Programmation de RSA

Votre travail doit être rendu avant le mercredi 6 décembre à midi. Le travail est effectué avec les binômes imposés. Tous les fichiers utiles doivent être envoyés par mèl (avec une description de ces fichiers) à `coulbois@cmi.univ-mrs.fr`. Le code doit être lisible et bien documenté.

L'objectif de ce travail est de programmer RSA en Java.

Les API Java contiennent des classes qui faciliteront votre travail. La classe `BigInteger` contient en particulier, toutes les opérations mathématiques dont vous avez besoin. Plus généralement, la `Java cryptography extension` vous aidera à faire l'interface avec d'autres systèmes cryptographiques. Vous êtes invités à minimiser votre effort de programmation en utilisant les fonctionnalités de Java.

L'objectif minimal de ces 4 séances (deux TP et deux TD) est de proposer :

1. une fonction de génération et de vérification de clés RSA ;
2. les fonctions de chiffrement et déchiffrement RSA ;
3. une interface pour chiffrer et déchiffrer un texte court (on déterminera la taille maximale du texte) qui respecte le bourrage normalisé par le PKCS #1. On supposera que la valeur de `L` est nulle et que la fonction de hachage est SHA-1, ce qui permet d'utiliser la valeur donner dans la note page 20 de PKCS #1. Pour la fonction de génération de masque MGF, on répètera la chaine passée en paramètre autant de fois que nécessaire.
4. une interface pour chiffrer et déchiffrer un texte de longueur arbitraire en mode CBC.

Optionnellement vous pourrez écrire une fonction d'importation de clés RSA de SSH ou GPG. Vous pourrez au choix utiliser la librairie openssl (et notemment openssl rsa) ou bien les packages Java java.security, et notemment les classes KeyFactory, X509EncodedKeySpec ou PKCS8EncodedKeySpec. Pour tester vos programmes, vous déterminerez les exposants publics utilisés par SSH et vous vérifierez les clés disponibles sur votre machine. Pour ce travail optionnel vous serez amenés à lire les RFC correspondant à SSH et à ses fichiers de stockage de clés.