitats *File* i *Folder* en el context E2EE, evitant la coexistència de camps no aplicables al mode xifrat i garantint un model específic per a dades protegides. Aquestes entitats van contenir únicament metadades necessàries per a la gestió dels recursos, com identificadors, estructura jeràrquica, mida i referències d'emmagatzematge.

Aquesta separació va permetre simplificar el model de dades i evitar dependències condicionals entre els modes del sistema base i E2EE.

5.4.2 Persistència de fitxers

El sistema va reutilitzar la infraestructura d'emmagatzematge del sistema base per al tractament del contingut xifrat, mantenint el sistema de fitxers del servidor com a capa de persistència física.

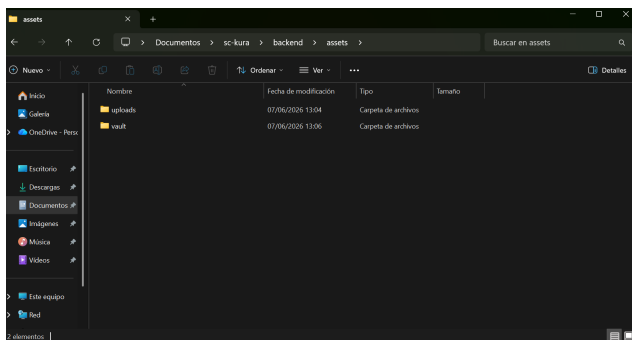


Fig. 2: Estructura dels directoris del sistema.

Es va definir una separació lògica entre dos espais d'emmagatzematge: el directori *uploads*, corresponent al sistema original, i el directori *vault*, destinat exclusivament al sistema E2EE. Aquesta separació es mostra a la Figura 2.

El nom físic dels fitxers emmagatzemats al servidor es va substituir per un identificador intern (*storage_id*) generat per la base de dades. Aquest identificador es va emmagatzemar a les metadades del fitxer i es va utilitzar com a referència per recuperar el contingut xifrat del sistema de fitxers. Aquesta aproximació va desacoblar la capa de persistència de metadades de l'emmagatzematge físic i va evitar l'exposició de noms originals al servidor.

Aquest model va permetre reutilitzar la infraestructura d'emmagatzematge existent sense introduir canvis en el mecanisme de persistència física dels fitxers.

5.4.3 Gestió del cicle de vida de claus

El sistema va implementar un mecanisme de gestió de cicle de vida de claus orientat a entorns E2EE, en què el material criptogràfic es genera, protegeix, persisteix i utilitza exclusivament al client (Figura 3).

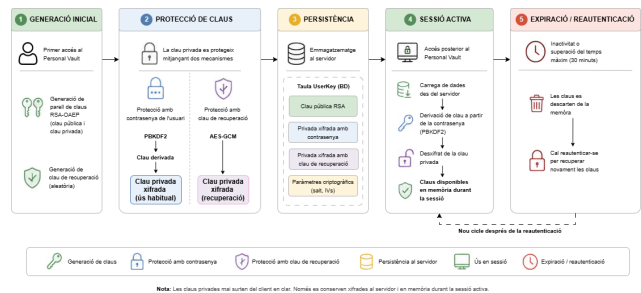


Fig. 3: Cicle de vida de les claus en el sistema E2EE.

La generació del material criptogràfic es va realitzar en el primer accés de l'usuari, incloent la creació del parell de claus RSA i d'una clau de recuperació. Aquestes claus es van protegir al client i es van persistir al servidor en forma xifrada mitjançant l'entitat *UserKey*.

En accessos posteriors, el sistema va recuperar les claus xifrades i els paràmetres associats, permetent reconstruir el material criptogràfic al client a partir de la contrasenya de l'usuari. Un cop desxifrades, les claus es van mantenir únicament en memòria durant la sessió activa.

Per reduir el risc d'exposició del material criptogràfic, el sistema va implementar un mecanisme de reautenticació després de 30 minuts d'inactivitat, seguint les recomanacions de gestió de sessions definides per NIST [11]. Un cop assolit aquest límit, les claus es van eliminar de la memòria i es va requerir una nova recuperació per continuar operant amb el sistema E2EE.

5.4.4 Operacions sobre fitxers

Les operacions sobre fitxers es van implementar seguint l'esquema híbrid de protecció de dades descrit anteriorment. Totes les operacions criptogràfiques es van executar al client mitjançant la Web Crypto API, mentre que el servidor va actuar exactament com a repositori de contingut xifrat i metadades. Les Figures 4, 5, 6 i 7 mostren els fluxos corresponents a les principals operacions implementades.

Pujada de fitxer La Figura 4 mostra el procés de pujada d'un fitxer al sistema. Durant aquesta operació, el client va generar una clau simètrica específica per a l'element, utilitzada per xifrar tant el contingut com el nom del fitxer. Posteriorment, aquesta clau es va protegir mitjançant *key wrapping* emprant la clau pública RSA de l'usuari abans d'enviar les dades al servidor. Per tant, el servidor únicament emmagatzema contingut xifrat, metadades protegides i claus encapsulades, sense disposar d'accés a la informació en clar.

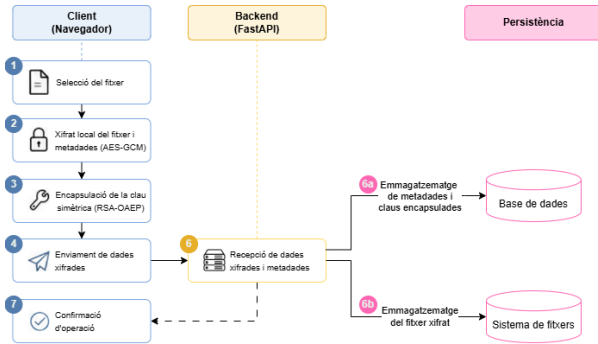


Fig. 4: Flux de pujada de fitxers al sistema E2EE.

Descàrrega de fitxer El procés de descàrrega es mostra a la Figura 5. En aquesta operació, el servidor retorna el contingut xifrat juntament amb les metadades associades. El client recupera la clau simètrica mitjançant la seva clau privada RSA i executa localment el procés de desxifrat. Un cop el fitxer ha estat desxifrat automàticament al client, es simula la descàrrega per part de l'usuari, permetent-ne l'accés i la seva emmagatzematge local. Aquest mecanisme garanteix que el contingut del fitxer només sigui reconstruït en clar al dispositiu de l'usuari.

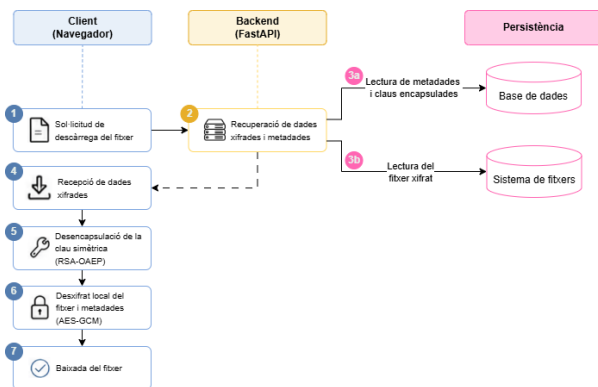


Fig. 5: Flux de descàrrega de fitxers al sistema E2EE.

Canvi de nom La Figura 6 mostra el flux corresponent a l'operació de canvi de nom. El nou valor introduït per l'usuari es xifra al client utilitzant la mateixa clau simètrica associada a l'element. Per a cada operació de xifrat del nom es genera un nou IV, el qual s'utilitza juntament amb la clau simètrica per garantir la seguretat semàntica del xifrat. Posteriorment, el nom xifrat i l'IV associat s'envien al servidor, on s'actualitzen les metadades corresponents a la base de dades. Aquesta estratègia permet mantenir la confidencialitat de les metadades sense modificar la clau simètrica de l'element.

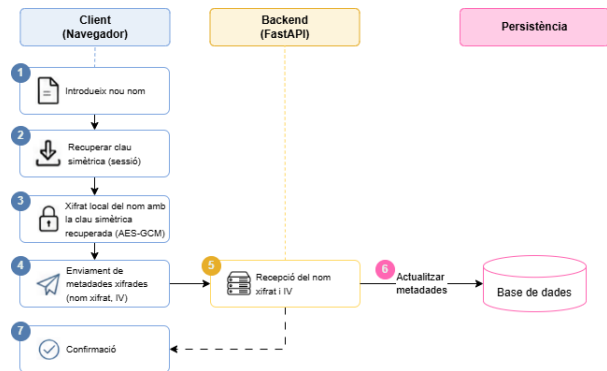


Fig. 6: Flux de canvi de nom d'un element al sistema E2EE.

Eliminació El procés d'eliminació es representa a la Figura 7. Aquesta operació no requereix transformacions criptogràfiques addicionals, ja que únicament implica la supressió de les metadades associades i del contingut xifrat emmagatzemat al sistema de fitxers.

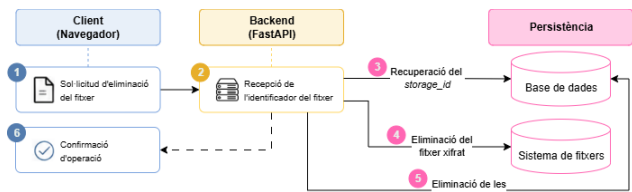


Fig. 7: Flux d'eliminació d'un element al sistema E2EE.

5.4.5 Gestió de carpetes

Les operacions sobre carpetes es van implementar seguint el mateix esquema de protecció de metadades que en el cas dels fitxers. A diferència d'aquests, les carpetes no contenen contingut, sinó únicament informació estructural del sistema. Les Figures 8, 6 i 9 mostren els fluxos corresponents a les principals operacions implementades.

Creació de carpeta La Figura 8 mostra el procés de creació d'una carpeta al sistema. Durant aquesta operació, el client genera una clau simètrica específica per a l'element i xifra el nom de la carpeta utilitzant aquesta clau. Posteriorment, la clau simètrica es protegeix mitjançant *key wrapping* amb la clau pública RSA de l'usuari abans de ser enviada al servidor. Per tant, el servidor únicament emmagatzema metadades xifrades i claus encapsulades, sense accés a la informació en clar.

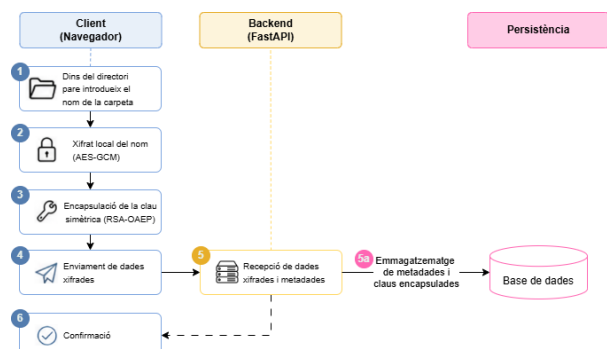


Fig. 8: Flux de creació d'una carpeta al sistema E2EE.

Les carpetes es modelen com una estructura jeràrquica basada en relacions pare-fill, on cada element conté una referència al seu directori pare. Aquesta informació es manté íntegrament a nivell de metadades xifrades.

Actualització de carpeta La Figura 6 mostra el flux corresponent a l'operació d'actualització de nom de carpeta, ja que aquesta és l'única modificació possible sobre una carpeta. Aquest procés segueix exactament el mateix mecanisme que el canvi de nom de fitxers. El nou valor es xifra al client utilitzant la clau simètrica associada a l'element. Per a cada operació de xifrat es genera un nou IV, garantint la seguretat semàntica del procés. Posteriorment, el nom xifrat i l'IV associat s'envien al servidor, que actualitza les metadades corresponents sense accedir a la informació en clar.

Eliminació de carpeta La Figura 9 mostra el flux corresponent a l'operació d'eliminació. Aquest procés és equivalent al de l'eliminació de fitxers, amb la particularitat que, en rebre l'identificador d'una carpeta, el servidor recupera de manera recursiva tots els *storage_id* dels fitxers associats a aquesta carpeta i a totes les seves subcarpetes. A partir d'aquesta informació, s'elimina el contingut corresponent del sistema de fitxers.

Posteriorment, s'elimina únicament la carpeta identificada, ja que la resta d'elements (subcarpetes i metadades associades) es propaguen automàticament mitjançant l'operació *CASCADE*, garantint la consistència de l'estructura sense intervenció addicional del client.

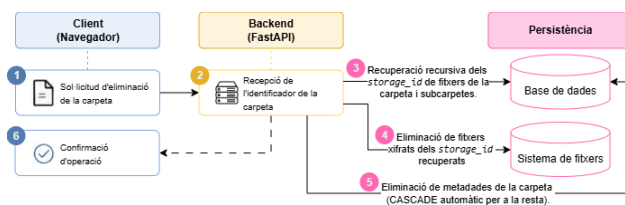


Fig. 9: Flux d'eliminació d'una carpeta al sistema E2EE.

5.4.6 Gestió de sessió i seguretat en client

El sistema va implementar un model de gestió de claus en memòria per minimitzar l'exposició del material criptogràfic al costat del client. Un cop recuperades i desxifrades, les claus es van mantenir exclusivament a la memòria del navegador durant la sessió activa, evitant-ne l'emmagatzematge persistent en clar mitjançant mecanismes com *localStorage*, *sessionStorage* o base de dades locals.

En cas de recàrrega de la pàgina o inici d'una nova sessió del navegador, les claus es van eliminar de la memòria i es va requerir un nou procés de recuperació a partir de les credencials de l'usuari. Aquest comportament va permetre limitar la persistència temporal del material criptogràfic i reduir l'impacte potencial d'un compromís del dispositiu client.

Adicionalment, es va implementar un mecanisme de re-autenticació automàtica cada 30 minuts. Un cop superat aquest interval, les claus es van descartar de la memòria i l'usuari va haver de tornar a autenticar-se per continuar accedint a les funcionalitats del sistema E2EE.

5.4.7 Mecanisme de recuperació d'accés

Per garantir la recuperació de l'accés al sistema en cas de pèrdua de la contrasenya, es va implementar un mecanisme basat en una clau de recuperació generada durant el primer accés al Personal Vault. Aquesta clau es va lliurar a l'usuari i va actuar com a únic factor de recuperació extern al sistema.

La Figura 10 mostra el flux complet del procés de recuperació. Quan l'usuari iniciava el procediment, el sistema sol·licitava la clau de recuperació i recuperava del servidor el material criptogràfic necessari. La validesa d'aquesta clau es verificava intentant recuperar la clau privada de l'usuari. En cas d'error, el procés era interromput i es requeria la introducció d'una nova clau de recuperació.

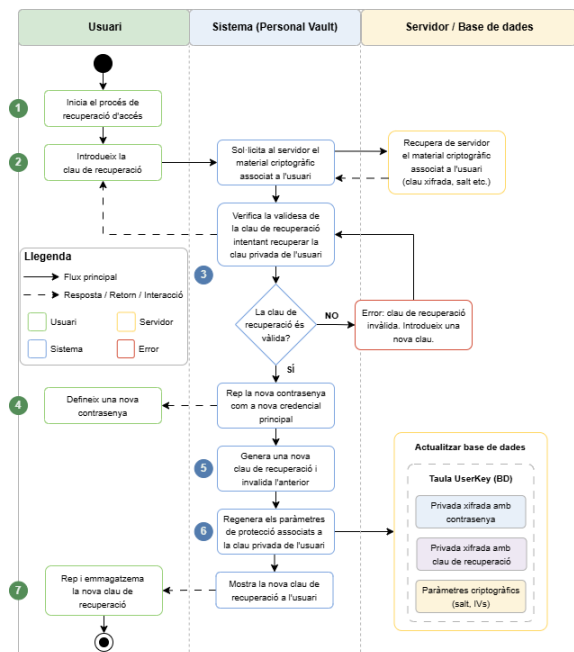


Fig. 10: Flux del mecanisme de recuperació d'accés al sistema E2EE.

Un cop validada la clau de recuperació, l'usuari definia una nova contrasenya, que passava a ser la credencial principal al Personal Vault. A continuació, es regeneraven els paràmetres de protecció associats a la clau privada i s'actualitzava la informació emmagatzemada a la base de dades.

Adicionalment, el procés generava una nova clau de recuperació i invalidava l'anterior. Aquesta decisió va permetre vincular el cicle de vida de la clau de recuperació al de la contrasenya, evitant que una clau de recuperació compromesa mantingués accés indefinit al compte després d'un canvi de credencials. La nova clau de recuperació es mostrava a l'usuari perquè pogués emmagatzemar-la de forma segura per a futures recuperacions.

5.4.8 Desplegament del sistema

El desplegament del sistema es va basar en l'ampliació de la infraestructura original mitjançant la incorporació d'un servidor Nginx actuant com a *reverse proxy*. Aquest component es va configurar com a únic punt d'entrada al sistema, centralitzant el trànsit de xarxa entre els clients i els serveis interns.

Nginx escolta les peticions HTTP al port 80 i redirigeix automàticament el trànsit cap a HTTPS al port 443. Addicionalment, realitza la terminació TLS, garantint que totes les comunicacions entre client i servidor es mantinguin xifrades.

El sistema s'ha executat en un entorn *self-hosted* basat en Docker, on tots els serveis es troben desplegats dins d'una xarxa privada. L'aplicació principal es troba en un únic contenidor que integra tant el front-end com el back-end, fet que simplifica la comunicació interna entre ambdues capes sense necessitat d'exposició de ports addicionals.

La base de dades s'executa com a servei independent dins de la mateixa xarxa de contenidors, amb persistència de dades mitjançant volums vinculats al sistema de fitxers del servidor. De manera similar, els fitxers carregats pels usuaris s'emmagatzemen en el sistema de fitxers del servidor a través de volums de Docker, garantint-ne la persistència i disponibilitat entre reinicis dels contenidors.

Tal com es mostra a la Figura 11, tots els serveis romanen aïllats dins de la xarxa privada de Docker, i únicament el *reverse proxy* actua com a punt d'accés al sistema.

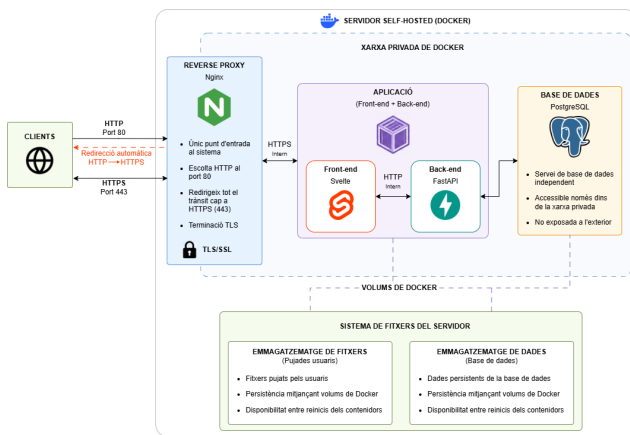


Fig. 11: Arquitectura de desplegament del sistema.

6 RESULTATS

Aquesta secció presenta els resultats obtinguts en el desenvolupament del sistema amb l'objectiu de validar el compliment dels objectius definits i demostrar el funcionament de les funcionalitats implementades.

6.1 Integració transparent del Personal Vault

La interfície del sistema presenta el Personal Vault com un espai d'emmagatzematge convencional, sense mostrar a l'usuari els mecanismes criptogràfics que s'executen internament. Les operacions de xifratge i desxifratge es realitzen al costat del client de manera transparent durant l'ús del sistema.

Les captures de la interfície web es poden consultar a l'Annex A.2.

6.2 Comparació de l'emmagatzematge de fitxers

Per validar el correcte funcionament del mecanisme de xifratge del costat del client, s'ha realitzat una comparativa

entre el sistema d'emmagatzematge tradicional i el Personal Vault. L'objectiu ha estat verificar l'estat real del contingut dels fitxers un cop persistits al servidor.

Per a aquesta validació, s'ha utilitzat un mateix fitxer en ambdós escenaris. En primer lloc, el fitxer s'ha pujat al sistema base, i posteriorment s'ha repetit el procés dins del Personal Vault. En ambdós casos s'ha accedit directament al sistema de fitxers del servidor per inspeccionar el contingut emmagatzemat.

En el sistema base, el contingut del fitxer es manté en clar al sistema de fitxers del servidor, tal com es mostra a la Figura 12. Això confirma que el servidor té accés directe a la informació sense cap mecanisme de protecció criptogràfica.

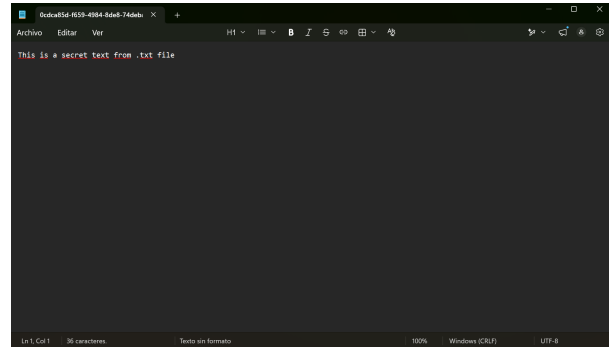


Fig. 12: Fitxer emmagatzemat en clar al sistema base sense xifratge.

En canvi, en el Personal Vault, el mateix fitxer apareix emmagatzemat en forma de dades xifrades, sense possibilitat d'interpretar-ne el contingut, tal com es mostra a la Figura 13. Això evidencia que el procés de xifratge es realitza completament al costat del client abans de la transmissió i persistència.

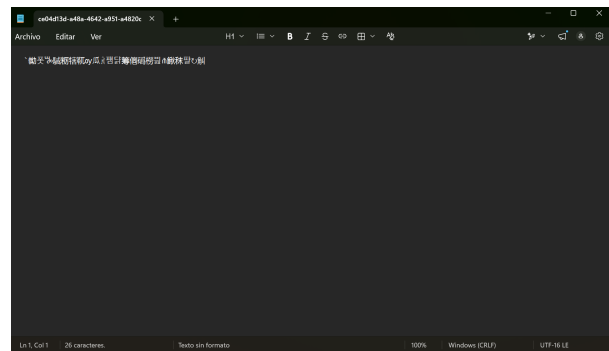


Fig. 13: Fitxer emmagatzemat xifrat al Personal Vault amb E2EE.

La comparació directa entre ambdós escenaris permet validar empíricament la diferència entre el model d'emmagatzematge tradicional i el model basat en E2EE, confirmant la confidencialitat del contingut en el segon cas.

6.3 Validació del model de comunicacions

Per validar el model de comunicacions segures implementat, s'han analitzat les transmissions de xarxa mitjançant l'eina Wireshark en tres escenaris diferents: el sistema base sense TLS, el sistema amb E2EE sense TLS, i el sistema complet amb TLS habilitat.

En el primer escenari, corresponent al sistema base, les comunicacions es realitzen mitjançant HTTP sense cap mecanisme de protecció a nivell de transport. Tal com es mostra a la Figura 14, és possible observar el contingut del fitxer en clar dins de les peticions HTTP, fet que evidencia l'absència de confidencialitat en la transmissió de dades.

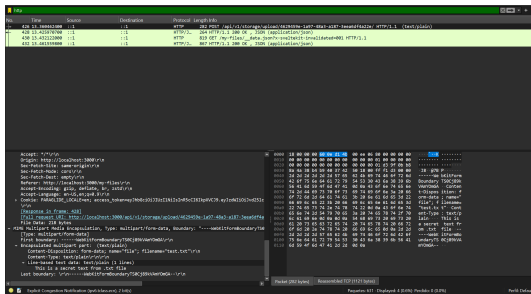


Fig. 14: Intercepció de paquets HTTP en el sistema base sense TLS, amb contingut en clar.

En el segon escenari, corresponent al Personal Vault sense TLS, el contingut dels fitxers ja es troba xifrat al costat del client abans de la transmissió. En aquest cas, tot i que les comunicacions continuen utilitzant HTTP, el contingut capturat per Wireshark apareix com a dades xifrades no interpretables, tal com es mostra a la Figura 15. Això confirma que la confidencialitat del contingut es garanteix a nivell d'aplicació, independentment del canal de comunicació.

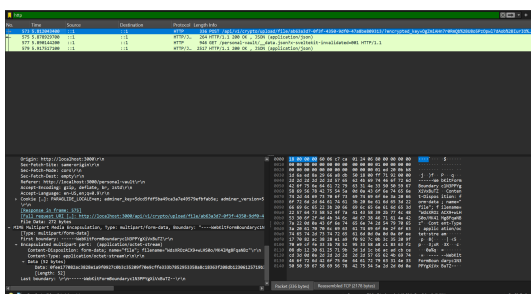


Fig. 15: Intercepció de paquets HTTP amb E2EE sense TLS, amb contingut xifrat.

Finalment, en el tercer escenari, amb la incorporació de TLS com a mecanisme de seguretat a nivell de transport, les comunicacions es realitzen mitjançant HTTPS. Tal com es pot observar a la Figura 16, el trànsit de xarxa apareix encapsulat dins del protocol TLS, de manera que no és possible accedir al contingut de les peticions HTTP ni a les dades transmeses. Això garanteix la confidencialitat i la integritat de les comunicacions entre client i servidor.

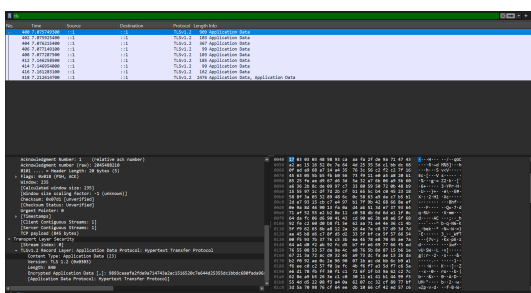


Fig. 16: Intercepció de paquets HTTPS amb TLS, trànsit encapsulat i no interpretable.

En conjunt, els resultats obtinguts evidencien la millora progressiva en seguretat aportada per la combinació de xifratge a nivell d'aplicació i a nivell de transport, eliminant la possibilitat d'intercepció de dades en clar durant la transmissió.

6.4 Avaluació del rendiment del sistema

L'avaluació del rendiment del sistema s'ha realitzat amb l'objectiu de quantificar l'impacte introduït per la incorporació del E2EE sobre les operacions de pujada i descàrrega de fitxers. Les mesures s'han obtingut comparant el sistema base amb el Personal Vault en un entorn controlat.

Metodologia experimental Les proves s'han realitzat en un ordinador portàtil amb processador Intel evo i5-1340P i 16 G de RAM, utilitzant Docker Desktop sobre Windows amb back-end basat en WSL2. El navegador utilitzat ha estat Google Chrome (versió 149.0.7827.104, 64 bits), sense extensions actives i en mode normal.

El sistema s'ha desplegat en entorn local i s'ha accedit mitjançant localhost, utilitzant una connexió wifi estable amb una velocitat aproximada de 180 Mbps. Les condicions de xarxa s'han mantingut constants durant totes les proves.

Per a cada mida de fitxer, les operacions s'han repetit cinc vegades i s'ha utilitzat la mitjana aritmètica dels valors obtinguts. Les mesures s'han obtingut mitjançant temporitzadors basats en la Web Performance API integrats directament en el codi del client, registrant punts de control en cada fase de les operacions de pujada i descàrrega.

Configuració dels experiments S'han avaluat diferents mides de fitxer per analitzar l'escalabilitat del sistema: 1 KB, 1 MB, 5 MB, 10 MB, 500 MB, 750 MB, 1 GB i 1.5 GB. Per a cada mida s'han mesurat tant el sistema base com el Personal Vault, considerant operacions de pujada i descàrrega.

En el cas del Personal Vault, les mesures es desglossen el cost de les diferents operacions criptogràfiques (xifrat de contingut, xifrat de nom, encapsulació de claus i desxifrat), així com el temps total de transmissió i persistència.

Resultats de pujada de fitxers A partir dels resultats recollits a les Taules 1 i 2, s'observa el comportament del sistema durant el procés de pujada en funció de la mida del fitxer i el model de seguretat aplicat. En el cas del Personal Vault, el temps total incorpora fases addicionals de processament local, incloent la generació i encapsulació de claus, el xifrat del contingut i del nom del fitxer i l'operació de *key wrapping*, inexistents en el sistema base.

El cost computacional associat al xifrat creix de manera aproximadament lineal amb la mida del fitxer, sent el xifrat del contingut la fase més costosa del processament local previ a la transmissió (Taula 1). No obstant això, aquest increment no es reflecteix proporcionalment en el temps total de pujada, ja que el component principal continua sent la transferència de dades al servidor (Taula 2).

En conseqüència, l'*overhead* introduït per E2EE es concentra principalment al client, mentre que el temps global de pujada està dominat per la latència de xarxa i l'amplada de banda disponible. Això explica que, fins i tot de grans

mides, la diferència entre el sistema base i el sistema xifrat sigui moderada en termes de temps total.

Resultats de descàrrega de fitxers A partir dels resultats obtinguts a les Taules 3 i 4, es caracteritza el comportament del sistema durant el procés de descàrrega sota el model E2EE i el sistema base. En el cas del Personal Vault, el temps total de descàrrega inclou, a més de la transferència de dades, les operacions de recuperació de claus, conversió de buffers, desxifrat del contingut i emmagatzematge local del fitxer.

Els resultats mostren que el temps de descàrrega està fortament dominat per la transferència de dades, especialment en fitxers de gran mida, mentre que les operacions criptogràfiques presenten un cost relatiu baix però creixent amb la mida del fitxer. En particular, el desxifrat del contingut i la conversió de buffers esdevenen les fases més rellevants dins del processament local per a fitxers grans (Taula 3).

En comparació amb el sistema base (Taula 4), s'observa que l'E2EE introdueix un *overhead* addicional en el processament de descàrrega, principalment concentrat en les operacions de desxifrat. No obstant això, aquest increment no modifica el patró general del rendiment, ja que el cost dominat continua sent la descàrrega de dades des del servidor, especialment per a fitxers de gran volum.

Síntesi global En conjunt, els resultats obtinguts mostren que la incorporació d'E2EE introdueix un cost computacional addicional local en les operacions de pujada i descàrrega, principalment associat al xifrat i desxifrat de contingut. No obstant això, aquest *overhead* es manté acotat en escenaris de mida petita i mitjana, mentre que en fitxer de gran volum el rendiment global del sistema continua estant dominat pel temps de transferència de xarxa. Això indica que el cost de seguretat s'integra de manera previsible dins del comportament del sistema i no altera la seva escalabilitat funcional.

6.5 Limitacions identificades i treball futur

A partir dels resultats obtinguts en l'avaluació del rendiment i la validació funcional, s'han identificat diverses limitacions derivades del model E2EE i de la seva implementació. En primer lloc, es detecta un *overhead* computacional en el costat del client associat a les operacions de xifrat i desxifrat, especialment en fitxers de gran mida, amb impacte en dispositius amb recursos limitats. En segon lloc, el temps total de pujada i descàrrega està dominat pel cost de transferència de dades, reduint l'efecte d'optimitzacions criptogràfiques sobre el rendiment global.

En descàrrega, el processament de buffers i el desxifrat incrementen el cost en escenaris de gran volum, evidenciant una limitació d'escalabilitat vinculada a la necessitat de carregar el fitxer complet en memòria. Addicionalment, el sistema no permet la cerca sobre contingut xifrat, ja que el servidor no té accés a les dades en clar per disseny del model E2EE.

Com a treball futur, es proposa l'estudi de tècniques de *Searchable Symmetric Encryption* per habilitar funcionalitats de cerca sobre dades xifrades, així com la incorporació de mecanismes basats en *streams* per reduir el consum de

memòria en operacions de descàrrega i desxifrat. Addicionalment, es podrien explorar estratègies d'optimització de la transferència de fitxers, com ara la transmissió per fragments (*chunking*), protocols més eficients o mecanismes de paral·lelització de la pujada i descàrrega, amb l'objectiu de mitigar l'impacte del cost de comunicació observat en els resultats.

7 CONCLUSIONS

Aquest treball ha presentat el disseny i implementació d'una funcionalitat de *Personal Vault* basada en E2EE integrada en un sistema *self-hosted* de gestió de fitxers. La principal aportació no resideix únicament en la incorporació de tècniques criptogràfiques, sinó en la seva integració transparent dins d'un flux funcional ja existent, mantenint simultàniament el model d'emmagatzematge tradicional.

Els resultats experimentals mostren que el cost computacional introduït pel xifratge és rellevant però acotat, i que el rendiment global del sistema està principalment condicionat per la transferència de dades. Això implica que el benefici del model E2EE és més evident en escenaris on la sensibilitat de la informació és prioritària i el volum de dades és reduït o moderat, mentre que en fitxers de gran mida el guany de seguretat no es tradueix en millores de rendiment.

Des d'una perspectiva crítica, el principal repte del treball ha estat la coordinació entre capes del sistema (client, servidor i xarxa) i la correcta separació entre lògica funcional i processos criptogràfics, garantint que la seguretat no comprometi la usabilitat.

Contribucions principals del sistema:

- Integració d'un esquema de xifratge E2EE en un sistema *self-hosted* existent.
- Millora de la privacitat i confidencialitat del contingut i metadades de fitxers.
- Introducció d'un model de control d'accés basat en claus gestionades al costat del client.
- Compatibilitat amb un flux d'ús tradicional sense modificacions en l'experiència d'usuari.
- Anàlisi empírica del compromís entre seguretat i rendiment en operacions de fitxers.

En conjunt, el treball demostra la viabilitat d'integrar E2EE en sistemes d'emmagatzematge de fitxers sense alterar-ne el funcionament base, assumint un cost computacional que resulta justificat en entorns on la confidencialitat és un requisit prioritari.

8 ÚS DE LA INTEL·LIGÈNCIA ARTIFICIAL GENERATIVA

Durant el desenvolupament d'aquest treball s'ha utilitzat la intel·ligència artificial generativa com a eina de suport puntual en tasques de redacció i revisió de text, així com per a la millora de la claredat i coherència de determinades explicacions tècniques.

En cap cas la intel·ligència artificial ha substituït el procés de desenvolupament del sistema, ni ha generat codi final

sense revisió. Totes les decisions d'arquitectura, implementació i validació han estat preses i verificades per l'autor del treball.

L'ús d'aquestes eines s'ha limitat a un suport auxiliar per a la formulació i estructuració del contingut escrit.

9 AGRAÏMENTS

Voldria expressar el meu agraïment al meu tutor del treball, Rubén Martínez Vidal, pel seguiment continuat durant el desenvolupament del projecte. Els seus comentaris, suggeriments i orientacions han estat fonamentals per guiar el treball i millorar-ne la qualitat en les diferents fases del desenvolupament.

Aquest suport ha permès estructurar adequadament el projecte i enfocar correctament les decisions tècniques preses al llarg del procés.

REFERÈNCIES

- [1] P. Mell i T. Grance, «The NIST Definition of Cloud Computing», National Institute of Standards and Technology, NIST Special Publication (SP) 800-145, set. 2011. doi: 10.6028/NIST.SP.800-145.
- [2] B. V. der Kolk, «How does MEGA's zero-knowledge encryption work?», MEGA Help Centre. Consulta: 21 de maig de 2026. [En línia]. Disponible a: <https://help.mega.io/security/data-protection/zero-knowledge-encryption>
- [3] «The Proton Drive security model», Proton. Consulta: 8 de juny de 2026. [En línia]. Disponible a: <https://proton.me/blog/protondrive-security>
- [4] «Bitwarden Security Whitepaper», Bitwarden. Consulta: 8 de juny de 2026. [En línia]. Disponible a: <https://bitwarden.com/help/bitwarden-security-white-paper/>
- [5] J. Peix, «Kura: Self-Hosted File Storage System», GitHub. Consulta: 23 de maig de 2026. [En línia]. Disponible a: <https://github.com/janpeix04/sc-kura>
- [6] «Web Crypto API - Web APIs — MDN», MDN Web Docs. Consulta: 25 de maig de 2026. [En línia]. Disponible a: https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API
- [7] M. Dworkin, «Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC», National Institute of Standards and Technology, NIST Special Publication (SP) 800-38D, nov. 2007. doi: 10.6028/NIST.SP.800-38D.
- [8] «What's the difference between AES-CBC and AES-GCM?», Private Internet Access, Consulta: 12 de juny de 2026. [En línia]. Disponible a: <https://helpdesk.privateinternetaccess.com/hc/en-us/articles/46777125333019-What-s-the-difference-between-AES-CBC-and-AES-GCM>
- [9] R. Housley, «Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS)», Internet Engineering Task Force, Request for Comments (RFC) 3560, jul. 2003. doi: 10.17487/RFC3560.
- [10] R. Moriarty, B. Kaliski i A. Rusch, «PKCS #5: Password-Based Cryptography Specification Version 2.1», Internet Engineering Task Force, Request for Comments (RFC) 8018, gen. 2017. doi: 10.17487/RFC8018.
- [11] P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr i J. P. Richter, «Digital Identity Guidelines: Authentication and Lifecycle Management», National Institute of Standards and Technology, NIST Special Publication (SP) 800-63B, jun. 2017, updated 2 Mar. 2020. doi: 10.6028/NIST.SP.800-63B.
- [12] M. Sönmez Turan, E. Barker, W. Burr, i L. Chen, «Recommendation for Password-Based Key Derivation: Part 1: Storage Applications», National Institute of Standards and Technology, NIST Special Publication (SP) 800-132, des. 2010. doi: 10.6028/NIST.SP.800-132.
- [13] MEGA Limited, «Security Whitepaper», Third Edition, jun. 2022. Disponible a: <https://mega.nz/SecurityWhitepaper.pdf>.

APÈNDIX

A.1 Vista general del sistema base

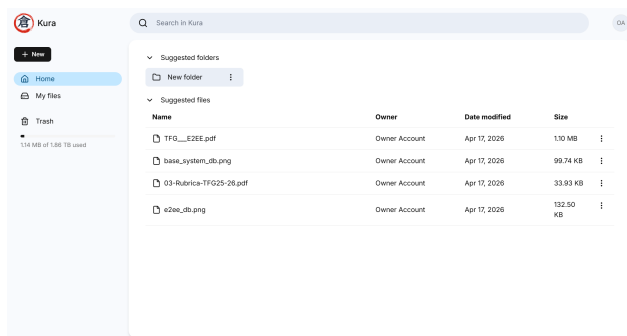


Fig. 17: Pantalla principal del sistema base, on es mostra la navegació general i l'accés als mòduls principals.

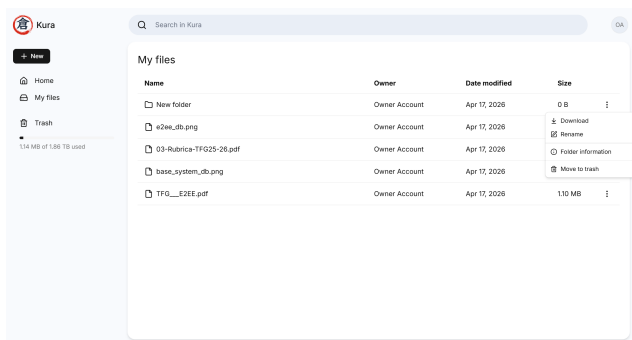


Fig. 18: Secció "My files", on es visualitzen i gestionen els arxius de l'usuari.

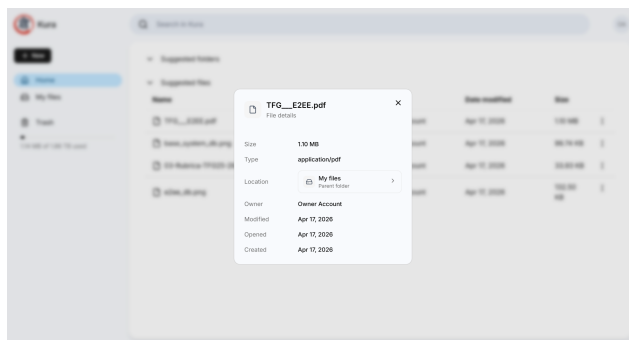


Fig. 22: Vista de detalls d'un fitxer, amb informació com el nom, tipus, mida i data de modificació.

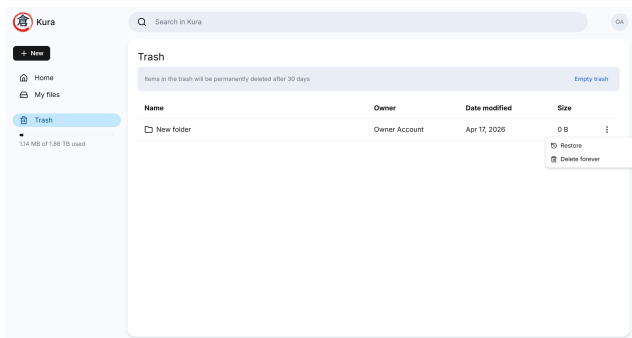


Fig. 19: Paperera del sistema base, on es poden recuperar o eliminar definitivament els fitxers esborrats.

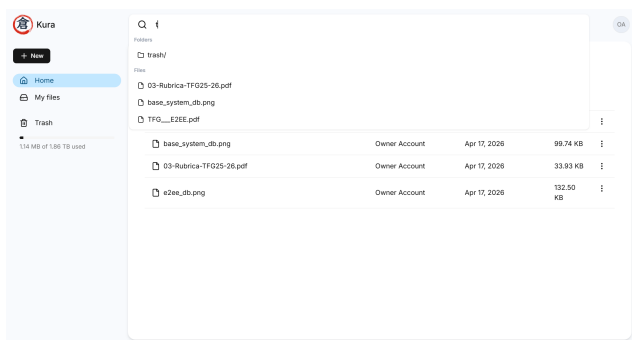


Fig. 20: Interfície de cerca del sistema base per localitzar fitxers o contingut dins del sistema.

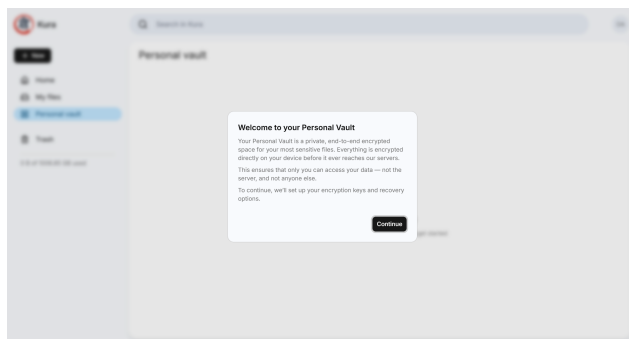


Fig. 23: Diàleg de benvinguda en el primer accés al sistema.

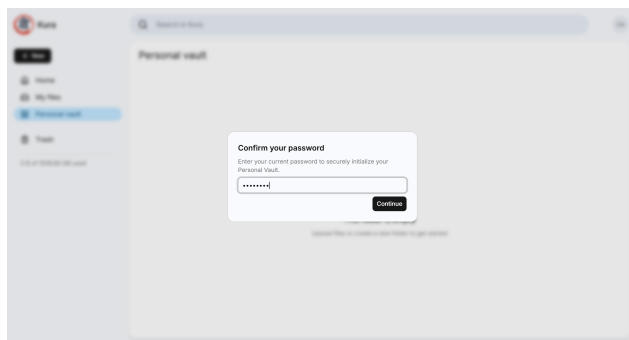


Fig. 24: Introducció de contrasenya durant el registre inicial.

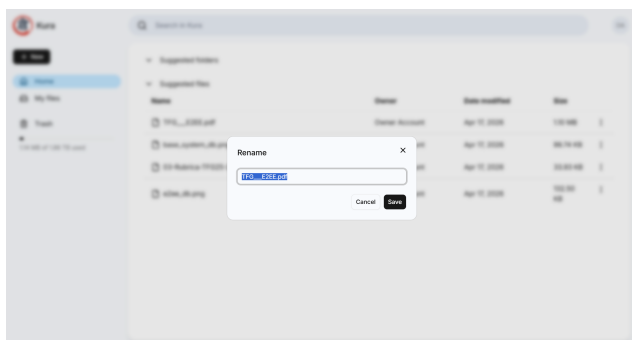


Fig. 21: Finestra de canvi de nom d'un fitxer dins del sistema base.

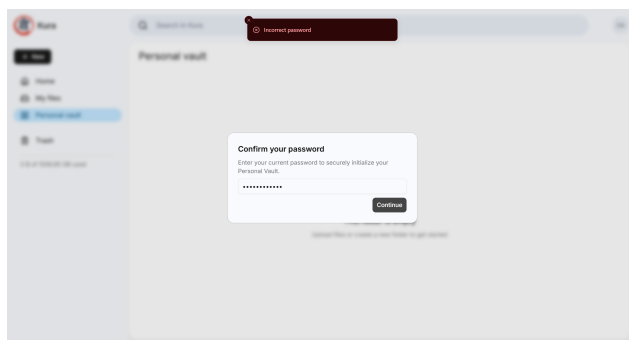


Fig. 25: Error de validació en la introducció de la contrasenya.

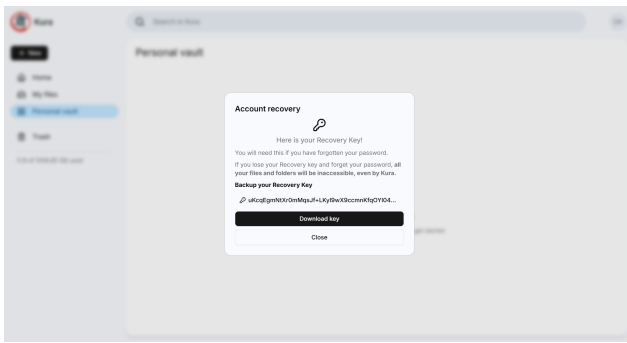


Fig. 26: Visualització de la clau de recuperació del compte.

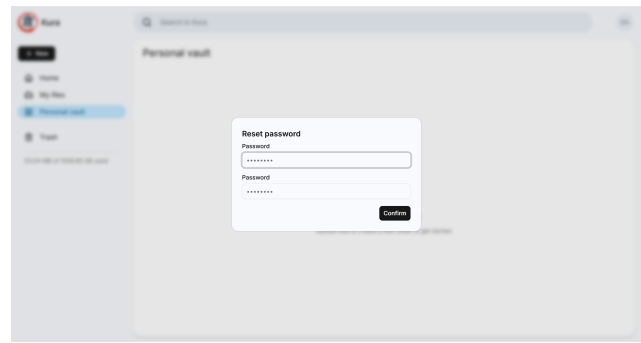


Fig. 30: Definició d'una nova contrasenya durant la recuperació del compte.

A.2.2 Autenticació i recuperació de compte

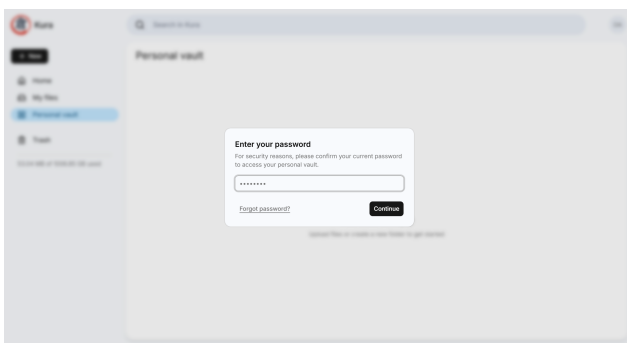


Fig. 27: Pantalla d'inici de sessió per a accessos posteriors.

A.2.3 Gestió de fitxers

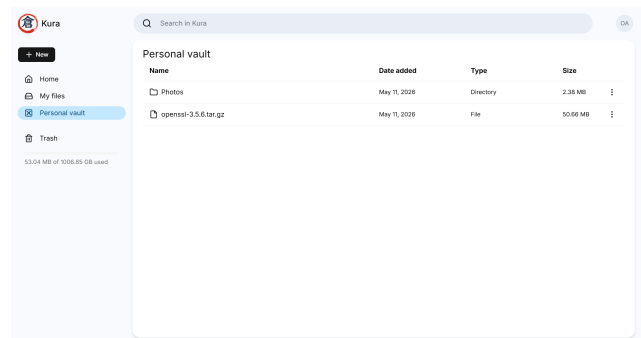


Fig. 31: Visualització de carpetes i fitxers amb interfície tipus sistema tradicional.

A.2.4 Accions sobre elements

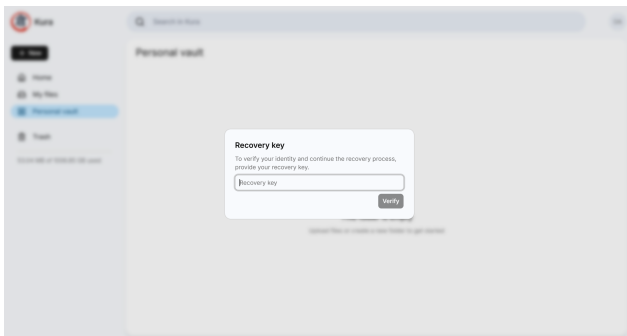


Fig. 28: Introducció de la clau de recuperació.

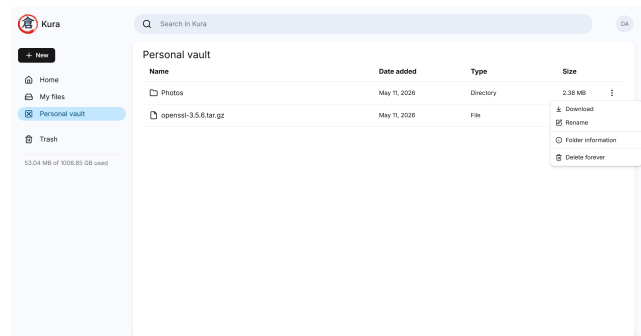


Fig. 32: Opcions disponibles per a la gestió d'elements.

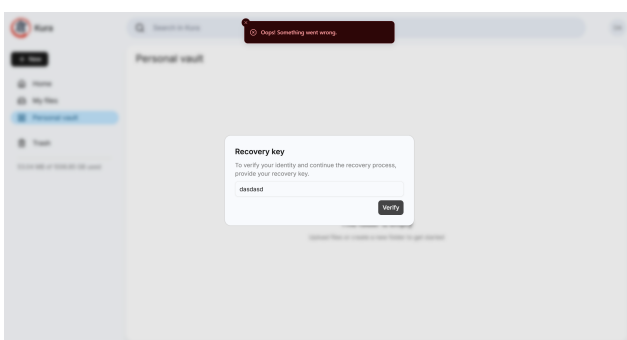


Fig. 29: Error en la validació de la clau de recuperació.

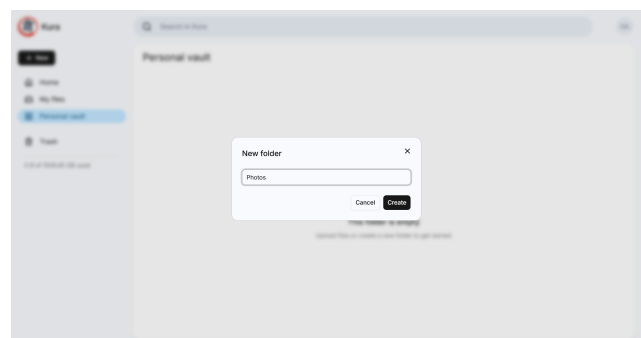


Fig. 33: Creació d'una nova carpeta dins del sistema.

A.3 Web Crypto API

Algorithm name	encrypt	decrypt	sign	verify	digest	generateKey	deriveKey	deriveBits	importKey	exportKey	wrapKey	unwrapKey
RSASSA-PKCS1-v1_5			✓	✓		✓			✓	✓		
RSA-PSS			✓	✓		✓			✓	✓		
RSA-OAEP	✓	✓				✓			✓	✓	✓	✓
ECDSA			✓	✓		✓			✓	✓		
ECDH						✓	✓	✓	✓	✓		
Ed25519			✓	✓		✓			✓	✓		
X25519						✓	✓	✓	✓	✓		
AES-CTR	✓	✓				✓			✓	✓	✓	✓
AES-CBC	✓	✓				✓			✓	✓	✓	✓
AES-GCM	✓	✓				✓			✓	✓	✓	✓
AES-KW						✓			✓	✓	✓	✓
HMAC			✓	✓		✓			✓	✓		
SHA-1					✓							
SHA-256					✓							
SHA-384					✓							
SHA-512					✓							
HKDF							✓	✓	✓			
PBKDF2							✓	✓	✓			

Fig. 34: Relació entre els algorismes criptogràfics suportats per la Web Crypto API i les operacions disponibles

A.4 Resultats avaluació del rendiment del sistema

TAULA 1: TEMPS DE PROCESSAMENT I PUJADA SEGONS LA MIDA DEL FITXER (MS)

Mida fitxer	Lectura	Nom xifrat	Contingut xifrat	Key wrap	Pujada	Total
1 KB	1,6	0,4	0,1	0,2	42,3	44,6
1 MB	8,1	0,2	2,3	0,3	169	179,9
5 MB	10,8	0,2	7,9	0,3	206,1	225,3
10 MB	17,5	0,3	13,4	0,3	375,9	407,4
500 MB	274,6	0,1	524,1	0,3	5481,5	6280,6
750 MB	563,6	0,1	702,8	0,4	8459,1	9726
1 GB	966,1	0,5	1822,4	0,3	21339,3	24128,6
1,5 GB	2026,4	0,1	3878,8	0,4	35443,2	41348,9

TAULA 2: TEMPS DE PUJADA SEGONS LA MIDA DEL FITXER (MS)

Mida fitxer	Temps de pujada (ms)
1 KB	38,2
1 MB	124,9
5 MB	175,6
10 MB	247,5
500 MB	6237,8
750 MB	9632,6
1 GB	16091,3
1,5 GB	37833,5

TAULA 3: TEMPS DE DESCODIFICACIÓ I DESCÀRREGA SEGONS LA MIDA DEL FITXER (MS)

Mida fitxer	Descàrrega	Key unwrap	Buffer conv.	Desxifrat	Desat	Total
1 KB	81,2	5,3	1,3	0,1	1,8	89,7
1 MB	1377,2	5,1	2,5	1,6	2,3	1388,7
5 MB	7813,3	5,7	4,6	7,3	3,3	7834,2
10 MB	15563,6	5,8	10,2	13,5	4,3	15597,4
500 MB	775282,8	5,7	490,2	845,3	142,6	776766,6
750 MB	1179079,9	5,3	1466,7	1987,7	349,4	1182889
1 GB	1396756,5	5,2	726,1	1148,3	268,9	1398905
1,5 GB	1789734,8	5,1	1070,7	2693,1	485,6	1793989,3

TAULA 4: TEMPS DE DESCÀRREGA I DESAT SEGONS LA MIDA DEL FITXER (MS)

Mida fitxer	Descàrrega	Desat	Total
1 KB	63,0	0,9	63,9
1 MB	63,6	0,6	64,2
5 MB	166,4	1,5	167,9
10 MB	268,2	0,6	268,8
500 MB	9773,2	0,8	9774,0
750 MB	15440,1	0,7	15440,8
1 GB	24702,3	0,9	24703,2
1,5 GB	33191,2	1,0	33192,2