GPG

Tecnologies de (des)xifratge: TLS vs GPG

Existeixen diferents tecnologies que ens permeten xifrar/desxifrar informació (és a dir, afegir confidencialitat a un missatge) i, addicionalment, algunes també ofereixen possibilitats d'afegir-hi autentificació i/o integritat. Les dues tecnologies més conegudes en aquest sentit són TLS i GPG. Totes dues són molt diferents entre sí (i anirem estudiant aquestes diferències poc a poc) però "a grosso modo", les dues diferències més importants que cal saber d'entrada són:

*TLS és un protocol híbrid (és a dir, que utilitza tant criptografia simètrica com asimètrica) el qual es basa en una infraestructura d'autoritats centrals de certificació reconegudes pel nostre sistema que possibiliten que aquest pugui confiar en l'autenticitat de les claus públiques dels sistemes remots amb els quals es vulgui comunicar. En canvi, GPG es basa en la "Web of trust" per aconseguir el mateix objectiu.

*TLS se centra en el xifratge del canal per on es transfereix la informació mentre que GPG se centra en el xifratge de la informació en sí (la qual, doncs, podrà viatjar a través de qualsevol canal insegur). Això vol dir que si es fa servir TLS en una connexió, es xifra tot el que hi passa "dins" d'aquesta connexió (comandes de petició/resposta, contrasenyes d´accés, capçaleres dels missatges, els propis missatges, etc). En canvi, amb GPG només es xifra les dades que haguem decidit, la qual cosa fa que, a la pràctica, sigui una tecnologia més habitual a l'hora d'emmagatzemar aquestes dades de forma "estàtica" (per exemple en un disc dur) i no tant per transmetre/rebre-les "al vol" a través d'una connexió, com seria el cas de TLS.

En qualsevol cas, en aquest document estudiarem la tecnologia GPG (Gnu Privacy Guard). Per obtenir el software necessari haurem d'instal.lar (tant a Ubuntu com a Fedora) el paquet anomenat "gnupg2".

(Des)Xifratge simètric amb GPG

Recordem que en el (des)xifratge simètric s´utilitza la mateixa clau per xifrar i per desxifrar. Per conèixer els algoritmes de (des)xifratge simètrics disponibles en la suite GPG podem observar la sortida de la comanda *gpg2 --version* . A partir d'aquí:

*Per xifrar el fitxer "document.txt" (en mode verbós) : *gpg2 -v -c document.txt* Després de demanar interactivament una "passphrase" (dues vegades) que serà utilitzada per generar la clau amb la què se xifrarà el fitxer, finalment s'obtindrà un nou fitxer anomenat "document.txt.gpg" amb el seu contingut xifrat mitjançant l'algoritme simètric per defecte (indicat a pantalla)

NOTA: Si es vol que "document.txt.gpg" tingui un format de text en comptes de binari (molt recomanable!), s'ha d'afegir l'opció *-a* . En aquest cas, llavors, el fitxer generat es dirà per defecte "document.txt.asc"
NOTA: Si es vol que el nom de "document.txt.gpg" sigui un altre, s'ha d'afegir l'opció *-o unaltrenom.txt.gpg*
NOTA: Si es vol utilitzar un altre algoritme de xifratge diferent al per defecte, s'ha d'afegir l'opció *--cipher-algo nomAlgo*

*Per desxifrar el fitxer "document.txt.gpg" : *gpg2 -d document.txt.gpg* El resultat desxifrat es veurà a pantalla. Si es vol guardar en un fitxer, s'ha d'afegir l'opció *-o sortida.txt*

NOTA: Si es vol canviar l'algoritme per defecte (o qualsevol altra opció de configuració) es pot crear un fitxer de texte anomenat "~/.gnupg/gpg.conf" i afegir-hi les opcions de configuració que desitjem (una per línia) amb la sintaxi *nomLlargOpcio valor* . En aquest sentit, per canviar l'algoritme per defecte podríem escriure la línia "cipher-algo AES256" , per exemple. Aquest fitxer també admet línies comentades si comencen pel símbol #.

NOTA: Si es volen encriptar simètricament molts fitxers de cop (dins d'un bucle for...in, per exemple),caldrà afegir els paràmetres *--batch* i *--passphrase-fd 0* i introduïr la contrasenya via stdin -amb un echo entubat prèviament, per exemple-)

NOTA: Si es vol eliminar la clau de la catxé un cop s'ha desxifrat un missatge per a què la torni a demanar en desxifratges posteriors, cal executar gpg2 --no-symkey-cache

(Des)Xifratge asimètric amb GPG:

Recordem que en el (des)xifratge asimètric s'utilitzen dues claus: una "privada" i una "pública". Com diu el seu nom, la clau privada de cadascú haurà de ser secreta, però la clau pública la coneixerà tothom. Concretament, si suposem que "A" és el remitent d'un missatge i "B" és el destinatari:

*"A" haurà de fer servir la clau pública de "B" per xifrar un missatge de només "B" pugui desxifrar

*Per desxifrar-lo, "B" haurà de fer servir la seva pròpia clau privada (per això només ell serà capaç)

Per tant, podem concloure que la clau pública serveix per xifrar, i la clau privada per desxifrar allò que va ser xifrat per la pública corresponent. Es fàcil veure d´aquest esquema que la nostra clau pública ha de ser difosa a tots els usuaris dels quals volguem rebre correu xifrat, però la nostra clau privada ha de ser només controlada per nosaltres!

NOTA: La manera de difondre la nostra clau pública (que no és més que un simple fitxer) és molt variada: pot ser entregada en "mà" en un pen-drive, pot ser penjada en algun servidor web de la nostra propietat, pot ser adjuntada en algun mail que enviem, pot estar guardarda en un servidor de claus especialment pensat per això (ja en parlarem més endavant),etc.

Generació del parell de claus:

El primer que hem de fer és crear un parell de claus privada/pública amb la comanda: *gpg2 --full-gen-key* (també existeix la comanda *gpg2 --gen-key* que fa el mateix però aquesta darrera no pregunta a l'usuari tantes dades com la primera, la qual sí ens permet personalitzar al màxim les característiques de les claus generades).

La comanda *gpg2 --full-gen-key* ens preguntarà el tipus d'algoritme utilitzat per generar la clau (hi ha diverses opcions: RSA o ElGamal/DSA -que ens permetran tant xifrar com signar- o DSA o RSA -només per signar-, el valor sel.leccionat per defecte -RSA- ja ens anirà bé), la llargària de la clau (el valor per defecte ja ens anirà bé: a més llargària més segura però més costosa computacionalment), el temps d´expiració (el valor per defecte -"mai"- ja ens anirà bé: si escollísim un altre, quan expirés el temps hauríem de tornar a difondre a tots els nostres contactes la nostra nova clau pública...un rotllo), una sèrie de dades personals que serviran per identificar aquest parell de claus respecte qualsevol altre que tinguem integrat al nostre sistema (tindrem en compte sobretot d´escriure el nostre nom i una determinada direcció de correu perquè són aquests valors els que s'utilitzaran com identificadors), i finalment ens preguntarà una "passphrase" opcional: si la indiquem, aquesta "passphrase" l´haurem d´escriure llavors cada cop que volguem utilitzar la nostra clau privada: és una mesura de seguretat per evitar que si algú s´apropia d´aquesta, la pugui fer servir.

Un cop generat el parell de claus, podrem veure que GPG li ha assignat un "fingerprint" (amb un aspecte semblant a F2C7 C647 717B 0210 66E1 5EBA 92D8 9207 EE74 D48D) i un "key-id" (amb un aspecte semblant a EE74D48D per exemple -de fet, són els darrers 8 caràcters del "fingerprint"-). Depenent del contexte haurem de fer servir alguna d'aquests valors (o el nom i/o l'email introduït a les preguntes) per identificar el parell de claus amb el què volguem treballar.

Importació de claus públiques de tercers:

El primer que hem fer per començar a enviar arxius xifrats a algú és obtenir la clau pública del destinatari en qüestió (que no deixa de ser un simple fitxeret de text). La manera pot ser qualsevol, tal com ja hem comentat en una nota anterior: se'ns pot haver entregat la clau pública en "mà" dins un "pendrive", se'ns pot haver enviat adjuntada a través del correu electrònic, la podem haver descarregat d'alguna pàgina web propietat del destinatari, la podem haver descarregat d'un servidor de claus especialment pensat per això (ja en parlarem més endavant)..

1.- Suposant que aquesta clau pública del destinatari (diguem que és "B") ja és al nostre poder i suposant que és un fitxer anomenat "clauPubB.asc", per poder-la fer servir l'haurem d'"importar" en el nostre sistema (en el que s'anomena "anell de claus"). Això es fa amb la comanda: *gpg2 --import ruta/clauPubB.asc* .

NOTA: No és tan habitual, però també és possible importar una clau privada prèviament exportada (per exemple perquè canviem de màquina). Es faria amb la mateixa comanda: *gpg2 --import ruta/clauPrivA.asc*

2.-Per veure si s´ha importat bé, ho comprovarem amb la comanda: *gpg2 -k* Aquesta comanda mostra la llista de claus públiques que tenim integrades a l'anell de claus

NOTA: També es pot indicar l'identificador d'una clau concreta per obtenir la informació només d'aquesta, així: *gpg2 -k unKeyID* (on "unKeyID" pot ser tant el nostre nom que vam introduïr a la creació del parell com la direcció de correu com el propi keyID)

NOTA: També podríem veure la llista de claus privades integrades en el nostre anell de claus amb *gpg2 -K [unKeyID]*

NOTA: En el cas de voler eliminar una clau pública ja importada en el nostre anell de claus , cal fer *gpg2 --delete-keys unKeyID* En el cas de què aquesta clau pública estigués associada a una clau privada (és a dir, que fos la nostra), caldrà

eliminar llavors les dues claus; això es pot fer en dues passes, així: *gpg2 --delete-secret-keys unKeyID && gpg2 –delete-keys unKeyID* o bé en una passa, així: *gpg2 --delete-secret-and-public-key unKeyID)*

3.-Un cop ja tenim la clau pública importada, ja podrem xifrar el missatge que enviarem posteriorment a aquesta persona (o persones) en concret. Suposant que el missatge està escrit en un fitxer anomenat "document.txt", podem executar: *gpg2 -a -r unKeyID [-r unAltreKeyID ...] -o documentxifrat.txt -e document.txt*, on:

*El paràmetre -a (de "--armor") indica que es xifrarà en mode ASCII (recomanable per comoditat) en comptes de mode binari.
*El paràmetre -r (de "--recipient") indica el keyID de la (o les perquè es poden indicar més d'una) clau/s públiques que volem utilitzar per xifrar el missatge.
*El paràmetre -o (de "--output") indica el nom que tindrà el document xifrat (si no s'indica s'anomenarà "document.txt.asc"
*El paràmetre -e (de "--encrypt") indica l'acció a realitzar sobre "document.txt" (xifrar-lo)

4.-Si observem el contingut del fitxer "documentxifrat.txt", veurem que no deixa de ser un reguitzell de símbols ASCII sense sentit. Bé, ara només caldria enviar aquest "documentxifrat.txt" d'alguna manera (via "pendrive", correu electrònic, etc) a la màquina del destinatari adient.

5.-En arribar a la màquina del destinatari, l´únic que aquest hauria de fer per desxifrar el missatge és executar la comanda *gpg2 -d missatgexifrat.txt -o missatgeoriginal.txt*.

NOTA: Si sou el remitent i proveu la comanda anterior, veureu que ni tan sols vosaltres no podreu desxifrar el vostre propi missatge (a no ser què hagueu utilitzat la vostra pròpia clau pública quan el vau encriptar): si vau escriure una altra keyID diferent, necessitaríeu la clau privada seva -i saber-ne el "passphrase", si n'hi ha-.

Exportació de la nostra clau pública:

En el cas de que fossim nosaltres els destinataris, per a què algú ens envïi un missatge xifrat que només poguem dexifrar nosaltres haurem de publicar prèviament la nostra clau pública d'alguna manera (pendrive, web, correu, etc). No obstant, per poder fer això primer hem d'"extreure" aquesta clau pública de l'anell de claus del nostre sistema per tal d'obtenir-ne un fitxeret (preferiblement de text fent servir el paràmetre *-a*) que representarà aquesta clau i que serà el que publicarem. Això s'aconsegueix amb la comanda: *gpg2 -a --export -o clauPubA.asc unKeyID*

NOTA: No és tan habitual, però també és possible exportar la nostra clau privada (per exemple perquè canviem de màquina). Això es faria amb la comanda: *gpg2 -a --export-secret-keys -o clauPrivA.asc unKeyID*

Signar/Verificar amb GPG:

A més de per xifrar/desxifrar, les claus RSA pública/privada que hem fet servir fins ara també es poden utilitzar per signar/verificar. Signar un missatge vol dir afegir-li una marca que asseguri que aquest missatge ha sigut escrit realment per qui diu que l´ha escrit (i sense repudi, a no ser que la seva clau privada s'hagi compromés), i que no s´ha modificat el seu contingut original. De poc serviria xifrar un missatge si resulta que qui ens ho envia no és qui diu que és, o bé si ha hagut algú al mig que ha modificat el seu contingut d´alguna manera. Concretament, si suposem que "A" és el remitent d'un missatge i "B" és el destinatari:

*"A" haurà de fer servir la seva pròpia clau privada per signar el missatge (només ell serà capaç)
*"B" haurà de fer servir la clau pública de "A" per verificar aquesta signatura

Per tant, podem concloure que la clau privada serveix per signar, i la clau pública de l´usuari remitent serveix per comprovar que la signatura és vàlida. D´aquesta manera podríem tenir missatges signats (amb la clau privada del remitent), missatges xifrats (amb la clau pública del destinatari) ó signats i xifrats (amb la combinació de les dues anteriors).

Per signar un fitxer (l'anomenarem "document.txt") només cal executar la comanda: *gpg2 -a -s document.txt* (si la clau privada tingués "passphrase", aquesta comanda ens la preguntarà interactivament). Això generarà un altre document (en format ASCII) anomenat "document.txt.asc" (a no ser que haguem afegit també el paràmetre *-o nomnoudocument.txt*) que representa el document original signat. Aquesta manera de signar, no obstant, ofusca el

contingut del document; si volem generar un fitxer "document.txt.asc" on aparegui clarament el contingut original i tot seguit, la signatura diferenciada, cal fer llavors *gpg2 -a --clearsign document.txt* (aquesta opció, però, només funciona en documents de tipus text). Una altra opció seria guardar només la signatura en un fitxer apart (també anomenat "document.txt.asc") amb la comanda *gpg2 -a -s -b document.txt* (cal dir, però, que llavors necessitarem sempre el document original al costat per poder usar la signatura).

> NOTA: Si el remitent tingués més d'una clau privada, es pot escollir quina fer servir per signar afegint a qualsevol de les comandes indicades al paràgraf anterior el paràmetre *-u unKeyID*

En qualsevol cas, el destinatari del document, per tal de verificar la signatura del remitent, haurà de tenir prèviament importada la clau pública d'aquest i llavors executar: *gpg2 --verify document.txt.asc* (en el cas d'haver generat la signatura en un fitxer apart amb -b, en executar la comanda anterior hem de tenir a la mateixa carpeta on es troba "document.txt.asc" el document original, "document.txt" o si no, caldrà indicar la ruta d'aquest com a darrer paràmetre).

Si el que volem és xifrar i signar un missatge tot a l'hora, només cal que afegim els paràmetres adients de cada acció en una mateixa comanda (per exemple: *gpg2 -s -a -r unKeyID -e document.txt*) . En aquest cas, el destinatari no caldrà que verifiqui explícitament la signatura amb *--verify* perquè en intentar desxifrar-lo ja es fa la verificació automàticament.

Servidors de claus:

Hi ha una altra manera d´obtenir claus sense que necessitem tenir contacte directe amb l'altre extrem (pendrive, correus) i una mica més seriosa que anar-les buscant pel Google: accedir a un servidor públic repositori de claus, com per exemple "keyserver.pgp.com" o "keyserver.ubuntu.com" o "gpg.mit.edu" (el servidor concret no importa perquè el seu contingut es replica entre ells).

Per descarregar una clau pública sabent el seu "key-id" (els últims vuit caràcters del "fingerprint") només cal executar la comanda *gpg2 --keyserver nomservidor --recv-keys elkey-id* . Un cop descarregada, la podrem importar al nostre anells de claus com qualsevol altra clau.

Si no es coneix el "key-id", es poden buscar claus segons la direcció de correu o el nom associats a ella, així: *gpg2 –keyserver nomservidor –search-keys una@dir.decorreu*

També es pot fer *gpg2 --keyserver nomservidor --refresh-keys* per actualitzar les claus que ja es tenen en local per si alguna ha canviat.

També podem pujar la nostra pròpia clau pública a un servidor amb *gpg2 –keyserver nomservidor –send-keys elkey-id*

"Web of trust" i validació de claus:

Sent una mica paranoics, alguns d´aquests mètodes de difusió de la nostra clau pública poden ser “hackejats”: el servidor web pot ser suplantat, els missatges de correu també, etc. Bé, d´entrada cal dir que si la nostra clau pública és compromesa, l´”únic” que passarà és que els missatges xifrats que ens envïin no els podrem llegir però... ¿com podem saber realment que la clau pública de B que nosaltres tenim és realment la clau verdadera i ningú ens ha enganyat? En el món GPG hi ha dues maneres: utilitzant l´estàndar GnuPG (més “amateur”) o bé l´estàndar S/MIME (més “empresarial” i “profesional”).

Amb el mètode GnuPG s´utilitzen les anomenades xarxes de confiança (o "web of trust"). Persones que saben de forma certa que una determinada clau pública és vertadera, la poden signar amb la seva pròpia signatura privada. Quantes més signatures tingui una clau pública, més confiança hi haurà en què aquesta clau pública és efectivament la que diu que és. Amb el mètode S/MIME no hi ha cap xarxa de “col.legues” que signen una clau: el que hi ha és una reconeguda entitat externa independent (com Thawte ó Verisign), que és l´encarregada de signar ella les claus i certificar que són vertaderes (en aquest sentit aquest mètode seria similar al que utilitza TLS).

Si fem servir el mètode GnuPG i tenim importada la clau pública d'algú de la qual confiem plenament que és la clau pública de qui diu que és (perquè ens la ha dit per telèfon, per exemple), tal com hem dit, podriem "certificar-ho" signant aquesta clau pública amb la nostra clau privada de forma que donem "fe". Això es pot fer executant la comanda

gpg2 --edit-key unKeyID Apareixerà una consola interna on executarem la comanda “trust”, que ens servirà per valorar del 0 al 5 si confiem o no en l´autenticitat d´aquesta clau i, a continuació, la comanda “sign” per signar la clau en qüestió amb la nostra clau privada (acció que, per cert, podríem fer també directament amb la comanda *gpg2 --sign-key unKeyID*), i finalment la comanda "quit" (o CTRL+D) per sortir de la consola interna. Un cop fet això, aquesta clau pública l´hauríem de retornar al propietari original per a que aquest la torni a redistribuir (per exemple, en un servidor de claus).

> NOTA: Altres comandes interessants de la consola interna són per exemple "fpr" (per veure el "fingerprint") o "check" (per comprovar que s'hagi realitzat la signatura correctament...aquesta darrera seria equivalent a la comanda *gpg2 --check-signs elkey-id*), entre d'altres

> NOTA: Precisament la manera d'evitar els "warnings" de tipus "There is no indication that the signature belongs to the owner" és signar-la com s'acaba d'explicar.

Revocació de claus:

Si per alguna raó volguéssim en algun moment rescindir la validesa de la nostra clau pública (que, recordem, en principi no té termini d´expiració però que, per exemple, sospiteu que algú us ha robat la vostra clau privada), la manera seria generant (en qualsevol moment) un fitxer especial anomenat "certificat de revocació" amb la comanda *gpg2 -o certrevoc.asc --gen-revoke unKeyID* i, en el moment de realment rescindir la validesa de la nostra clau pública, importar aquest certificat al nostre anell de claus amb *gpg2 --import certrevoc.asc* Igualment, és bona idea enviar aquest certificat de revocació als amics per a què facin el mateix en seu propi anell de claus.

> NOTA: Note that you cannot create this revocation certificate after you've lost your private key! So it is recommended to create one at the same time that you create the key pair, then keep it in a really safe place.

Si la clau, ja revocada, la vam pujar a un servidor de claus, cal notificar-li que aquesta clau ja no és vàlida. Això es pot fer amb la comanda ja coneguda: *gpg2 --keyserver keyserver.ubuntu.com --send-keys unKeyID* (on "unKeyID" representa l'identificador de la clau en qüestió)