3. LES DIFFÉRENTS HASARDS UTILISÉS

Hasard faible

Bon mélange, uniformité, longue période, satisfaction des tests statistiques. Peut être produit par des algorithmes rapides. Utile en simulation.

Hasard moyen

Imprévisibilité pour un observateur ne disposant que de moyens de calcul réalistes.
Peut être produit (vraisemblablement) par algorithmes moyennement rapides. Utile en cryptographie.

Hasard fort

Imprévisibilité totale, incompressibilité, contenu maximum en information.
Ne peut jamais être produit par algorithme,
mais vraisemblablement par des moyens physiques.
Utile en cryptographie et en théorie de l'information.

divise M divise aussi $a - 1$, et si M n'est pas divisible par 4, alors la période du générateur est maximale, c'est-à-dire M. Si M est un multiple de 4, il faut que $a - 1$ le soit aussi pour que la période soit maximale

Autrement dit, si vous vous y prenez bien, en partant d'une valeur $x(0)$ quelconque (entre 0 et $M - 1$), vous n'y reviendrez qu'après être passé par tous les nombres entre 0 et $M - 1$. Cette promenade par tous les nombres possibles entre 0 et $M - 1$ n'assure pas que la suite est vraiment désordonnée, mais montre qu'on ne va pas osciller entre des valeurs trop proches, ce qui est déjà rassurant.

Il s'agit d'une propriété dite structurelle. Les spécialistes ont analysé d'autres propriétés structurelles identifiant des conditions mathématiques suffisantes pour qu'une suite engendrée par des générateurs congruentiels linéaires soit acceptable, c'est-à-dire produise des points assez équitablement distribués ou ne s'alignant pas trop quand on les représente dans des espaces de test.

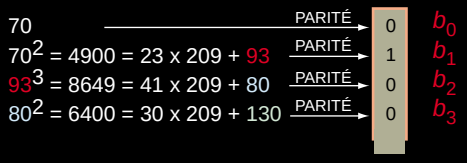
Dans une simulation utilisant n nombres pseudo-aléatoires, il est recommandé d'utiliser un générateur de période supérieure à n^2. On connaît des combinaisons de générateurs congruentiels linéaires engendrant des suites de période de 2^{200} et même plus. Récemment un générateur de période $2^{19937} - 1$ a été proposé. Un mot de prudence : il est moins risqué d'utiliser un générateur simple dont on connaît les propriétés qu'un générateur complexe pour lequel rien n'est démontré.

En plus des propriétés structurelles prouvées, on met en œuvre (sans cette fois chercher à faire de démonstrations mathématiques) des tests statistiques de toutes sortes portant sur les fréquences des différents chiffres et des suites de ces chiffres. Des jeux de tests ont ainsi été développés (en particulier, la série *Diehard* de George Marsaglia) et sont disponibles sur Internet pour qui souhaite tester la qualité d'une suite aléatoire : http://stat.fsu.edu/~geo/diehard.html

Reste que les spécialistes des générateurs aléatoires sont d'accord : même la démonstration de bonnes propriétés structurelles et le passage réussi à travers une volumineuse batterie de tests statistiques ne garantissent pas qu'un générateur donné sera satisfaisant pour une simulation donnée. Mathématiques et tests peuvent disqualifier une suite, mais pas en assurer la qualité.

4. LE PROCÉDÉ DE L. BLUM, M. BLUM ET M. SHUB (BBS)

Le générateur BBS est vraisemblablement satisfaisant en cryptographie. Soit n un produit de deux entiers premiers, chacun de la forme $4m + 3$. On prend comme graine un entier x sans facteur commun avec n et l'on calcule $x(0) = x^2 \bmod n$. On définit ensuite $x(i + 1) = x(i)^2 \bmod n$. La parité de $x(i)$ donne la suite pseudo-aléatoire de bits proposée par BBS.



Si la graine est choisie aléatoirement, et si l'hypothèse QRA est vraie (elle exprime que «trouver les racines carrées modulo n est un problème difficile») alors la suite de bits $b_0 b_1 \ldots b_m$ proposée par BBS est imprévisible (aucun algorithme rapide ne fait mieux que le hasard pour prédire le m-ième digit à partir des précédents). Il en résulte, d'après un résultat de A. Yao de 1982, que la suite de bits $b_0 b_1 \ldots b_m$ est indiscernable, par des tests fonctionnant en temps polynomial, d'une suite produite par une variable aléatoire équitable.

PRÉVISIBILITÉ

Cette situation est particulièrement dommageable en cryptographie, où l'on a souvent besoin de suites aléatoires et où le fait qu'une suite ne le soit pas vulnérabilise un procédé de codage. On raconte que, juste après la guerre, un algorithme de codage qui devait utiliser des clefs aléatoires toutes différentes fut utilisé deux fois de suite par les Russes avec la même clef, ce qui permit le déchiffrage des messages par les Américains et les conduisit à arrêter Klaus Fuchs, Julius et Ethel Rosenberg, Donald MacLean. L'officier russe responsable fut fusillé.

Les générateurs congruentiels linéaires sont depuis longtemps déconseillés en cryptographie. La raison tient à une série de résultats mathématiques qui indiquent qu'ils sont *prévisibles* : à partir de quelques points successifs de la suite $x(n)$, il est possible de calculer en un temps court les paramètres du générateur, et donc de prévoir son comportement futur avec une exactitude absolue, ce qui évidemment est contraire aux nécessités de la cryptographie. En cryptographie, conformément à l'idée que le hasard est imprévisible, il faut être certain que la suite aléatoire ne pourra pas être prédite *rapidement* à partir de la connaissance de quelques-uns de ses points.

«Rapidement», en informatique, signifie «ce qui peut être calculé en un temps plus petit qu'un polynôme de la taille des paramètres». Si la durée du calcul est inférieure à $n^3 + 2n^2 + 12$, où n est la taille des paramètres, on considère que le calcul est rapide. Si, en revanche, on ne sait faire un calcul qu'en un temps 2^n ou 3^n, etc., on considère que la longueur du calcul est inacceptable.

En utilisant cette idée, on arrive à une définition précise de la notion de «géné-

rateur aléatoire cryptographique sûr». Est sûr un générateur dont la période est trop grande pour être explorée exhaustivement, et tel qu'en plus les moyens exigés pour mener une analyse conduisant à en prévoir le comportement sont inacceptablement coûteux.

Il faut remarquer qu'en cryptographie on accepte parfois qu'un générateur ne soit pas uniforme (par exemple, qu'il donne 45 pour cent de '0' contre 55 pour cent de '1') s'il est imprévisible, alors qu'en simulation la situation est inverse : l'uniformité est plus importante que l'imprévisibilité.

Existe-t-il des générateurs cryptographiquement sûrs? Vraisemblablement oui, et le générateur BBS décrit sur la figure 4 est un bon candidat comme générateur cryptographiquement sûr. Pourquoi vraisemblablement? Parce que les mathématiciens ont seulement réussi à établir que le BBS est sûr en admettant qu'une certaine conjecture est vraie (il s'agit de la conjecture sur la difficulté du calcul de la *résiduosité quadratique*). La conjecture est jugée très vraisemblable, et donc on est presque certain que BBS est un bon générateur aléatoire... mais pas absolument.

Cette situation où un résultat, conjecturé et jugé vraisemblable par les spécialistes, reste non démontré est fréquente en théorie de la complexité. Bien d'autres conjectures attendent le génie mathématique qui saura les élucider. La situation est inconfortable : les cryptographes utilisent des méthodes vraisemblablement bonnes, sans avoir d'assurances fermes à leur sujet. Cette remarque s'applique en particulier aux systèmes utilisés pour la carte à puce ou les transactions entre banques.

Qui a dit qu'on faisait trop de mathématiques? La création de suites aléatoires sûres est un exemple parmi d'autres (qu'on trouverait en physique) où apparaît l'insuffisance du développement des mathématiques contemporaines face aux besoins de l'industrie. Même sur des sujets économiquement cruciaux, car liés au commerce électronique et à la sécurité bancaire, on attend des avancées mathématiques.

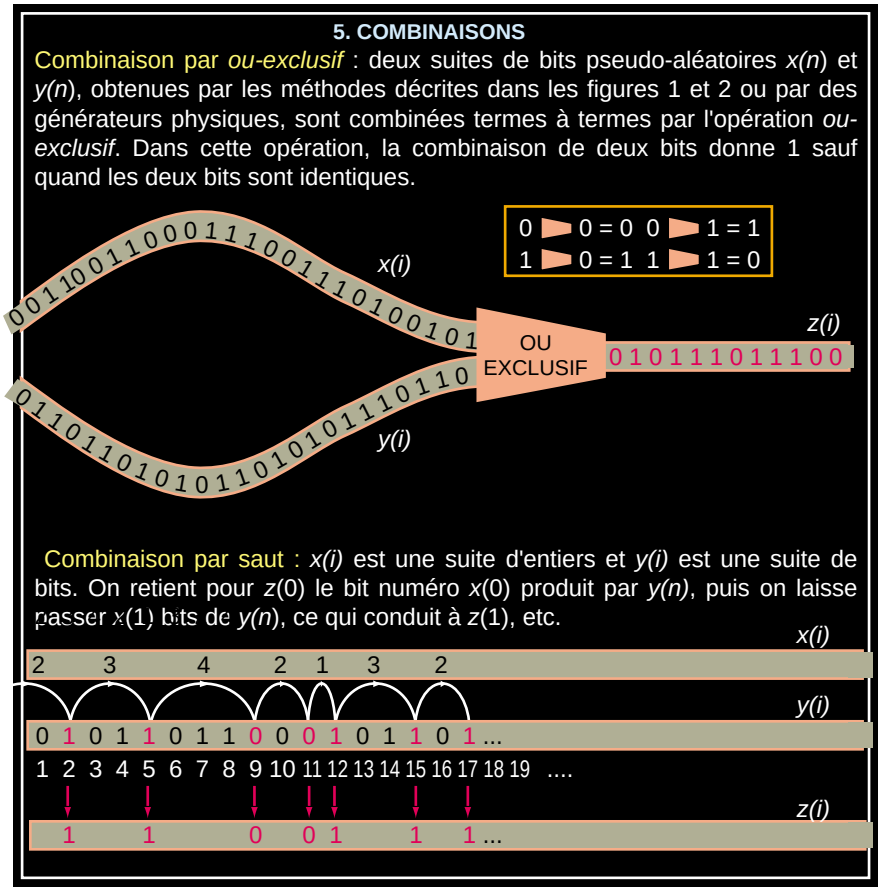
LE HASARD FORT

Ce hasard cryptographique (défini par la difficulté de la prédiction), sans doute atteint pas BBS, est le *hasard moyen*. Le *hasard fort* est défini par l'impossibilité absolue de la prédiction, quelle que soit la puissance de calcul utilisée. On a démontré, il y a quelques années, que cette impossibilité absolue est équivalente à l'impossibilité de la compression des données : une suite de nombres est aléatoire au sens fort s'il est impossible de la décrire en utilisant moins de place que celle qu'elle occupe.

Le malheur de ce hasard fort défini en 1965 par le mathématicien suédois Pier Martin-Löf est que, par nature, aucun programme déterministe d'ordinateur ne peut l'engendrer. Une procédure d'ordinateur (par nature déterministe) n'engendrera jamais de suites aléatoires au sens fort.

Dès 1951, avant même qu'elle n'ait pris un sens mathématique précis, von Neumann, conscient de cette difficulté fondamentale, écrivait : «Quiconque envisage une méthode arithmétique pour produire des chiffres aléatoires est en état de péché.»

À côté de l'impossibilité pour un programme déterministe d'engendrer une suite aléatoire au sens fort, on démontre que, si l'on réussit à disposer d'un phénomène physique non déterministe équiréparti (produisant des 0 et des 1 en proportion égale), alors en faisant travailler ce générateur on tombera presque certainement sur une suite aléatoire au sens de P. Martin-Löf (donc au sens le plus fort). Ce résultat semble être un jeu de mots – l'indéterminisme produit le hasard –, mais mathématiquement il lui correspond un énoncé précis qui n'est ni évident ni une boutade.



RÉALISATION DE TABLES DE NOMBRES ALÉATOIRES

Toutes ces raisons expliquent pourquoi on s'est souvent tourné vers des mécanismes physiques de production du hasard.

Il n'y a pas si longtemps, on éditait des tables de nombres aléatoires. En 1927, une table de 40 000 chiffres aléatoires fut publiée par L. Tippett. En 1939, un mécanisme physique fut utilisé par M. Kendal et B. Babington-Smith pour produire une table de 100 000 chiffres aléatoires. En 1955, la *RAND Corporation* publia une table d'un million de chiffres aléatoires engendrés, là encore, par une machine. Les tests statistiques utilisés pour mettre cette table à l'épreuve furent repris par J. Guilloud et M. Bouyer quand, en 1973, ils calculèrent un million de décimales de π. Le nombre π passa les tests. Cela ne signifie par que π est aléatoire au sens fort : le cryptographe saura reconnaître la suite des décimales de π et calculer les suivantes pour déchiffrer le message.

Les chiffres de π, contrairement à une légende tenace, ne conviennent donc ni comme hasard faible pour la simulation (trop compliqués à calculer), ni comme hasard moyen en cryptographie, où ils ne possèdent pas les propriétés d'imprévisibilité souhaitées.