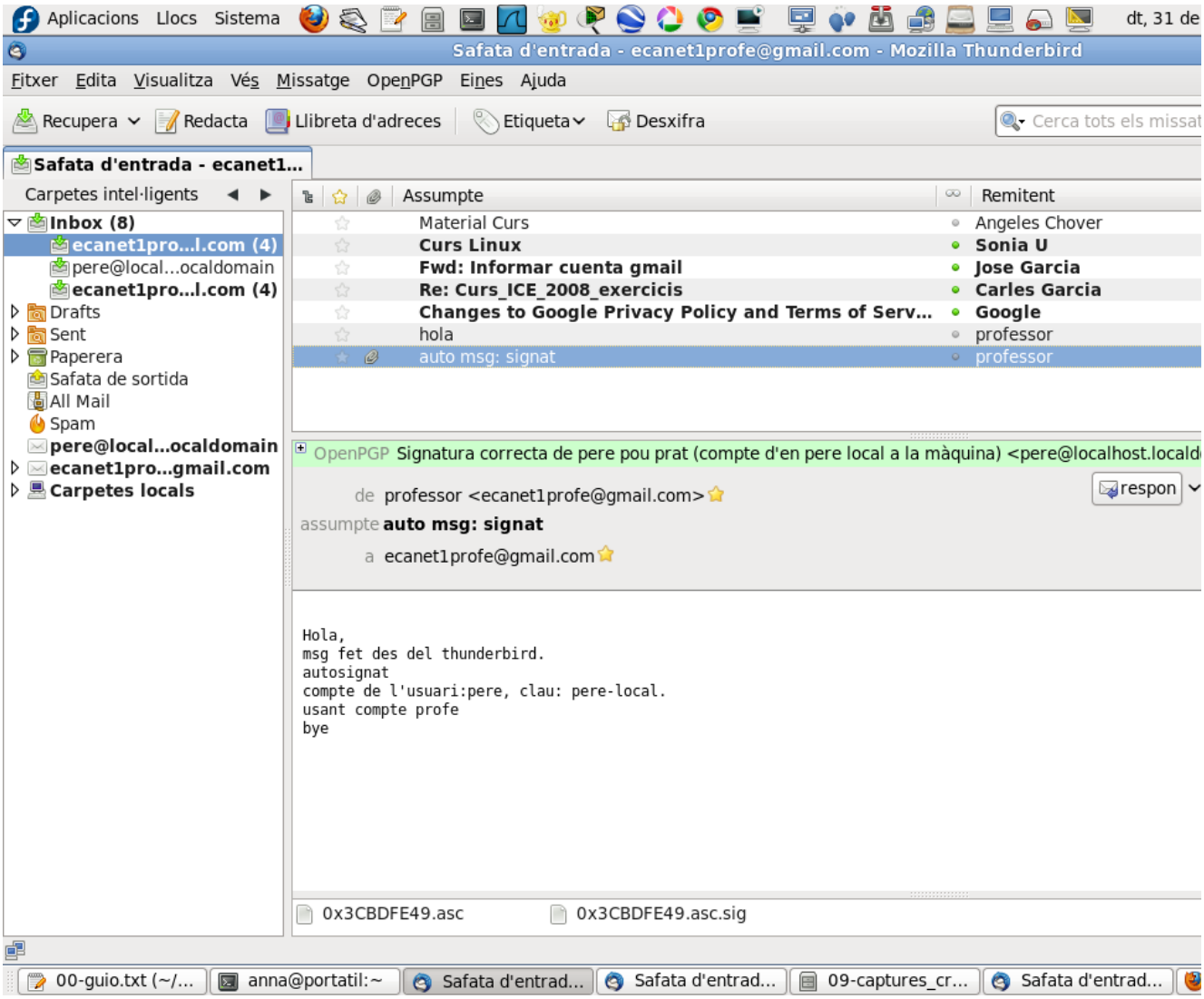

A2 Seguretat: conceptes generals

En aquest apartat s'intentarà fer un repàs general als principals conceptes de seguretat en global, per tenir una idea general de conceptes com certificats, claus, PKI, etc.

Criptografia

Una primera classificació de seguretat és el tipus de criptografia a utilitzar:

- Criptografia de clau simètrica.
- Criptografia de clau asimètrica.
- Criptografia híbrida.

Els algorismes de xifrat simètric són de menor cost computacional en comparació amb els asimètrics. És a dir, xifrar i desxifrar amb clau simètrica és més ràpid, requereix claus més petites i menys cost computacional de l'ordinador. La robustesa entre algorismes simètrics i asimètrics és equivalent, és a dir, no es pot dir que uns són més 'bons', que xifren 'més bé', 'més segur' que els altres.

Criptografia de clau simètrica: s'anomena criptografia de clau simètrica quan s'utilitza la mateixa clau per encriptar que per desencriptar. La clau o secret ha de ser compartida per l'emissor i pel receptor, un la necessita per encriptar i l'altre per desencriptar. Les

claus d'aquest tipus són fàcils de gestionar en el sentit de que permeten una seguretat molt robusta amb claus 'petites' (pocs bits comparades amb les asimètriques) i un cost algorísmic menor. El problema del xifrat simètric és que cal que aquesta clau o secret sigui compartit entre els dos interlocutors, per tant continua existint el problema de com compartir aquest secret de forma segura. Proporciona un bon rendiment per xifrar fluxos continus de dades, per xifrar comunicacions TCP. Són algorismes de clau simètrica DES, 3DES, AES, i IDEA entre altres.

Criptografia de clau asimètrica: S'anomena criptografia de clau asimètrica al fet d'utilitzar dues claus diferents una anomenada *clau privada* i l'altre anomenada *clau pública*. Les dues claus s'utilitzen "com una parella" per aconseguir les finalitats de xifrar i signar. No s'ha d'interpretar com que la una només xifra i l'altra només desxifra sinó com que allò que s'ha 'fet' amb una es pot 'desfer' amb l'altra. Cada interlocutor que intervé en la comunicació ha de disposar de la seva parella de claus privada/pública. La clau privada s'ha de mantenir privada, fora de l'abast de qualsevol altre. La clau pública s'ha de fer arribar als altres interlocutors. Així per exemple per xifrar es xifra amb la clau pública del destinatari i el destinatari desxifra amb la seva clau privada. Per signar digitalment es signa amb la clau privada de l'emissor i el receptor verifica la firma amb la clau pública del emissor. Aquest model en el que s'utilitzen dues claus permet salvar l'inconvenient de "compartir" un secret que té la criptografia de clau simètrica. Ara ja no hi ha cap secret a compartir, tothom té una clau privada que no ha de compartir amb ningú i una clau pública que ha de fer arribar a tothom però que no és 'un secret', precisament per això es pot compartir. Aquest model és bo per xifrar continguts concrets com per exemple emails però massa costos de càlcul per poder xifrar comunicacions de dades continues com una sessió TCP. Són algorismes de clau asimètrica RSA, DSA i ElGamal entre altres.

En el proper apartat es descriu clarament com s'utilitzen les parells de clau privada/pública per xifrar i signar per una banda i per desxifrar i verificar per l'altra.

Criptografia híbrida: tal i com s'ha exposat la criptografia simètrica és més ràpida i menys costosa computacionalment que la criptografia asimètrica, però té l'inconvenient que cal que els dos interlocutors coneguin la clau simètrica. Un mecanisme molt usat per a comunicacions segures intenta aprofitar els avantatges dels dos models anteriors. Així per exemple tant SSH com SSL combinen els dos tipus de criptografia en un tercer model que s'anomena 'híbrida'. El model asimètric (basat en els certificats digitals de claus privada/pública) permet que els dos interlocutors puguin iniciar una comunicació segura sense necessitat de cap secret compartit. Un cop establerta la comunicació segura els dos interlocutors poden intercanviar-se 'un secret compartit' i iniciar una comunicació xifrada simètrica que proporciona un rendiment millor per al xifrat d'un diàleg continuo TCP.

Claus

Una altra classificació que podem realitzar és basada en les claus:

- **Privada**: una clau privada és una clau usualment del tipus RSA o DSA d'un número de bits variable (actualment s'utilitzen valors de 1024, 2048 o 4094 bits). Aquesta clau s'ha de mantenir fora de l'abast de tothom excepte del seu propietari. Un mecanisme extra de seguretat és xifrar el fitxer de la clau amb un password (un xifrat simètric ja que el mateix password serveix per xifrar i per desxifrar).
- **Pública**: les claus públiques van aparellades a les claus privades formen una parella 'indissoluble'. A partir de la clau privada es genera la clau pública (o les dues de cop). De fet, però, les claus públiques no són útils com a tals sinó quan es transformen en certificats. Cal tenir la parella privada/pública per poder implementar seguretat amb certificats digitals.
- **Certificats**: Les claus públiques es transformen en 'certificats digitals' que són els que realment s'utilitzen. És a dir, quan parlem de 'clau pública' en general ens referim al 'certificat digital'. Un certificat digital no és més que una clau pública 'avalada' per una entitat. Igual que un certificat de naixement és un paper amb un segell oficial de l'oficina del padró, o un certificat de pagament és un paper amb un segell oficial del banc, un certificat digital és una clau pública 'signada' per una entitat que 'avala' com a vàlid el certificat. Aquest certificat o clau pública és el que s'ha de fer arribar als altres interlocutors, que hi confiaran o no en funció de qui avali el certificat. El format dels certificats actuals és **X509**.

La finalitat d'un certificat es demostrar que el posseïdor del certificat és qui diu ser i té dret a fer allò per al què s'ha expedit el certificat. Així un

certificat vàlid per a fer de “emailProtection” identifica a l'usuari i permet xifrar i signar contingut email. Un certificat vàlid per a “serverAuth” identifica al servidor i li permet actuar com a servidor segur usant SSL.

Perquè no s'utilitzen simplement les claus públiques en lloc dels certificats? Perquè la clau pública d'en “pere” no sabem si realment és d'en “pere” o d'algú altre que s'hi fa passar. En canvi un certificat d'en “pere” és una clau pública on algú amb 'autoritat' a avalat que realment és en “pere”.

> Un certificat és associar la clau pública d'algú amb la seva identitat. El certificat es firma per una CA o per un mateix (auto-certificats). Per firmar es fa un hash i se li aplica la clau privada de la CA (generalment algorisme RSA).

Formats dels fitxers

- **PEM**: aquest és el format més usual i el format per defecte d'eines com *openssl*. Conté les claus (privada o pública) en format *ascii*. De fet el contingut binari de les claus (format DER) es codifica en *base64* per generar un text *ascii* imprimible i se li afegeixen capçaleres i peus identificadors del tipus de clau.
- **DER**: contenen les claus en format binari.
- **xifrat**: els fitxers de clau privada s'acostumen a xifrar amb un password com a mesura extra de protecció, ja que si la clau privada cau en mans d'altres tindran accés a totes les dades privades. Els formats més usuals amb els que es xifra el fitxer de la clau privada són: DES, 3DES (tres-DES és el format actual per defecte) i IDEA.
- **altres** sovint es troben fitxers amb claus que tenen les extensions *.cert*, *.crt*, *.crs* o fins i tot *.key*. Totes questes extensions no signifiquen res ni descriuen realment el format del fitxer sinó que són una mala manera d'intentar descriure per a què serveix el certificat. De fet segurament el certificat és un PEM. Això no s'ha de fer, és com posar en un fitxer de text no una extensió ”.txt” o ”.odt” segons sigui el seu format sinó .document o .informe o .carta tot intentant en l'extensió dir què conté en lloc de descriure el 'format' del què conté (text ascii pur en un ”.txt” i un document del *openoffice* en un ”.odt”).

Formats de les claus i certificats

- **RSA** Un tipus de claus, siguin públiques o privades, són les claus RSA. Actualment són les claus usades per defecte en *openssl*. El nom prové dels seus autors “Ron Rivest, Adi Shamir i Leonard Adleman”. Aquest algorisme es vàlid per xifrar i signar contingut digital.
- **DSA** Aquest és un altre format de claus privades i públiques. El nom significa “Digital Signature Algorithm”. És un estàndard de criptografia del govern dels Estats Units usat per signar digitalment.
- **X509**: els certificats digitals s'anomenen sovint 'clau pública' però no són el mateix. Un certificat és la clau pública més la informació identificativa del propietàri i l'emissor, més la firma de l'entitat que ha emès el certificat, que l'avala. De fet en un certificat digital hi pot haver-hi a més a més el certificat de l'entitat emissora o de vàries entitats emissores (les que formen la cadena de confiança).
- **PKCS#12**: alguns clients de correus com per exemple *thunderbird* utilitzen un tipus de fitxers de claus diferents als fitxers *.pem* usats normalment tant per a claus privades, públiques i certificats. De fet existeixen formats diferents de fitxers per a propòsits diferents. El format *PKCS#12* és el format més usual amb que els clients gràfics emmagatzemen els anells de claus. No contenen només un certificat sinó que ho poden contenir tot (la clau privada i tots aquells certificats que facin falta). És una mena de contenidor per a tot.
- **PKCS**: s'anomena “Public-Key Cryptography Standards” a un conjunt

d'estàndards de seguretat publicats per *RSA Security*, l'empresa privada que marca el camí en els temes de seguretat. Així per exemple alguns dels seus estàndard més coneguts són:

- PKCS#1 és l'estàndard que defineix el format de les claus de tipus RSA (tant privades com públiques).
- PKCS#7 es l'estàndard usat per definir S/MIME.
- PKCS#8 és el nou estàndard per a la gestió de claus privades.
- PKCS#10 és l'estàndard que defineix el format de les peticions de certificació o *request*.

Creació de claus i certificats

Cada usuari es pot crear la seva pròpia clau privada. Per crear la clau pública disposa de dues opcions:

- Crear una clau 'avalada' per ell mateix, que s'anomenen 'certificats auto-signats'. Els altres interlocutors confiaran amb aquest certificat tant com confiïn amb el seu emissor, és a dir, amb l'usuari.
- Crear una petició de certificat (clau pública + dades descriptives de l'emissor) i demanar a una entitat de certificació (anomenades CA 'Certification Authority') que en generi un certificat 'avalat' per aquesta CA. Els altres usuaris confiaran amb aquest certificat en la mateixa mesura en la que confien amb l'entitat de certificació CA.

Models de confiança

Per implementar comunicacions segures, siguin comunicacions tipus SSH o SSL/TLS, o sigui correu xifrat i signat cal establir mecanismes de confiança amb els certificats. Igual que passa amb els certificats de paper són vàlids segons la confiança que tenen els altres amb l'emissor del certificat.

Suposem per exemple que un alumne es posa malalt i no va a classe. L'endemà porta al professor un paper escrit per ell a mà i signat per ell mateix on 'certifica que estava malalt'. Es tracta d'un certificat auto-signat i evidentment el professor té molt poca confiança amb la validesa d'aquest certificat.

Un altre alumne també va faltar a classe i ha portat també un certificat fet a mà 'excusant-se perquè estava malalt' signat per la seva mare. Ara el certificat no és auto-signat sinó que està avalat per una entitat externa de certificació (la mare). El professor es creurà el paper o no en funció de la confiança que tingui en el paperet de 'la mare'. Quan naveguem per intenet i accedim a segons quins continguts el navegador genera una excepció de seguretat i ens demana si creure's el certificat (descarregar-lo i incorporrar-lo a la llistat d'entitats de certificació) o abortar. Quan es troba un certificat emès per algú desconegut (com 'la mare') cal decidir si s'hi confia o no s'hi confia al teu propi risc.

Resulta que un altre alumne també va faltar a classe, és el fill del director de l'escola i porta un justificant emès per el seu pare. Com que el professor coneix de sobres la firma del director dona el certificat per vàlid. És a dir, el professor té en la seva llistat d'entitats en les que confia al 'Senyor Director' i la seva firma.

Però voleu dir que amb el que no s'ha de confiar no és amb el professor? Si li falten tots els alumnes!

I per finalitzar resulta que un altre alumne que també va faltar ha vingut avui a classe portant un justificant mèdic de l'hospital (amb el segell i tota la pesca). En aquest cas el profesor també confia amb el certificat perquè porta un segell 'oficial'. Això passa també amb els nevegadors que porten carregats a priori una sèrie d'entitats de certificació, les més importants a internet. Quan el navegador examina algun certificat i detecta que ha sigut emès per una de les entitats de certificació amb les que confia dóna el certificat per bó immediatament.

La dissertació anterior serveix per introduïr els dos models de confiança següents:

- **Web of trust**. En quest model cada usuari és el responsable de confiar o no amb els certificats dels altres. No hi ha entitats de certificació ni cap mena d'estructura de control dels certificats. Els usuaris se'ls han d'intercanviar entre ells i han de decidir amb quins confien i amb quins no. Es pot implementar la confiança a tercers en el sentit que si en “pere” confia en la usuària “anna” en pere pot decidir si hi confia tant com per confiar també amb tothom amb qui “anna” confia. Aquest és el model usat per PGP (i Open PGP) per xifrar i signar correu. Els certificats que s'utilitzen són 'self-signed', auto-certificats.

- **PKI** o *Public Key Infrastructure* és el model on la confiança es basa en l'entitat emissora del certificat. Existeix una estructura piramidal d'entitats de certificació o CA. Un certificat emès per una CA és vàlid o no segons la confiança que es tingui en la CA. Aquesta CA pot ser de primer nivell (top) o pot ser una delegació que al seu temps va avalada per una CA de rang superior. Llavors la confiança depèn de la "cedena de confiança".

Elements relacinats amb la confiança:

- **CA self-signed** una Certification Authority és una entitat de certificació. Si és independent (s'ha creat ella mateixa) tindrà la confiança que hi tinguin els altres amb ella. Per exemple una empresa pot ser enitat CA per als certificats dels seus pròpis treballadors. Aquests hi confien com a CA però no el món exterior a l'empresa.

- **CA top** les CA que exieteixen a internet poden cer entitats de certificació 'top' o entitats de certificació delegades. Una entitat de primer nivell o 'top' és de fet una entitat auto-signada. A internet existeixen una sèrie d'entitats conegudes amb les que s'hi confia per defecte. Bé, per defecte vol dir que els navegadors web ja porten incorporades aquestes entitats com a CA vàlides (pagant!).

- **CA delegada** una empresa pot emetre certificats actuant com a delegada d'una entitat de certificació de grau superior. Els certificats que emet seràn vàlids per als usuaris que confiïn en ella o podran examinar qui avala la CA i seràn vàlids si confien en l'entitat de grau superior.

- **Cadena de confiança** l'estructura de certificació PKI és una estructura piramidal però no amb un únic node superior, n'hi ha tants com calgui. Una entitat CA delegada actua avalada per una entitat de grau superior que al mateix temps pot actuar avalada per una altra CA, etc. Per validar un certificat es mira qui l'ha emès i es determina si s'hi confia o no. Si l'emisor del certificat està avalat per una altra CA s'examina la confiança amb aquesta altra CA, i així fins a determinar si si confia o no. Aquest mecanisme pel qual "l'alumne" confia en "el profe" que està avalat per "el director" que està valat per "el conseller" s'anomena cadena de confiança.

El funcionament no consisteix en demanar a les CA si un certificat es vàlid o no, sinó a l'inrevés. Qui vol verificar un certificat contrasta si l'emissor és una de les CA de la seva llista de CA vàlides. Els certificats poden incloure certificats d'altres CA generant una cadena de certificacions o *certificate hierarchy*.

> Es confiarà en un certificat si en algun punt de la cadena (cap a l'arrel) es troba una entitat de certificació en la que es confia.

- **Anells de claus** el model de clau privada/pública implica que cal fer arribar els certificats (la clau pública) als altres. Així els usuaris que volen poder xifrar i signar contingut a altres usuaris han d'anar acumulant les claus públiques dels altres. A més a més un usuari pot disposar de diverses parelles pròpies e claus privada/pública destinades a finalitats diferents. De fet també pot tenir certificats diferents generats a partir d'una mateixa clau privada. Cada certificat l'utilitza per a una cosa diferent. Totes aquestes claus es van acumulant en l'anomenat 'anell de claus' o 'clauer'. Igual que al clauer tenim la clau privada de casa i la clau pública de la porta de l'edifici, la privada del cotxe i la pública del garatge, etc.

- **Verificar un certificat** Com es verifica un certificat? Un certificat és una "clau pública més la informació personal del propietàri" i tot això signat per la clau privada de la CA. El que vol fer la verificació ha de disposar de la clau pública de la CA per validar que la firma es vàlida. De fet el certificat usualment també inclou "el pròpi certificat" o clau pública de la CA. Si la verificació és OK sabem que l'ha 'avalat' qui diu que l'ha avalat, però sabem qui és? hi confiem? Si, quan està a la llistat d'entitats de confiança, i si/no quan no, l'usuari ho decideix en la coneguda pantalla del navegador "d'excepció de seguretat". Cal fixar-se que els certificats incliuen doncs les claus públiques de les entitats CA que els emeten. I què són aquestes claus públiques? Són certificats 'avalats' per una altra CA o auto-signats. Es a dir, la clau pública signada per algú altre… i tornar a començar.

Hash

Els algorismes de hash s'utilitzen molt freqüentment en l'informàtica en tot allò que té a veure amb signar, generar 'fingerprints' i fins i tot generar passwords. De fet usualment els passwords són en format MD5 i els 'fingerprint' o 'empremptes dactilars' o 'codis resum' s'utilitzen per exemple en SSH per identificar màquines, també per generar els codis de xequeig per validar el software que ens descarreguem.

Un algorisme de **hash** és un algorsime que donat un contingut d'entrada genera un reqüència resum (per exemple de 128 bits) amb la propietat que el resum generat és únic. És a dir, no hi ha dues entrades que puguin generar el mateix resum. A més a més és impossible determinar quina era l'entrada a partir del *resum*.

Segur que heu vist que els creadors de software (sigui un paquet .rpm, .deb o un .tar) quan el publiquem en una web perquè es pugui descarregar adjunjten també la 'firma digital' o *hash* del paquet. És un *resum* que s'ha det d'ell. Un usuari que es descarrega el paquet pot calcular el resum d'allò que s'ha descarregat realment i comprovar que coincideix amb el *hash* publicat a la web. Si no coincideix és que s'ha produït algun error en la descàrrega o que algú malintencionadament ha modificat el contingut del paquet durant el procés de descàrrega (un *man-in-the-middle*).

> Un resum o **hash** és una cadena de bits que es generen a partir d'un contingut d'entrada donat. Té la propietat de que dues entrades diferents generaran sempre dos resums diferents (no poden generar el mateix hash), i que és imposible obtenir l'entrada a partir del resum.

Els principals algorismes resum són:

- **MD5** o 'Message Digest Algorithm' és un algorisme de hash que genera resums de 128 bits. Usualment es mostra com una cadena de 32 dígits exadecimals.
- **SHA1** o 'Secure Hash Algorism' és l'algorisme de resum fet per l'agència de seguretat dels Estats Units (la famosa NSA de les pel·lícules).

El següent llistat mostra com generar un resum del fitxer dels comptes d'usuari:

```
[root@host m08]# md5sum /etc/passwd
ade21a3a6b41871f3455daa58c794162  /etc/passwd

[root@host m08]# openssl dgst -md5 /etc/passwd
MD5(/etc/passwd)= ade21a3a6b41871f3455daa58c794162

[root@host m08]# sha1sum /etc/passwd
0b2e1d1353c183becfb1294a38a8e30fc75f27c9  /etc/passwd

[root@host m08]# openssl dgst -sha1 /etc/passwd
SHA1(/etc/passwd)= 0b2e1d1353c183becfb1294a38a8e30fc75f27c9
```

En l'exemple següent es mostra com generar un password (el password és “secret”) xifrat amb MD5 i amb SHA1:

```
[root@host m08]# md5pass secret
$1$/ZToHJ75$Ha2c1oklNkk73Li3LBJpo.

[root@host m08]# sha1pass secret
$4$cmnaN0zQ$fK+sXEA8h3ZphKydTcvKQFF5oEo$
```

Signar

El procediment per signar contingut es descriu en el següent llistat:

```
Autenticació + Integritat (firma digital):
M + (Hash(M) + EkprivA) --> ME --> M, Hash(M) = Hash2
(Hash(M) + DkpubA) = Hash
Hash == Has2
M missatge
ME Missatge encriptat
Hash(M): es genera un hash o message digest o fingerprint del missatge.
Hash(M) + EkprivA: Es xifra el hash amb la clau privada de A. Això és una firma digital.
Hash2: es rep el missatge M i es calcula un nou hash en el receptor
```

```
(Hash(M) + DkpubA) La firma digital rebuda (hash xifrat) es desxifra amb la clau publica de l'emisor.
Els dos hash han de ser igual i garanteix l'autenticació i la integritat.
```

Els passos que es realitzen són:

- Es genera un *hash* o resum del missatge.
- Aquest resum es xifra amb la clau privada de l'emissor. Només el resum, no tot el missatge.
- El receptor calcula un resum del missatge que rep (de la part corresponent al missatge).
- El receptor desxifra el hash que ha rebut. El desxifra amb la clau pública de l'emisor i obté el resum original que ha calculat l'emissor.
- Els dos hash han de ser iguals. Si ho són la firma digital és correcta, si no ho són és incorrecta.

Les condicions que es compleixen són:

- Es garanteix l'autenticació. Només l'emissor pot haver generat el missatge si el hash és desxifrable per la clau pública de l'emissor.
- Es garanteix la integritat. Ningú més pot haver modificat el missatge perquè en fer-ho s'haguès modificat el hash que calcula el receptor i no coincidirien. Com que ningú més que l'emissor té la seva clau privada un atacant pot modificar el contingut però no el pot signar de nou (calcular el nou hash)
- No cal xifrar tot el missatge, només el hash, que és la part que garanteix que no s'ha modificat.

Xifrar

El procediment per xifrar contingut és descriu en el següent llistat:

```
Xifrat:
(M + Ek)=ME + (Ek + EkpubB)=EkE --> ME + EkE ==> (EkE + DkprivB) --> Ek
ME + Dk --> M
M Missatge
Ek: clau simetrica de sessió utilitzada per xifrar el contingut. S'utilitza perquè és més
eficient que l'asimètrica.
ME: missatge xifrat amb clau simètrica.
(Ek + EkpubB)=EkE: la clau simètrica Ek s'encripta amb la clau publica del destinatàri.
DkprivB: el receptor desencripta EkE el desencripta usant la clau privada del receptor
per obtenir la clau simètrica.
ME + Dk: el missatge encriptat es desencripta utilitzant la clau simètrica Ek.
```

Els passos que es realitzen són:

- El missatge es xifra amb una clau simètrica. Aquesta clau només la coneix l'emisor. S'utilitza una clau simètrica perquè el xifrat és menys costós computacionalment que usant claus asimètriques.
- La clau simètrica s'encripta usant la clau pública del destinatàri. És el mecanisme per enviar la clau simètrica, 'el secret compartit' de manera segura.
- S'envia el missatge xifrat(simètric) i la clau de xifratge (asimètric).
- El receptor desencripta la clau de xifratge usant la seva pròpia clau privada.
- Un cop sap quin és el 'secret compartit' pot passar a desxifrar el missatge usant aquesta clau simètrica.

Les condicions que es compleixen són:

- S'encripta usant criptografia simètrica perquè és més ràpid i eficient.
- La criptografia asimètrica serveix per transferir-se el 'secret compartit', la clau usada realment per encriptar.
- Aquests tipus de claus s'anomen claus de sessió.

A3 Seguretat Open PGP

En aquest annex es tractaran tots aquells aspectes relacionats amb la gestió de claus Open PGP i l'intercanvi de missatges xifrats i signats utilitzant aquesta tecnologia. Els aparats s'an organitzat dela següent manera:

- A3.1 Gestió de claus Open PGP.
- A3.2 Auxiliar de configuració.
- A3.3 Signar les claus.
- A3.4 Establir la confiança.

En un primer aparat es mostra com gestionar les claus PGP en el *thunderbird*. Com importar-les i exportar-les, examinar el seu contingut, llistar-les etc. En un segon apartat s'examina a fons el funcionament de "l'auxiliar de configuració" que permet generar claus a través de les pantalles d'un auxiliar. El codel de confiança de PGP es basa en el model "*Web of trust*" en que cada usuari és responsable de determinar si accepta o no les claus dels altres, en el tercer apartat es veurà com signar claus d'altres usuaris indicant que tenen el nostre 'aval'. Finalment en l'últim apartat es veurà com establir confiances i graus de confiançes, si jo confio en 'tal' també confio amb qui ell confia? o no?.

A3.1 Gestió de claus Open PGP.

Per poder utilitzar correu PGP cal disposar de claus, de les parelles de claus privada/pública. L'usuari en pot tenir ja de pròpies (per exemple fetes en consola) o en pot generar de noves usant el propi PGP del *thunderbird*. En aquest annex es mostren les opcions de gestió de claus PGP en el *thunderbird*. El llistat d'imatges és:

1. Paràmetres Open PGP.
2. Llistat de les claus de l'usuari "veritat".
3. Dades de la clau pública PGP de l'usuari veritat.
4. Canvia l'identificador principal de la clau.
5. Canvia la contrassenya de la clau.
6. Genera un certificat de revocació de la clau.
7. Crea i desa un certificat de revocació de la clau.
8. Signar una clau (una altra).
9. Estableix la confiança amb una altra clau.

01) En la gestió dels 'paràmetres del compte' del'usuari "veritat" es pot observar el panell anomenat "seguretat OpenPGP" disponible si s'han instal·lat *enigmail*. Aquest panell mostra les opcions de configuració d'OpenPGP. Les principals són:

- Indicar si la seguretat 'OpenPGP' ha d'estar disponible o no per al compte de correu. Un usuari pot activar o desactivar la seguretat amb PGP individualment per a cada un dels seus comptes.
- Establir com identificar 'la clau' si a través de l'adreça de correu a a trvés del ID (identificador) de la clau.
- Opcions de redacció dels missatges:
  - indicar si cal xifrar sempre o no els missatges.
  - indicar si cal signar els missatges xifrats i/p si cal signar els missatge sno xifrats. Si es marquen les dues opcions es signen tots els missatges.
  - si la seguretat "OpenPGP" és la que s'utilitza per defecte o no amb els missatges. Poden existir altres tipus de seguretat, de fet *thunderbird* utilitza certificats digitals S/MIME de manera que es pot escollir quin ha de ser el sistema de seguretat a usar per defecte en la redacció de missatges.

Figura 01. Paràmetres Open PGP.