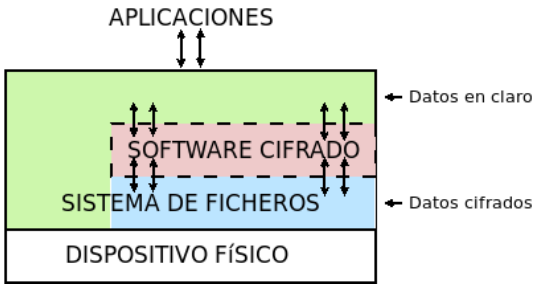
Xifrat de discos/particions o carpetes (II)

A nivell de carpetes

Les solucions de xifratge a nivell no de bloc sinó de sistemes de fitxers s'implementen com una capa que s'apila a sobre d'un sistema de fitxers existent. Això fa que totes els arxius escrits en una carpeta específicament habilitada per a xifratge es xifrin (gràcies a un determinat software específic) abans que el sistema de fitxers subjacent els escrigui al disc i es desxifrin sempre que el sistema de fitxers els llegeixi des del disc. Aquest xifratge no només afecta al contingut dels fitxers i subcarpetes presents sota la carpeta en qüestió sinó també als seus propis noms, substituint-se per cadenes d´aspecte aleatori.

La manera com s'implementa aquest mecanisme de xifrat, tal com mostra la figura, és mitjançant el muntatge de la carpeta del sistema de fitxers amfitrió que emmagatzema els arxius xifrats ("directori inferior") sobre una ubicació diferent -o sobre sí mateixa!- ("directori superior") utilitzant un pseudo-sistema de fitxers apilat especial (aquest procés demanarà una contrasenya-passphrase). En aquesta nova ubicació és on apareixeran els mateixos fitxers en forma llegible, fins que aquesta es desmunti (o el sistema s´apagui).



La idea és que els fitxers que es creïn al directori "superior" es guarden aparentment de forma *normal* (sense xifrar) però no podran manipular-se directament a través del "directori inferior" perquè allà apareixeran com que, efectivament, estan xifrats. I si els fitxers es creen directament al directori xifrat només es podran veure a través d'aquest directori perquè no seran visibles al directori "superior".

> NOTA: No cal utilitzar sempre el mateix directori "superior" per un determinat directori "inferior", pot ser un de diferent cada cop. L´únic important és que la contrasenya utilitzada pel muntatge sempre haurà de ser la correcta.

Hi ha diversos mètodes/tecnologies/programari disponibles a Linux:

Encfs (https://vgough.github.io/encfs): Treballa a nivell d'usuari amb FUSE (no cal ser root per muntar)
eCryptfs (http://ecryptfs.org): Treballa a nivel de kernel (cal ser root per muntar). Obsolet
Securefs (https://github.com/netheril96/securefs) : Similar a Encfs però més actualitzat i mantingut
Gocryptfs (https://nuetzlich.net/gocryptfs) : Similar a Encfs però més actualitzat i mantingut
CryFS (https://www.cryfs.org/howitworks) : Aconsegueix xifrar, a més dels continguts dels fitxers, també les seves metadades (per tal de què no es pugui conéixer, per exemple, la seva mida, els seus timestamps o la seva ubicació dins de l'arbre de fitxers, etc). Aquesta és una funcionalitat que altres solucions no tenen.
Cryptomator (https://cryptomator.org) : Ofereix interfície gràfica
Fscrypt (https://github.com/google/fscrypt) : Una altra solució similar a Gocryptfs (però només funciona sobre Ext4 ja que fa internament ús de la "Linux native filesystem encryption API", veure nota següent)

> NOTA: El sistema de fitxers Ext4 incorpora "de sèrie" (és a dir, sense la necessitat de cap capa de software extra) la capacitat de xifrar/desxifrar carpetes concretes. Per fer ús d'aquesta capacitat, anomenada oficialment "Linux native filesystem encryption API" (https://www.kernel.org/doc/html/latest/filesystems/fscrypt.html), es poden seguir les passes explicades a https://sudonull.com/post/71005-Encryption-in-EXT4-How-it-works i també a https://wiki.gentoo.org/wiki/Ext4_encryption

Suposant que fem servir el programa "Gocryptfs" (que és la més fiable avui dia), els passos bàsics per gaudir d'una carpeta xifrada al nostre sistema (i sense ser "root"!), són els següents:

gocryptfs -init /ruta/carpeta/cipher : Pas imprescindible per definir la carpeta indicada com una carpeta xifrada. Només cal fer-ho una vegada. Ens preguntarà la contrasenya que volem establir, la qual s'associarà a la clau simètrica que s'emprarà per xifrar les dades a incloure

gocryptfs /ruta/carpeta/cipher /ruta/carpeta/plain : Munta, després de preguntar interactivament la contrasenya establerta en el seu moment amb la comanda anterior, la carpeta "plain" sobre la carpeta "cipher", de forma que la primera serà la carpeta de treball "superior" (aparentment sense xifrar) mentre que la segona actuarà com a directori "inferior" (on es guardarà realment el contingut xifrat)

fusermount -u /ruta/carpeta/plain : Desmunta la carpeta "plain". D´aquesta manera, el seu contingut desxifrat desapareixerà (en estar desmuntada) mentre que romandrà xifrat a "cipher"

gocryptfs -passwd /ruta/carpeta/cipher : Permet canviar la contrasenya inicialment assignada a la carpeta "cipher" per una altra, la qual es preguntarà també interactivament

gocryptfs -info /ruta/carpeta/cipher : Mostra informació sobre els mètodes i algoritmes de xifratge aplicats a la carpeta indicada (de fet, aquesta informació es troba en format JSON a l'arxiu "/ruta/carpeta/cipher/gocryptfs.conf")

NOTA: Per saber més detalls interns sobre el funcionament de Gocryptfs, es pot consultar https://nuetzlich.net/gocryptfs/forward_mode_crypto

EXERCICIS:

1.-a) Instala a una màquina virtual qualsevol els paquets "gocryptfs" i "fuse". Si el primer no es trobés als repositoris oficials de la teva distribució (com passa, per exemple, a Fedora), executa llavors les següents comandes per tenir disponible la comanda *gocryptfs* al teu sistema:

mkdir ~/gocryptfs && cd ~/gocryptfs
wget https://github.com/rfjakob/gocryptfs/releases/download/v2.4.0/gocryptfs_v2.4.0_linux-static_amd64.tar.gz
tar -xzf gocryptfs_v2.4.0_linux-static_amd64.tar.gz

aII) Crea, a la teva carpeta personal, una carpeta anomenada "cipher" i una altra anomenada "plain". És a dir, executa les comandes següents: *mkdir ~/cipher && mkdir ~/plain*

b) Inicialitza la carpeta "cipher" per tal de fer-la servir com a carpeta xifrada amb gocryptfs. És a dir, executa la comanda següent: *gocryptfs -init ~/cipher* Estableix la "passphrase" que desitgis i copia en algun lloc la "master key" associada que se't mostrarà en pantalla (que és el que Gocryptfs utilitzarà per xifrar/desxifrar el contingut de la carpeta "~/cipher".

NOTA: Si el paquet "gocryptfs" s'ha "instal·lat" fent servir els passos indicats a l'apartat a), llavors caldrà situar-se dins de la carpeta "~/gocryptfs" i executar totes les comandes *gocryptfs* precedides dels símbols "./" (sense cometes)
NOTA: Com a resultat d'executar la comanda *gocryptfs -init* anterior, es crearan automàticament sota la carpeta "~/cipher" dos arxius, "gocryptfs.conf" (on es guarda, entre altres configuracions, la "passphrase" xifrada) i "gocryptfs.diriv" (la importància del qual es descriu a https://github.com/rfjakob/gocryptfs/issues/456) , que no caldrà modificar mai

c) Tot seguit, munta la carpeta "plain" sobre la carpeta "cipher". És a dir, executa la comanda següent: *gocryptfs ~/cipher ~/plain* (se't preguntarà la "passphrase" indicada a l'apartat anterior).

d) Crea dins de la carpeta "plain" un arxiu de text (amb un contingut qualsevol) anomenat "carta.txt". Si fas *ls ~/plain* , ¿què veus? ¿I si fas *cat ~/plain/carta.txt*? ¿I si fas *ls ~/cipher*? ¿I si fas *cat ~/cipher/**? ¿Per què?

e) Crea ara dins de la carpeta "cipher" un arxiu de text (amb un contingut qualsevol) anomenat "cartasecreta.txt". Si ara fas *ls ~/plain*, ¿què veus? ¿I si fas *ls ~/cipher*? ¿I si fas *cat ~/cipher/cartasecreta.txt*? ¿Per què?

f) Ara desmunta la carpeta "plain" executant la comanda *fusermount -u ~/plain*. Si fas *ls ~/plain*, ¿què veus? ¿I si fas *ls ~/cipher*? ¿Per què?

g) Torna a muntar la carpeta "plain" sobre la carpeta "cipher" però ara indicant directament la "master key" que vas guardar a l'apartat b), així: *gocryptfs -masterkey=xxxxx ~/cipher ~/plain* ¿Què explica el missatge que apareix en majúscules? Desmunta finalment la carpeta "plain" de nou.

h) ¿Què explica el següent apartat de la pàgina del manual d'aquesta comanda: https://github.com/rfjakob/gocryptfs/blob/master/Documentation/MANPAGE.md#fstab ?

> NOTA: Existeixen diverses "wrappers GUI" de Gocryptfs que permeten gestionar/muntar carpetes xifrades amb aquesta eina de forma gràfica. La més actualitzada a dia d'avui éshttps://github.com/moson-mo/cryptor

2.-a) Instal·la Cryptomator (https://cryptomator.org) executant la comanda *flatpak install --user org.cryptomator.Cryptomator*

> NOTA: Si la comanda anterior dóna error perquè no troba "org.cryptomator.Cryptomator" és perquè no està instal·lat el repository "FlatHub", que és d'on es descarreguen els paquets de tipus Flatpak. Per solucionar això, només caldrà executar la comanda *flatpak remote-add --user --if-not-exists flathub https://dl.flathub.org/repo/flathub.flatpakrepo* Tot seguit, ja es podrà instal·lar qualsevol programan en format Flatpak
> NOTA: Una altra opció és descarregar l'arxiu "AppImage" disponible a https://cryptomator.org/downloads ; simplement donant-li permisos d'execució (*chmod +x*) ja tindrem Cryptomator llest per funcionar al nostre sistema

b) Obre Cryptomator i, a la finestra que s'hi mostra, vés al botó "Afegir" -> "New Vault...". A l'assistent que hi apareix, indica el nom i la ruta del "vault" a crear (és a dir, de la carpeta xifrada) i la "passphrase" que hauràs de fer servir per desxifrar-la (no cal que facis servir les "opcions avançades").

c) Tria, a la finestra principal del programa, el "vault" creat a l'apartat anterior de la llista de "vaults" disponibles i clica en el botó "Desbloquejar". Tot seguit, clica en el botó "Mostra unitat". ¿Què passa? ¿Quina és la ruta de la carpeta que s'obre amb el Nautilus? Copia-hi un fitxer qualsevol allà dins

d) ¿Per a què serveix l'opció, present a la finestra principal del programa, de "Trobar fitxer xifrat"? (per veure-ho, a la finestra emergent que hi apareixerà, sel·lecciona l'arxiu creat a l'apartat anterior i clica sobre el botó "Obre")

e) Bloqueja el "vault" des de la finestra principal del programa. ¿Quin contingut és el que hi ha sota la ruta del "vault"? ¿I sota la ruta de la carpeta desbloquejada?

3.-a) CrypFS funciona aparentment de forma similar a Gocryptfs (https://www.cryfs.org/tutorial) però internament té una característica molt interessant mencionada al començament d'aquest article: (https://www.cryfs.org/howitworks) ¿Quina és?

A nivell de fitxers

Ja hem vist una manera de xifrar/desxifrar fitxers individuals: fent servir la comanda *openssl enc ...* No obstant, tot i que aquesta és una solució molt vàlida, la comanda *openssl* és molt més genèrica i versàtil que ofereix un munt de funcionalitats diverses, essent la funcionalitat concreta de xifrar/desxifrar fitxers una més però no pas la més popular. En canvi, existeixen aplicacions especialitzades purament dissenyades en realitzar només aquesta tasca, la de xifrar/desxifrar fitxers individuals, d'una forma molt més còmoda i senzilla. Concrement, en destaca la comanda **Age** (https://github.com/FiloSottile/age). A continuació es mostren alguns exemples d'ús:

* Per xifrar un fitxer (l'anomenarem "fitxer.txt", tot i que no cal que sigui un fitxer de text, pot ser de qualsevol tipus) amb una clau simètrica indicada interactivament (el resultat es guardarà en un fitxer anomenat "fitxer.age"): *age -a -p -o fitxer.age fitxer.txt*

NOTA: El paràmetre *-a* és opcional: indica que "fitxer.age" estarà codificat en Base64 en lloc de ser binari
NOTA: El paràmetre *-o* és opcional: si no s'indica el contingut xifrat anirà a la sortida estàndard (la pantalla)

* Per desxifrar un fitxer "age" amb una clau simètrica (indicada interactivament); el resultat es guardarà en un fitxer anomenat "fitxer.txt": *age -d -o fitxer.txt fitxer.age*

NOTA: El paràmetre *-o* és opcional: si no s'indica el contingut desxifrat anirà a la sortida estàndard (la pantalla)

* Per generar un parell de claus privada/pública ("privkey.txt" i, a partir d'aquesta, "pubkey.txt"", respectivament): *age-keygen -o privkey.txt && age-keygen -y -o pubkey.txt privkey.txt*

* Per xifrar un fitxer (l'anomenarem "fitxer.txt") amb una clau pública (el resultat es guardarà a un fitxer anomenat "fitxer.age"): *age -a -R pubkey.txt -o fitxer.age fitxer.txt*

NOTA: El paràmetre *-a* és opcional: indica que "fitxer.age" estarà codificat en Base64 en lloc de ser binari
NOTA: El paràmetre *-o* és opcional: si no s'indica el contingut xifrat anirà a la sortida estàndard (la pantalla)
NOTA: Es pot no indicar cap "fitxer.txt"; d'aquesta manera, el xifratge es farà del contingut que s'indiqui interactivament fins pulsar CTRL+D (o bé obtingut a través d'una canonada)

* Per desxifrar un fitxer "age" amb una clau privada (el resultat es guardarà en un fitxer anomenat "fitxer.txt"): *age -d -i privkey.txt -o fitxer.txt fitxer.age*

NOTA: El paràmetre *-o* és opcional: si no s'indica el contingut desxifrat anirà a la sortida estàndard (la pantalla)
NOTA: Es pot no indicar cap "fitxer.age"; d'aquesta manera, el desxifratge es farà del contingut que s'indiqui interactivament fins pulsar ENTER (o bé obtingut a través d'una canonada)

NOTA: La comanda Age utilitza de forma predefinida un conjunt d'algoritmes criptogràfics que no es poden canviar perquè es consideren que són prou segurs. El no deixar que l'usuari pugui triar evita que pugui, inadvertidament, rebaixar la seguretat dels xifratges emprats. Concretament, Age utilitza ChaCha20-Poly1305 per xifrar (amb claus X25519 en el mode asimètric) i Scrypt com a KDF de les "passphrases" (a més de HKDF-SHA-256 per altres derivacions de clau accesòries, no per les "passphrases"). Tot això està documentat a https://github.com/C2SP/C2SP/blob/main/age.md
NOTA: A Age per ara li "manquen" algunes característiques que podrien ser interessant, com ara un "agent" per catxejar les "passphrases" i així no haver d'escriure-les repetidament en una mateix sessió d'usuari (tal com té per exemple GPG) o el poder llegir una "passphrase" des d'un fitxer
NOTA: Altres programes similars i alternatius a Age poden ser Encpipe (https://github.com/jedisct1/encpipe) o Ccrypt (https://ccrypt.sourceforge.net)

D'altra banda, una altra funcionalitat que ofereix *openssl* és la de signar/verificar fitxers. Per això també tenim la possibilitat d'utilitzar una altra eina especialitzada només en aquesta tasca: **Minisign** (https://jedisct1.github.io/minisign). A continuació es mostren alguns exemples d'ús:

* Per generar un parell de claus pública/privada (la primera es guardarà per defecte en un arxiu anomenat "minisign.pub" ubicat dins de la carpeta actual i la segona es guardarà per defecte en un arxiu anomenat "minisign.key" ubicat dins de la carpeta "~/.minisign"): *minisign -G*

* Per signar un fitxer amb una clau privada (amb el resultat d'obtenir la signatura en forma de fitxer amb el nom de "fitxer.minisign" a la mateixa carpeta on es troba el fitxer en qüestió): *minisgin -Sm fitxer*

* Per verificar la signatura de l'arxiu indicat amb una clau pública determinada (cal estar en posessió de la signatura de l'arxiu original per tal de fer la comprovació, la qual haurà d'estar ubicada a la mateixa carpeta on es trobi aquest): *minisgin -Vm fitxer -p pubkey.txt*

Existeix una comanda anomenada **Kryptor** (https://www.kryptor.co.uk) que és capaç de xifrar/desxifrar i signar/verificar tot ell (és a dir, seria com la suma d'Age + Minisign). A continuació es mostren algunes de les seves possibilitats:

kryptor -e -p fitxer : Xifra simètricament el fitxer indicat amb la passphrase indicada interactivament
kryptor -d -p fitxer.bin : Desxifra " " " " " " " " "

kryptor -g : Crea un parell de claus pública/privada per xifrar/desxifrar (la primera es guardarà per defecte en un arxiu anomenat "encryption.public" ubicat dins de la carpeta "~/.kryptor" i la segona es guardarà per defecte en un arxiu anomenat "encryption.private" ubicat dins de la mateixa carpeta) o bé per signar/verificar (la primera es guardarà per defecte en un arxiu anomenat "signing.public" ubicat dins de la carpeta "~/.kryptor" i la segona es guardarà per defecte en un arxiu anomenat "signing.private" ubicat dins de la mateixa carpeta). Es pot indicar directament (és a dir, no interactivament) si el parell a crear és per xifrar/desxifrar o bé signar/verificar si s'afegeix el paràmetre *-e* o *-s*, respectivament.

kryptor -e -y encryption.public fitxer : Xifra el fitxer indicat amb la clau pública indicada
kryptor -d -y encryption.public fitxer.bin : Desxifra el fitxer indicat amb la clau privada pròpia (guardada en "~/.kryptor") associada a la pública indicada

kryptor -s fitxer : Signa el fitxer indicat amb la clau privada pròpia (guardada en "~/.kryptor")
kryptor -v -y signing.public fitxer : Verifica el fitxer indicat fent servir la clau pública indicada

EXERCICIS:

1.-a) Instal·la el paquet "age" des dels repositoris de la teva distribució. Tot seguit, crea un fitxer de text amb un contingut qualsevol i a continuació, xifra'l amb *age* fent servir una "passphrase" qualsevol. A continuació, desxifra el fitxer xifrat per mostrar el seu contingut (desxifrat) a pantalla

b) Genera un parell de claus asimètriques i utilitza-les per xifrar amb *age* el mateix fitxer de l'apartat anterior i tot seguit desxifrar-lo mostrant el seu contingut (desxifrat) a pantalla

c) ¿Per a què serveixen aquestes webs: https://rage-encry.pt o https://age-wasm.ey.r.appspot.com?

2.-a) Instal·la el paquet "minisign" des dels repositoris de la teva distribució (si, com és el cas d'Ubuntu, no hi és, segueix llavors les instruccions indicades a la "Nota" següent). Tot seguit, genera un parell de claus asimètriques (se't preguntarà una "passphrase" per protegir la clau privada) i crea un fitxer de text amb un contingut qualsevol.

NOTA: Per instal·lar "minisign" a Ubuntu executa les següents comandes:
wget https://github.com/jedisct1/minisign/releases/download/0.11/minisign-0.11-linux.tar.gz
tar -xzf minisign-0.11-linux.tar.gz
cd minisign-linux/x86_64

chmod +x minisign
sudo cp minisign /usr/bin (aquest pas és opcional però convenient per poder executar *minisign* des de qualsevol lloc)
cd ~

b) Signa amb *minisign* (fent servir la teva clau privada) el fitxer anterior i tot seguit verifica aquest fitxer signat (fent servir la teva clau pública).

3.-a) Instal·la Kryptor executant les següents comandes (no es troba encara a cap repositori de les distribucions importants):

mkdir ~/kryptor && cd ~/kryptor
wget https://github.com/samuel-lucas6/Kryptor/releases/latest/download/kryptor-linux-x64.zip
unzip kryptor-linux-x64.zip
chmod +x kryptor
sudo cp kryptor /usr/bin (aquest pas és opcional però convenient per poder executar *kryptor* des de qualsevol lloc)
cd ~

b) Crea un fitxer de text amb un contingut qualsevol i a continuació, xifra'l amb *kryptor* de forma simètrica fent servir una "passphrase" qualsevol. A continuació, desxifra el fitxer generat al pas anterior per mostrar el seu contingut (desxifrat) a pantalla

NOTA: Es pot indicar la passphrase de forma no interactiva amb la sintaxi *-p:"xxxx"* (on *xxxx* representa la passphrase)

c) Genera un parell de claus asimètriques (que estaran protegides per una "passphrase") i utilitza-les per xifrar amb *kryptor* el mateix fitxer de l'apartat anterior però de forma asimètrica, i tot seguit desxifrar-lo mostrant el seu contingut (desxifrat) a pantalla

d) Genera un altre parell de claus asimètriques (que estaran protegides per una "passphrase") i utilitza-les per signar amb *kryptor* (fent servir la teva clau privada) el fitxer anterior i tot seguit verifica aquest fitxer signat (fent servir la teva clau pública).

e) ¿Per a què serveix el paràmetre *-o*? ¿I *-n*? (pots llegir https://www.kryptor.co.uk/tutorial/encryption-options per saber-ho) ¿I *-k* (pots llegir https://www.kryptor.co.uk/tutorial/encrypting-files-for-yourself)? ¿I *-m* i *-r* (pots llegir https://www.kryptor.co.uk/tutorial/key-pair-options)?