Développement Logiciel Cryptographique

Examen de session 1

Février 2015

Préambule

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

L'usage d'une calculatrice est autorisé.

1 Quizz RSA [6 points]

1. Pour quelle raison est-il dangereux d'utiliser un module RSA égal au produit de deux premiers p et q de 300 bits chacun ?

2. Pour quelle raison est-il dangereux d'utiliser un module RSA égal au produit d'un premier p de 200 bits par un premier q de 2000 bits ?

3. 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 ?

   b) Comment feriez-vous pour retrouver p ?

4. Pourquoi n'utilise-t-on pas le mode CRT du RSA pour chiffrer un clair ou vérifier une signature plus rapidement ?

2 Quizz GMP [4 points]

1. La librairie GMP fournit la fonction `mpz_pow_ui` qui permet d'élever une base de type `mpz_t` à une puissance de type `unsigned long int` :

   ```
   void mpz_pow_ui (mpz_t rop, mpz_t base, unsigned long int exp)
   ```

   Pourquoi la fonction duale suivante n'existe pas dans GMP ?

   ```
   void mpz_ui_pow (mpz_t rop, unsigned long int base, mpz_t exp)
   ```

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 Attaque SQUARE [6 points]

1. Dans la version à un seul λ-set de l'attaque SQUARE sur l'AES à 4 tours, l'adversaire obtient 16 listes de candidats, une pour chaque octet de la clé K_4 du dernier tour.

   Écrivez le pseudo-code d'une fonction permettant d'exhauster l'ensemble des candidats K_4 formés de 16 octets provenant respectivement de chacune de ces listes.

   Votre programme devra éviter l'utilisation de 16 niveaux de boucles imbriquées.

2. Décrivez en détail comment calculer un candidat pour la clé de chiffrement K à partir d'un candidat pour la clé de tour K_4.

3. Comment adapteriez-vous l'attaque SQUARE sur l'AES à 4 tours à une version de l'AES à 5 tours ?

   Expliquez en détail votre méthode pour retrouver la clé, en n'hésitant pas à faire usage de schémas pour illustrer votre propos.

4 Génération de premier [4 points]

1. Combien doit-on effectuer d'itérations du test de FERMAT pour obtenir un premier avec un degré de confiance dans sa primalité de l'ordre de 10^{-9} ?

2. Même question pour le test de MILLER RABIN.

3. Voici le pseudo-code d'une méthode de génération aléatoire d'un nombre premier de k bits :

```
fonction generate_prime( E k, t : entier ) : entier
{
  variables :
    b_prime : booléen
    i : entier

  début
  tant que vrai faire
  {
    candidat <- entier aléatoire de k bits
    b_prime <- vrai
    i <- 1
    tant que (i <= t) et (b_prime = vrai)
    {
      base <- entier aléatoire entre 1 et candidat-1
      si (miller_rabin(base, candidat) = "composé")
        b_prime <- faux
      i <- i + 1
    }
    si (b_prime = vrai) retourner candidat
  }
  fin
}
```

Proposez plusieurs idées permettant d'optimiser cette fonction en terme de temps d'exécution. Dans chaque cas, dites si votre proposition a un impact sur la probabilité que l'entier généré soit effectivement premier.