Here, we summarize some key results on sampling discrete Gaussians over lattices. Much of the material is adapted from [Pei16, GPV08, Pei10, MP12].

**Discrete Gaussians over lattices.** Let \( \mathcal{L} = \mathcal{L}(B) \) be a lattice. The (spherical) discrete Gaussian distribution \( D_{c,s} \) on a lattice coset \( c + \mathcal{L} \) is simply the Gaussian distribution with parameter \( s \) and its support restricted to \( c + \mathcal{L} \). Namely, for \( x \in c + \mathcal{L} \),

\[
D_{c+\mathcal{L},s}(x) := \frac{\rho_s(x)}{\rho_s(c + \mathcal{L})} = \frac{\rho_s(x)}{\sum_{y \in c + \mathcal{L}} \rho_s(y)},
\]

and for \( x \notin c + \mathcal{L} \), \( D_{c+\mathcal{L},s}(x) = 0 \). This definition naturally extends to non-spherical Gaussians.

**Theorem 1 ([GPV08]).** There exists an efficient algorithm that takes as input a basis \( B \) for a lattice \( \mathcal{L} = \mathcal{L}(B) \), any coset \( c + \mathcal{L} \), and any width parameter \( s \geq \|B\| \cdot \omega(\sqrt{\log n}) \) and outputs a sample that is statistically close to \( D_{c+\mathcal{L},s} \).

**The SIS lattice.** For a matrix \( A \in \mathbb{Z}_q^{n \times m} \), the SIS lattice is defined as

\[
\mathcal{L}^+(A) := \{ x \in \mathbb{Z}_q^m : Ax = 0 \mod q \} \supseteq q\mathbb{Z}^m.
\]

For a vector \( u \in \mathbb{Z}_q^n \), we define

\[
\mathcal{L}^+_u(A) := \{ x \in \mathbb{Z}_q^m : Ax = u \mod q \} = z + \mathcal{L}^+(A),
\]

for some \( z \in \mathbb{Z}_q^m \) where \( Az = u \).

**Gadget trapdoors.** We say that \( R \in \mathbb{Z}_q^{m \times n} \) is a gadget trapdoor for \( A \in \mathbb{Z}_q^{n \times m} \) if \( AR = G \). We can take the following approach (from Micciancio and Peikert [MP12]) to sample from \( D_{c+\mathcal{L},s} \) using a gadget trapdoor for \( A \):

- Set \( s' = \omega(\sqrt{\log n}) \). Sample a perturbation vector \( p \leftarrow D_{\mathbb{Z}_q, s'2I_m} \). We can do this as long as \( s^2I_m - (s')^2RR^T \) is positive definite. Taking \( s = s_1(R) \cdot \omega(\sqrt{\log n}) \), where \( s_1(R) := \max_{\|u\|=1} \|Ru\| \) denotes the largest singular value of \( R \) suffices here. When \( s^2I_m - (s')^2RR^T \), we can decompose it as \( LL^T \) (e.g., by computing its Cholesky decomposition). Then, we can sample \( p \) by first sampling \( p' \leftarrow D_{\mathbb{Z}_q,1} \) (using Theorem 1) and setting \( p \leftarrow Lp' \).
The above algorithms satisfy the following properties:

- Let \( z \leftarrow u - Ap \). Sample \( y \leftarrow D_{L_u^\perp}(G)_{s'} \). Recall that \( G \) has a basis \( B \) where \( \|\hat{B}\| \leq \sqrt{5} \) (when \( q \) is a power of 2, \( \|\hat{B}\| = 2 \)), so we can use Theorem 1 to implement this step.

- Output \( x \leftarrow Ry + p \).

For correctness, observe that

\[
Ax = ARy + Ap = Gy + Ap = z + Ap = u,
\]

so \( x \in L_u^\perp(A) \). Consider the distribution of \( x \). The distribution of \( y \) is a discrete Gaussian with width \( s' \), so \( Ry \) is a discrete Gaussian with covariance \( (s')^2 R R^T \). The vector \( p \) is Gaussian with covariance \( s^2 I_m - (s')^2 R R^T \), so by the Gaussian convolution lemma (see [Pei10] for a precise description), the sum \( Ry + p \) is statistically close to a discrete Gaussian with covariance \( (s')^2 R R^T + (s^2 I_m - (s')^2 R R^T) = s^2 I_m \). This precisely coincides with the desired distribution \( D_{L_u^\perp(A),s} \). Refer to [MP12] for more details.

**Preimage sampleable trapdoor functions.** Using the above algorithm, we can construct a preimage sampleable trapdoor function as follows:

- \( \text{TrapGen}(n, q) \): On input lattice parameters \( n, q \), set \( \bar{m} = 3n \log q \), let \( t = n[\log q] \), and \( m = \bar{m} + t \). Sample \( A \leftarrow Z_q^{n \times \bar{m}} \) and \( R \leftarrow \{0, 1\}^{m \times t} \). Construct matrices

\[
A = [\hat{A} | G - AR] \in Z_q^{n \times m} \quad R = \begin{bmatrix} \bar{R} \\ I_t \end{bmatrix} \in Z_q^{m \times t}.
\]

Output the public matrix \( A \) and the trapdoor \( R \). Note that we can also sample \( R \) from other distributions to get smaller parameters; see [MP12].

- \( \text{SampleGaussian}(m, s) \): On input the dimension \( m \), sample and output \( x \leftarrow D_{Z_m, s} \) (e.g., using Theorem 1).

- \( \text{SamplePre}(A, R, u, s) \): On input the public matrix \( A \in Z_q^{n \times 2m} \), a trapdoor \( R \in Z_q^{2m \times m} \), and a target vector \( u \in Z_q^{2m} \), sample and output \( x \leftarrow D_{L_u^\perp(A),s} \) (using the procedure described above).

The above algorithms satisfy the following properties:

- Let \( (A, R) \leftarrow \text{TrapGen}(n, q) \). By the leftover hash lemma, the distribution of \( A \) is statistically close to uniform over \( Z_q^{n \times m} \). Since \( R \in \{0, 1\}^{m \times t} \), we can naïvely bound \( s_1(R) \) by \( \sqrt{mt} = O(n \log q) \).

- Let \( x \leftarrow \text{SampleGaussian}(m, s) \). If \( s \geq s_1(R) \cdot \omega(\sqrt{\log n}) \), then the distribution of \( Ax \) is statistically close to uniform over \( Z_q^n \). This follows from the fact that \( \eta(L_u^\perp(A)) \leq s_1(R) \cdot \omega(\sqrt{\log n}) \) (see [MP12, Lemma 5.3]) and the result shown from class.

- When \( s \geq s_1(R) \cdot \omega(\sqrt{\log n}) \), the following two distributions are statistically indistinguishable:

\[
\{ x \leftarrow \text{SampleGaussian}(m, s) : (x, Ax) \} \quad \text{and} \quad \{ y \leftarrow Z_q^n, x \leftarrow \text{SamplePre}(A, R, y, s) : (x, y) \}.
\]

**References**


