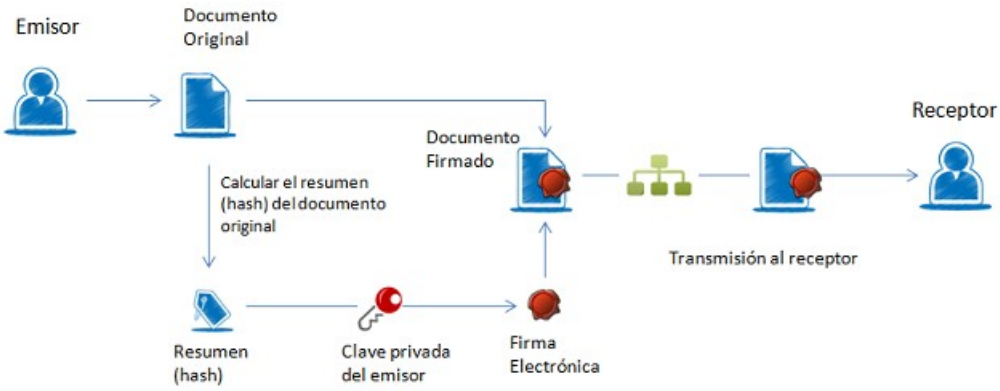
Institut Poblenou
Consorci d'Educació de Barcelona
Generalitat de Catalunya
Ajuntament de Barcelona

ECAIB
Estudis de Comerç, Administració i Informàtica a Barcelona

Pràctica 1 – Firma digital (casolana)

Part 1 - Signar



En la figura de la dreta tenim el procediment per a signar digitalment.

El primer que cal és un fitxer que signar. Hem vist que els algoritmes de HASH fan un resum dels fitxers de manera que es poden identificar unívocament (amb una certesa casi absoluta)

Crearem una classe *Utilitats* on s'hauran de posar tots els mètodes que es faran servir durant el procès. El primer que farem és una utilitat que permeti obtenir el HASH d'un fitxer. D'entrada tothom ha de fer servir l'algoritme MD5. Ha de tenir com a capçalera la següent (més les excepcions si s'escau).

```
public static byte[] digestiona(File f, String algoritme)
```

Per a encriptar el *hash* emprarem l'algoritme RSA (però guardeu el nom de l'algoritme en una variable global per si es volgués canviar). El següent que farem serà generar les claus públiques i les claus privades. Envieu la clau pública que genereu al professor per a que la posi a disposició de la resta dels alumnes (i aquesta serà la vostra oficial). **SI EN ALGUN MOMENT EL PROFESSOR DETECTA QUE UNA CLAU PRIVADA HA ESTAT COMPARTIDA, EL SEU PROPIETARI HAURÀ SUSPÈS LA PRÀCTICA.** El professor no compta.

Només podeu lliurar una clau pública, per tant un cop fet ja no hi haurà marxa enrere. El fitxer com a nom el vostre cognom i com a extensió *.pub*.

```
public static KeyPair generateKey()
```

Un cop tenim la clau i el *hash* del fitxer, haurem de encriptar-lo. Feu una funció que l'encripti aquesta part. Un cop ho haguéssiu fet, afegir els 128 bytes resultants al final del fitxer original. Aquest serà el fitxer signat. Atès que haureu de manegar-vos amb *arrays* de bytes, us recomano que feu servir la funció *System.arraycopy*. A continuació hi ha una serie de capçaleres recomanades.

```
public static byte[] signar(byte[] text, PrivateKey key)
```

La funció *read* passar un fitxer a un array de bytes. La funció *write* guarda un array de bytes en un fitxer.

```
public static byte[] read(File file) throws IOException
```

```
public static void write(String f, byte[] byteArray)
```

Part 2 – Verificar

Feu un programa que faci justament el procés invers, és a dir, extregui els últims bits on està la informació, la descodifiqui amb la clau pública i que finalment compari el *hash* desencriptat amb el *hash* del fitxer que heu rebut, on haureu retallat els últims bytes.

```
public static byte[] verificar(byte[] text, PublicKey key) throws
InvalidKeyException
```

Part 3 – Posada de llarg

Un cop feta la part 1, el professor us facilitarà un document. Ús dirà el seu propietari. Verifiqueu que realment el document és del propietari i que el professor no l'ha de modificat. Si us dona error heu de ser capaços de distingir quin error ha estat, és a dir, que el professor us ha dit el propietari erroni, o bé, el professor ha modificat el fitxer, o les dues coses.

NOTA – La nota d'aquesta pràctica, així com les correccions i comentaris us seran entregats encriptats amb la vostra clau pública, així que si la voleu saber necessitareu un programa que desencripti amb la vostra clau privada. Com a pista, heu de saber que el procés d'encriptar és just al revés que el de signar. L'usuari encripta amb la vostra clau pública i vosaltres sou els únics que podreu desencriptar-lo a partir de la vostra clau privada. Les funcions són iguals que les de signar i verificar, però canviant *PublicKey* per *PrivateKey* i viceversa.