Activitat 1.6. Certificats i Magatzems de claus.

EXERCICI 1: Integrant els quatre serveis de seguretat

OBJECTIU: Combinar tots els coneixements adquirits en xifrat i firma RSA per generar dues aplicacions en Java que garanteixin els quatres serveis bàsics de seguretat, alhora, a un missatge: privadesa, integritat, autenticació i no repudi.

Volem que dues persones, anomenades origen i destí, es puguin intercanviar missatges de manera segura. Per fer-ho, seguiu les següents passes:

1. Genereu mitjançant la **Keytool** un magatzem de claus JKS per cadascú, anomenats desti.jks i origen.jks.

   A cada magatzem hi ha dues entrades:
   a. Una pel parell de claus del propietari, RSA de 2.048 bits
   b. Una altra amb el certificat de confiança de l'altre persona.

   Podeu usar certificats autosignats, tal com els genera automàticament Keytool.
   El propietari d'origen.jks és qui generarà els missatges, i el de desti.jks qui els rebrà.

   Tot i que executareu l'aplicació al mateix ordinador, heu de comptar com si cada magatzem fos a un equip diferent a Internet.

   Per generar els fitxers necessaris en desti fem:

- **`keytool -genkey -alias desti -keyalg RSA -keystore desti.jks -keysize 2048`**
- **`keytool -exportcert -file desti.crt -keystore desti.jks -alias "desti"`**
- **`keytool -importcert -file desti.crt -keystore origen.jks -alias "desticert"`**

Haureu de fer el mateix per a origen però canviant-ho per desti.

2. Crear una classe origen.java que, donada una frase de text de mida arbitrària (sense límit de llargària) escrita pel teclat, la signi i la xifri, a partir de les dades que hi ha a origen.jks.

   Tant per xifrar com per signar cal usar l'algorisme RSA. El resultat del procés es desa a un fitxer anomenat secureMsg.rsa. Ningú que obri aquest fitxer ha de ser capaç de saber quina era la frase original.

Consell: Com que ha de permetre dades de mida arbitrària, cal embolcallar una clau simètrica a través d'RSA. En aquest cas s'ha usat una clau AES de 128 bits.

PASOS A SEGUIR:

- Feu una funció que carregueu el magatzem d'origen per tal d'extraure les dades per signar i després xifrar. A les transparències podeu veure la funció que apareix a l'apartat de gestió de claus en magatzems de claus.
- Fes una funció o varies que facin les següents tasques:
    - Obtenir les dades necessàries a partir del magatzem carregat anteriorment. Quines dades son les necessàries?
        - magatzem: Tipus KeyStore. Passeu el alias del magatzem
        - clau privada: Tipus PrivateKey i és d'origen
        - certificat: Tipus X509Certificate i es del destí. Heu de crear una variable com la següent:

```
X509Certificate c =
(X509Certificate)magatzem.getInstance(CRT_DESTI)
;
```

        - clau publica: Tipus PublicKey i serà del destí
    - **Generem la firma i signem.** Aqueta funció apareix a les transparències a l'apartat de generació de firma amb RSA. El que fem es generar un objecte tipus Signature i signar amb aquest les dades que hem replegat per teclat.
    - Xifrem am el mètode embocallat de RSA.
        - Hem de xifrar el missatge simètricament
        - Hem de xifrar la clau simètrica usant un mecanisme asimètric.

    Aquest procediment apareix a les transparències.

    - Desar tot al fitxer secureMsg.rsa de la següent manera:

```
File f = new f("secureMsg.rsa");
FileOutputStream out= new FileOutputStream (f);
out.write(encKey);
out.write(encMsg);
```

3. Crear un programa que llegeixi el fitxer secureMsg.rsa i sigui capaç de desxifrar i validar les dades usant el fitxer desti.jks. Si les dades del missatge original són vàlides, es mostren per pantalla. Aquestes funcions han d'estar en una classe desti.java

Podeu triar vosaltres mateixos qualsevol detall sobre l'aplicació no indicada a l'enunciat

ENTREGA:

- Entrega un fitxer comprimit en el qual has d'afegir els fitxers .java generats i necessaris per poder executar l'aplicació.
- Penja-ho la moodle amb el teu nom i nom de la activitat