

Annexe A : Lexique de Cryptologie

Code: Système de symboles (mots, nombres, signes, etc.) remplaçant des mots entiers. Exemples: "007 à la place de "James Bond"

Encoder: Modifier la structure d'un ensemble de données en lui appliquant un algorithme (chiffrement, méthode de compression ...). L'encodage n'a pas forcément un but cryptographique.

Force brute: L'attaque par la force brute est la seule à laquelle aucun algorithme ne résiste: elle consiste à tester toutes les clefs possibles, jusqu'à trouver la bonne. Elle ne constitue pas souvent une bonne approche car elle nécessite souvent des jours, des mois, voire des années pour trouver la clef. On peut l'optimiser en se servant d'un dictionnaire.

Hachage : Fonction appliquée à un document de longueur variable qui renvoie un nombre de longueur fixe caractéristique du document : c'est l'empreinte du document. Une légère modification du document entraînant une modification visible de l'empreinte, celle-ci permettra de vérifier l'intégrité du document.

NSA: National Security Agency, l'agence nationale de sécurité américaine, qui s'occupe autant de cryptographie que de cryptanalyse. On dit souvent qu'elle est le plus grand employeur de mathématiciens du monde. Voir le site officiel de la NSA: <http://www.nsa.gov/>.

Padding: Ajout de valeurs aléatoires pour obtenir une longueur de message constante.

Scytale : Une scytale consiste en un bâton de bois autour duquel est entourée une bande de cuir ou de parchemin.

Watermarking: Application particulière de la stéganographie consistant à camoufler dans une image des informations sur son origine (nom de l'auteur, copyright ...).

XOR : Ou exclusif

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

Annexe B : Génération des nombres en RSA

P avec la taille 2048 bit

208830037722030455246866928275923134154151594576192265020792124003476370774
 514579023326999091671063370398937008606566283145637918135596312070026221985
 803417150665943197507044389629344713797655411305523838221819042984562243684
 581017415092252873835311856542335677499660164631086470629487165632576434499
 492975194724834283722743065863743266608651817362771609528352097009602885731
 390605883263636957970726289110050906478053254434717659566525046214922442519
 371114966792153782835827071351513457513972441596793452220896693905174362424
 282359074748035852585335897274104660657704125882576757929778452626630934818
 88517793722170877

Q avec taille 2048

182716682072029781721587307740383010846060248567184644484438739789789308372
 245853236398481571241227662956285026790729019680359196842549525125899765845
 560672505719459164888767398498051843399258619755502429150263836008555799984
 036455575028951571005495914952525140399224177804655899725783603502563646782
 316411299323577006826604188190705051873976334621516034630964367724713165050
 692869352865372443471370861296628635820192070064070098168607921382093755839
 825427084981155385013215027607438271542096430010723935975357400960906437919
 053584650576111854396106572923489361308327707662500129661975247936854241119
 03394815949499739

Module de chiffrement $N = P \cdot Q$

636194455987463591908342597798128657137254141045134964970631737938102215535
 673233222190643845333701410045376671299394851440645057106581237869860369802
 940845496024529225108859813899202383765909006076560544057974591177273922166
 565549628371524598394589389615257054356284371136408068071932672855376295589
 450620928725231508682813277036496724525567196447102982744968669072639616015
 511598030131928978776996532641918546260263334697259253783406763722758918354
 648473027058407900960807326202483925139143232492906964257572020049345265335
 853007616553058722094299470485417673867112770300720046294774049749812674255
 689435210177584986999688775040278037252170730908096808208764489097036699364
 975833959438064730916768787793958434591525152004550053020595548790963929761
 291841082966552968879272203268138429827677080879286213920429694507830451536
 635871777211603233743676599503033646821306619441053878236360256476772343524
 225921593103808111931787822763118846754552811783134565632656056843661630681
 672650460593797231922226800525390683668802210078147362422925554536378430045
 528122369007235980982983577776109555261864360165484584089513453929983285380
 147571033444004194933250499300850483179164456712694331857376340554373474316
 278225921038642066298628790756809.

Annexe D : Code source en java

Ü Génération des nombres premiers aléatoires :

```
BigInteger nbr = new BigInteger (1, new Random ()).probablePrime (size/2, new Random ());
```

Ü Multiplication entre deux grands nombres

```
n = p.multiply (q);
```

Ü Calcul du modulo inverse à e pour avoir la clé privée d

```
d = e.modInverse (w);
```

Ü Génération une clé d'AES

```
KeyGenerator keyGen = KeyGenerator.getInstance ("AES");
```

```
keyGen.init (KEY_SIZE);
```

```
SecretKey = keyGen.generateKey ();
```

Ü Chiffrement par AES un texte M

```
Cipher cipher = Cipher.getInstance ("AES");
```

```
cipher.init (Cipher.ENCRYPT_MODE, secretKey);
```

```
Return cipher.doFinal (M.getbyte);
```

Ü Lecture manuelle des composantes de pixel (alpha, rouge, vert, bleu)

```
Int rvb = photooriginale1.getRGB (i, j);
```

```
Color color=new Color (rvb);
```

```
Int a= color.getAlpha ();
```

```
Int r= color.getRed ();
```

```
Int v= color.getGreen ();
```

```
Int b= color.getBlue ();
```

Ü Construction d'un pixel

```
Int rvb = (a << 24) + (r << 16) + (v << 8) + b;
```

Ü Le temps d'un système en quel instant par nanoseconde

```
Long d1 = System.nanoTime ();
```