Nombres premiers aléatoires

Thales Communications & Security

Les clés utilisées en cryptographie à clé publique sont en général construites à partir de grands nombres premiers choisis aléatoirement.

Pour générer des nombres premiers, la méthode habituellement utilisée est de générer des entiers sans petits facteurs, en utilisant un algorithme qui parcourt les entiers de manière intelligente et une méthode pour filtrer ceux qui sont premiers.

Pour le parcours des entiers, on se réfère par exemple à une méthode développée par M. Joye et P. Paillier [1], [2].

Il existe trois grandes classes algorithmes pour la déterminer primalité. L'une est représentée par l'algorithme AKS qui détermine en temps polynomial si un nombre est premier, mais dont le coût en pratique est prohibitif. La seconde est représentée par des algorithmes dont le coût est en pratique relativement raisonnable, bien que mal maîtrisé. La dernière est représentée par les tests probabilistes dont le coût est très intéressant, mais qui ont l'inconvénient de pouvoir déclarer premier — mais avec une probabilité extrêmement faible — un nombre qui ne l'est pas.

Pour que la partie privée de la clé ne soit pas divulguée, il est souhaitable que les nombres premiers soient générés par le dispositif qui va l'utiliser. Ce dispositif est souvent une carte à puce ou un processeur de faible puissance, et la méthode des tests probabilistes s'impose dans ce cas.

Il y a plusieurs familles de tests probabilistes. La plus répandue est basée sur une version forte du petit théorème de Fermat. Le test est paramétré par un entier b appelé la *base*. Un nombre passant ce test est — avec très grande probabilité — un nombre premier, ou bien très exceptionnellement un nombre non premier que l'on appelle alors un *pseudopremier de base* b, abrégé en psp(b). Dans la pratique, on utilise ce test avec une dizaine de bases et la probabilité d'obtenir un nombre non premier est quasiment nulle. Mais l'envie de se débarrasser du qualificatif « quasiment » persiste.

Une autre famille de tests probabilistes est basée sur les suites de Lucas. Ce *test de Lucas* ne coûte pas beaucoup plus cher que le test précédent et sa probabilité de se tromper est aussi faible. Lorsque l'on combine les deux tests (le test résultant s'appelle le test de primalité de Baillie-PSW [3]) on se trouve dans une situation paradoxale : il n'y a pas de raison ferme de penser qu'aucun nombre composé passant les deux tests n'existe et pourtant, les tentatives faites pour en trouver ou en construire un ont toutes échoué.

Le sujet consiste :

• à trouver d'autres idées pour obtenir un test de primalité encore plus rapide et/ou encore plus fiable

• à proposer des méthodes astucieuses pour construire des entiers composés échappant aux deux tests principaux.

• à mesurer la vitesse de génération de nombres premiers de taille 2048 et 4096 bits au moyen de cette méthode.

Il y a très peu de prérequis pour aborder ce sujet qui repose sur des manipulations d'arithmétique modulaire.

[1] Efficient Generation of Prime Numbers. Marc Joye, Pascal Paillier, and Serge Vaudenay. CHES, volume 1965 of Lecture Notes in Computer Science, page 340-354. Springer, (2000).

[2] Marc Joye and Pascal Paillier. Fast generation of prime numbers on portable devices: An update. In L. Goubin and M. Matsui, Eds, Cryptographic Hardware and Embedded Systems - CHES 2006, vol. 4249 of Lecture Notes in Computer Science, pp. 160-173, Springer, 2006.

[3] http://en.wikipedia.org/wiki/Baillie-PSW_primality_test