PRNG :

Un algorithme est une suite d’instructions prédéfinies. C’est donc toujours déterministe. On aura donc toujours le même résultat en sortie. Afin de générer des nombres aléatoires, on utilise des algorithmes dont les résultats en sortie sont suffisamment imprévisibles pour considérer qu’ils sont aléatoires. On appelle ces nombres des nombres pseudo aléatoires, d’où le nom « Pseudo Random Number Generator » (PRNG).

En python, la bibliothèque random se base sur le Mersenne Twister, un générateur développé par Makoto Matsumoto et Takuji Nishimura en 1997, et plus particulièrement la variante MT19937. Elle a notamment l’avantage d’avoir une période de 219937-1 et d’etre plus rapide que la plupart des PRNG de même qualité. Cependant, il présente plusieurs défauts, notamment celui de la complexité linéaire du LFSR , révélé par l’algorithme TestU01.

Aujourd’hui, on a surtout besoin de l’aléatoire afin de chiffrer correctement les données. Le problème des nombres pseudo aléatoires, c’est qu’il est relativement facile de les prédire, car ce sont des algorithmes déterministes qui les produisent. Par exemple pour MT19937, l’algorithme se base sur les 624 derniers nombres générés pour produire le suivant. Si quelqu’un récupère ces 624 nombres, il serait en mesure de prédire la suite de la génération.

Reslultats : fréquence d’apparition de 1/0 autour de 50%

Vitesse d’oscillation moyenne entre les 1 et les 0 : 2 bits

Spectre :

