Aujourd'hui la tradition de l'édition des tables de nombres aléatoires s'est adaptée, et un CD-ROM de plus de quatre milliards de bits aléatoires a été récemment édité par Marsaglia selon une méthode que je vais décrire. Des sites Internet proposent aussi des monceaux de bits aléatoires à qui veut bien.

Partant du constat que *(a)* les méthodes purement physiques produisent souvent des séquences avec des déviations de fréquences et autres défauts de mélange et qu'il faut donc s'en méfier en simulation, mais que *(b)* les méthodes par générateurs algorithmiques ne sont pas sûres, principalement à cause de leur prévisibilité, Marsaglia a additionné les deux procédés.

Il a pris des séries de bits engendrées par des procédés physiques et des séries générées par des générateurs algorithmiques ayant passé les tests de bon mélange, et a opéré un *ou-exclusif* entre les deux séries : '0' et '0' donnent '0' ; '0' et '1' donnent '1' ; '1' et '0' donnent '0' ; '1' et '1' donnent '1'. Cela combine les bonnes propriétés des deux méthodes : l'imprévisibilité est garantie par la composante physique, le bon mélange par la composante algorithmique. Bien sûr, ces séries ont été soumises aux batteries de tests dont on dispose. Notons que ces même batteries indiquaient clairement que la plupart des générateurs physiques ne fonctionnaient pas de manière très satisfaisante.

Même s'il a été produit avec soin et qu'il sera certainement utile, le CD-ROM n'est malheureusement pas la solution ultime. D'une part, pour la simulation, il se peut qu'on ait besoin de séries aléatoires encore plus longues que celles du CD-ROM. D'autre part, en cryptographie, le fait que le CD-ROM soit disponible à tous est quelque peu inquiétant !

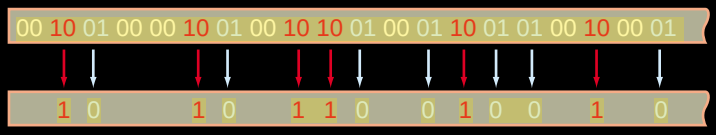
Les générateurs physiques fondés sur le bruit thermique tel QNG de la Société *ComScire* (http://shell.rmi.net/~comscire), ou sur la diode Zener exploitant l'effet tunnel des électrons, comme ORION (http://195.11.242.249/ave/com/orion/home.html) ont encore de beaux jours devant eux. Selon les applications, on conseillera donc de les utiliser en les combinant (en utilisant un *ou-exclusif*, par exemple) avec des méthodes simples (par exemple, des générateurs congruentiels linéaires) ou des méthodes dont l'imprévisibilité est admise.

L'ERREUR DE NETSCAPE

Dans les applications où peu de bits aléatoires sont nécessaires, l'idée ancienne de mesurer un paramètre dépendant du contexte physique est bonne, surtout si on la combine avec une autre.

6. UNIFORMISATION D'UN GÉNÉRATEUR BIAISÉ

Lorsqu'on utilise un générateur physique, il est souvent difficile d'avoir autant de «0» que de «1» (c'est-à-dire un générateur équilibré). Il y a plus de 40 ans, von Neumann a proposé un moyen très simple de corriger un tel défaut : prendre les bits deux par deux ; supprimer tous les couples «00» et tous les couples «11» ; transformer tous les couples «01» en «0» et tous les couples «10» en «1». Exemple :



On perd quelques bits, mais on corrige un éventuel déséquilibre entre le nombre moyen de «0» et le nombre moyen de «1».
Si la suite initiale donne «1» avec une probabilité p et «0» avec une probabilité $1 - p$, le couple «01» a pour probabilité $p(1–p)$ comme le couple «10», et donc la suite obtenue par le procédé de von Neumann donne bien «0» et «1» avec la même probabilité, c'est-à-dire 1/2.

C'est ce qui est fait dans les machines à sous des casinos où un générateur simple tourne en permanence, dont on ne prend en compte le résultat qu'à l'instant précis où le joueur joue (ce qui assure une forme d'imprévisibilité). Des mécanismes de mesure de la date ou des caractéristiques de la frappe de celui qui est devant l'ordinateur peuvent être utilisés pour obtenir un petit nombre de bits aléatoires, et c'est ce que fait PGP (*Pretty Good Privacy*, le fameux logiciel de cryptographie utilisé sur *Internet*) pour engendrer ses clefs aléatoires.

Mais là encore attention : il y a deux ans, le protocole de communication confidentielle de *Netscape*, célèbre logiciel de navigation sur *Internet*, a été cassé par des étudiants. Ils avaient remarqué que les clefs étaient engendrées par lecture de certaines parties de la mémoire de l'ordinateur selon un procédé qui ne pouvait conduire qu'à des clefs appartenant à une partie assez réduite de l'ensemble des clefs possibles (ensemble *a priori* trop grand pour être exploré exhaustivement). Les clefs possibles étant en nombre réduit, l'approche exhaustive devenait utilisable et permettait de décrypter les messages secrets. L'erreur a été corrigée.

Cette mésaventure s'est produite fréquemment, et c'est pourquoi une note intitulée *Randomness Recommendation for Security*, rédigée par des experts en sécurité, insiste sur une erreur commune : en piochant dans une très grande base de données (par exemple, le disque dur de l'ordinateur de l'utilisateur), on ne tombe pas toujours sur des bits aléatoires.

Les disques des ordinateurs (surtout s'ils sont neufs) comportent de grandes zones vides ; de plus, les fréquences d'utilisation des caractères ne sont pas uniformes, ce qui conduit à fausser gravement l'uniformité *a priori* sur les clefs engendrées et rend parfois, comme dans le cas évoqué ci-dessus, envisageable une approche exhaustive.

Les recommandations des experts sont là encore de mélanger plusieurs sources non corrélées, en utilisant le *ou-exclusif* déjà mentionné ou d'autres fonctions plus perfectionnées. Ils insistent aussi sur la taille des clefs qui protègent contre les attaques par énumération exhaustive : elles doivent comporter au moins 50 bits (10^9 clefs pourraient bientôt être traitées par seconde, et donc plus de 10^{15} par mois, ce qui correspond à 50 bits).

Ces exemples confirment que, dans le domaine de la génération de suites aléatoires, courtes ou longues, à usage cryptographique ou pas, les pièges sont partout présents et infiniment variés : on ne se méfie jamais assez du hasard.

Jean-Paul DELAHAYE est directeur adjoint du Laboratoire d'informatique fondamentale de Lille du CNRS.

e-mail : delahaye@lifl.fr

L. BLUM, M. BLUM et M. SHUB, *A simple unpredictable pseudo-random number generator,* SIAM J Computing 15.2 pp. 364-383, 1986.

J.-P. DELAHAYE, *Le fascinant nombre Pi*, Éditions *Pour La Science*, Paris, 1997.

L. GOUBIN et J. PATARIN, *La génération d'aléas sur l'ordinateur,* in *Quadrature*, oct.-nov.-déc. 1997, pp. 27-36.

P. L'ÉCUYER, *Random number generation. Hanbook on Simulation*, Ed. Jerry Banks, Wiley, 1997.

B. SCHNEIER, *Cryptographie appliquée*, Thompson Publishing, 1995.

C. P. SCHNORR, *A survey of the theory of random sequences*, in *Basic Problems in Methodology and Linguistics*, Dordrecht, pp. 193-210, 1977.