

## Sistema criptogràfic

Aquesta pràctica consisteix a combinar les funcions construïdes en les pràctiques anteriors per tal d'implementar un sistema de comunicacions segur, amb garanties de privacitat, integritat, autenticitat i no-repudi de la informació intercanviada entre els usuaris del sistema.

Es suposa que cada usuari té un parell de claus per al sistema DH i un parell de claus, associat a uns paràmetres, per al sistema ECDSA.

**Enviar un missatge.** Quan un usuari (l'emissor) vol enviar un missatge  $M$  a un altre (el receptor) procedeix de la manera següent:

1. Firma el missatge en clar  $M$  amb la seva clau privada ECDSA afegint-li els 128 bytes corresponents a la firma; diguem  $M\|F$  al missatge amb la seva signatura.
2. Genera aleatòriament una llista de 64 bytes,  $KSE$ , i fent servir una clau privada ECC i una clau pública ECC crea una clau  $KS$  de 256 bits per l'AES, que s'anomena clau de sessió:  $KS$  són els primers 256 bits obtinguts al fer servir ECCDHKT amb  $KSE$ , la clau privada ECC i la clau pública ECC.
3. Amb  $KS$  xifra el missatge  $M\|F$  segons el mode d'operació CBC. La longitud dels blocs de l'AES la fixarem a 128 bits. Notarem per  $E(M\|F)$  el missatge xifrat.
4. Concatena  $KSE$  amb  $E(M\|F)$  i envia el resultat  $KSE\|E(M\|F)$  al receptor.

**Rebre un missatge.** Quan el receptor rep el criptograma  $KSE\|E(M\|F)$  procedeix en sentit invers per tal de recuperar el missatge en clar i verificar la signatura:

1. Primer descompon la informació rebuda en dos troços corresponents a  $KSE$  i  $E(M\|F)$ .
2. Recupera la clau de sessió  $KS$  fent servir  $KSE$  i les claus ECC corresponents.
3. Desxifra el missatge  $E(M\|F)$  amb la clau de sessió  $KS$  i el sistema de clau secreta AES, obtenint  $M\|F$ .
4. Recupera el missatge  $M$  i verifica la firma  $F$  amb la clau pública ECDSA de l'emissor. Si la verificació és correcta retorna  $M\|F$  concatenat al byte 0x00 ( $M\|F\|0x00$ ), si la verificació és incorrecta retorna  $M\|F$  concatenat al byte 0xff ( $M\|F\|0xff$ ).

**Implementació: signatures.** Definiu la classe `sistemaCriptografic` amb els següents mètodes:

```
public static byte[] enviarMissatge(byte[] M, BigInteger clauDeFirma, BigInteger clauPrivadaECC,
                                   BigInteger[] clauPublicaECC, BigInteger[] parametresECC)
```

entrada:  $M$  és una llista de bytes que és el missatge a enviar,  
           $clauFirma$  és la clau privada del firmant,  
           $clauPrivadaECC$  és un enter,  
           $clauPublicaECC=\{P_x, P_y\}$  (diferent del punt de l'infinit)  
           $parametresECC=\{n, G_x, G_y, a, b, p\}$ ,  $G = (G_x, G_y)$  punt d'ordre  $n$  de la corba  $y^2 = x^3 + ax + b \bmod p$  (evidentment,  $G$  no és el punt de l'infinit);  
sortida: una llista de bytes que representa  $KSE\|E(M\|F)$ .

```
public static byte [ ] rebreMissatge(byte[] C, BigInteger[] clauDeVerificacioDeFirma,
                                     BigInteger clauPrivadaECC, BigInteger[] clauPublicaECC,
                                     BigInteger[] parametresECC)
```

entrada: **C** és una llista de bytes imprimible que és el criptograma rebut,  
**clauDeVerificacioDeFirma** clau pública de verificació de firma del signant,  
**clauPrivadaECC** clau privada corresponent a la clau pública feta servir per xifrar el missatge,  
**clauPublicaECC** clau pública corresponent a la clau privada feta servir per xifrar el missatge,  
**parametresECC**={ $n, G_x, G_y, a, b, p$ },  $G = (G_x, G_y)$  punt d'ordre  $n$  de la corba  $y^2 = x^3 + ax + b \bmod p$  (evidentment,  $G$  no és el punt de l'infinit);

sortida: una llista de bytes  $M\|F\|ver$  on **M** és el missatge en clar un cop dexifrat, **F** és la signatura i **ver** és un byte que val 0x00 si la verificació de la firma és correcta i 0xff si no s'ha verificat la firma.