Développement Logiciel Cryptographique

Examen de session 1

Février 2020

Préambule

Les supports de cours de l'UE Développement Logiciel Cryptographique **sont les seuls documents autorisés** pour cet examen.

L'usage d'une calculatrice non programmable est autorisé.

Durée 1h30.

Important : Même lorsque la question ne le demande pas explicitement, vous devez expliquer et justifier toutes vos réponses.

1 Quizz [10 points]

1. On utilise le RSA avec une clé dont le module a une taille de 1024 bits et dont l'exposant public est $e = 2^{32} + 1$.

   a) Lorsque l'on utilise la méthode du Square & Multiply de gauche à droite, le temps de chiffrement d'un message m est-il significativement ou marginalement plus rapide lorsque m a une taille de 100 bits que lorsque m est "full size" ?

   b) Même question lorsque l'on utilise la méthode du Square & Multiply de droite à gauche.

   c) Lorsque l'on utilise la méthode du Square & Multiply de gauche à droite, le temps de déchiffrement d'un chiffré c est-il significativement ou marginalement plus rapide lorsque c a une taille de 100 bits que lorsque c est "full size" ?

   d) Même question lorsque l'on utilise la méthode du Square & Multiply de droite à gauche.

2. Quelle peut être la taille du produit d'un entier de n_1 bits par un entier de n_2 bits ? Justifiez votre réponse.

3. Dans l'attaque SQUARE à 4 tours avec un seul λ-set :

   a) Comment utilise-t-on les chiffrés pour réduire le nombre de candidats pour chaque octet de K_4 ?

   b) En moyenne combien de candidat reste-t-il par octet de clé ? Expliquez.

   c) Expliquez en détail comment terminer l'attaque.

4. Est-il plus rapide de chiffrer avec RSA en mode standard plutôt que de déchiffrer en mode standard ?

5. Est-il plus rapide de chiffrer avec RSA en mode standard plutôt que de déchiffrer en mode CRT ?

6. Mêmes questions que les deux précédentes mais en considérant maintenant un chiffrement en mode CRT.

7. Écrivez le pseudo-code d'une multiplication scalaire sur courbes elliptiques par la méthode du *double-and-add de droite à gauche*.

8. L'exposant public d'une clé RSA doit-il nécessairement être premier ? Peut-il être choisi pair ? Justifiez vos réponses.

9. Supposons que le module d'une clé RSA de 1024 bits soit généré en multipliant un entier p choisi aléatoirement parmi les premiers de 512 bits par l'entier q égal à $nextprime(p)$.

   a) Est-il possible d'exhauster l'ensemble des couples de premiers pouvant entrer dans la composition d'un tel module ? Justifiez.

   b) Est-il possible de retrouver p facilement ? Expliquez.

2 Bibliothèque GMP [5 points]

1. La fonction `mpz_probab_prime_p(const mpz_t n, int reps)` permet de tester la primalité de l'entier `n`.

   a) Expliquez pourquoi il est déconseillé de générer un nombre premier avec l'implémentation suivante :

```
do
    mpz_urandomb(z_n, prng, bit_size);
while (mpz_probab_prime_p(z_n, 5) != 2);
```

   b) Dans quelle situation adopter cette implémentation ne présente aucun problème ?

2. Écrivez en langage C une fonction qui prend en entrée deux grands entiers a et b de type `mpz_t` (on supposera que $a < b$), et qui génère pour l'appelant un grand entier n pair aléatoire et uniformément distribué dans l'intervalle $[a, b]$.

3 Génération de premiers [5 points]

1. Un nombre de Carmichael peut-il être déclaré premier par un test de Miller-Rabin ? Expliquez.

2. On suppose une génération de premiers consistant à répétitivement tirer au hasard un entier n et à en tester la primalité par un test de Miller-Rabin à 5 itérations. On considère que le temps mis pour générer un premier de cette manière est essentiellement dominé par les exponentiations modulaires. On note que la densité des premiers autour de x est bien approximée par $1/\ln x$.

   a) Évaluez le nombre moyen de candidats testés pour générer un premier de 512 bits.

   b) Évaluez le nombre total moyen d'itérations effectuées pour générer un premier de 512 bits.

c) En supposant que la calcul d'une exponentiation modulaire sur des nombres de 512 bits prend 10 μs, évaluez le temps moyen de génération d'un premier de 512 bits.

d) Que deviennent les réponses aux questions ci-dessus dans le cas d'une génération d'un premier de 1024 bits ?