Pour hacher les nombres, on les écrit sur 512 bits (64 octets) en représentation "*big-endian*".

▷ **Schéma 1 :** On considère le générateur pseudo-aléatoire défini par :

$$X_{i+1} \leftarrow (2^{61} - 1) \cdot X_i \bmod \left(2^{127} - 1\right) \qquad \text{et} \qquad Y_i \leftarrow X_i \bmod 2^8$$

La graine du PRNG est X_0 (elle fait 127 bits), et la séquence de ses sorties est $Y_1, Y_2, \ldots$

▷ **Schéma 2 :** On considère le générateur pseudo-aléatoire défini par :

$$X_{i+1} \leftarrow {X_i}^2 \bmod \left(2^{1279} - 1\right) \qquad \text{et} \qquad Y_i \leftarrow \left\lfloor X_i / 2^{384} \right\rfloor$$

La graine du PRNG est X_0 (de taille 1279 bits), et la séquence de ses sorties est $Y_1, Y_2, \ldots$

▷ **Schéma 3 :** On considère la famille de fonctions à sens unique définie par :

$$\begin{array}{rcl} f_{a,b} : \{0,1\}^k \times \{0,1\}^k & \to & \mathbb{Z} \\ (x, y) & \mapsto & ax + by \end{array}$$

L'algorithme d'indexation produit deux entiers a, b de k bits, aléatoires et premiers entre eux.

▷ **Schéma 4 :** On considère la famille de fonctions de hachage résistante aux collisions définie par :

$$\begin{array}{rcl} f_{a,b} : \{0,1\}^k \times \{0,1\}^k & \to & \{0,1\}^k \\ (x, y) & \mapsto & ax + by \bmod 2^k \end{array}$$

L'algorithme d'indexation produit deux entiers a, b de k bits, aléatoires et premiers entre eux.

▷ **Schéma 5 :** On considère le générateur pseudo-aléatoire défini par :

$$X_i = \left((k+i)^{-1} \bmod (2^{607} - 1)\right) \bmod 2^{512}$$

La graine est k (de 607 bits) et la séquence de ses sorties est $X_1, X_2, \ldots$

▷ **Schéma 6 :** On considère le générateur pseudo-aléatoire défini par :

$$X_i = aX_i + b \bmod p \qquad \text{et} \qquad Y_i \leftarrow X_{2i}$$

La graine du PRNG est (a, b, p, X_0). p doit être un nombre premier d'au moins 128 bits, et tous les autres membres de la graine sont tirés aléatoirement modulo p. La séquence des sorties est $Y_1, Y_2, \ldots$

▷ **Schéma 7 :** On considère la famille de fonctions à sens unique définie par

$$\begin{array}{rcl} f_p : \mathbb{Z}_p^\times \times \mathbb{Z}_p^\times & \to & \mathbb{Z}_{p^3} \\ (x, y) & \mapsto & x/y \bmod p^3 \end{array}$$

L'algorithme d'indexation produit un entier p premier de n bits, où n est le paramètre de sécurité.

▷ **Schéma 8 :** On considère le générateur pseudo-aléatoire défini par :

$$X_i = k \times (x + \mathsf{SHA256}(i)) + \mathsf{SHA256}(x + i)$$

La graine du PRNG est (k, x), deux nombres d'au moins 384 bits. La séquence de ses sorties est $X_1, X_2, \ldots$.

▷ **Schéma 9 :** On considère le générateur pseudo-aléatoire défini par :

$$X_i = \mathsf{SHA256}(i) \times k + \mathsf{SHA256}(k + i)$$

La graine du PRNG est k, un nombre d'au moins 384 bits. La séquence de ses sorties est $X_1, X_2, \ldots$.

▷ **Schéma 10 :** On considère la fonction à sens unique définie par :

$$\begin{array}{rcl} f_A : \mathbb{Z}_{65537}^8 & \to & \mathbb{Z}^{12} \\ \mathbf{x} & \mapsto & \tau(\mathbf{x} \cdot A) \end{array}$$

L'entrée est un vecteur de 8 entiers (modulo 65537), et la matrice A est de taille 8×12. Le produit matrice-vecteur est réalisé modulo 65537. La fonction τ effectue la division euclidienne de chacune des coordonnées d'un vecteur par 16 :

$$\tau(\mathbf{x}) = (\lfloor \mathbf{x}_1/16 \rfloor, \ldots, \lfloor \mathbf{x}_n/16 \rfloor$$

L'algorithme d'indexation produit une matrice A de taille 8×12 à coefficients aléatoires modulo 65537.