**Partie 2 : QRNG**

Afin de chiffrer correctement les données, nous avons aujourd’hui besoin d’avoir des générateurs de nombres aléatoires de plus en plus imprévisibles. 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.

Afin de palier au problème de prédictibilité des nombres pseudo-aléatoires, les scientifiques se sont peu à peu tournés vers l’informatique quantique. En effet, en mécanique quantique, certains phénomènes, comme la désintégration des atomes dans un réacteur nucléaire, sont complètements imprévisibles. Nous nous sommes donc intéressés à la génération de ces nombres aléatoires et nous avons comparé leurs caractéristiques statistiques par rapport aux nombres pseudo aléatoires.

Nous avons écrit un programme en python, et grâce à la bibliothèque Qiskit, nous avons pu utiliser les processeurs quantiques d’IBM afin de générer des séquences de nombres aléatoires. Nous avons aussi cherché quelques QRNG en ligne comme ANU (<http://qrng.anu.edu.au/index.php>), ou celui du Nano-optics groups du département de physique de l'université de Humboldt.

De même que pour l’algorithme en PRNG, nous allons mesurer la fréquence d’apparition, la vitesse d’oscillation et le spectre pour chaque QRNG. Nous allons ensuite les comparer aux résultats que nous avons trouvé dans la première partie pour pouvoir conclure sur la qualité de chaque générateur.