Lattice Cryptography

Ahmer Raza

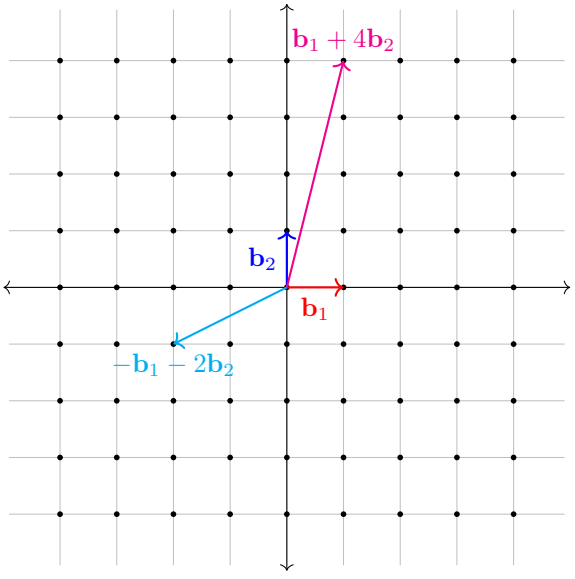
Lattices

First, we need to define the concept of a lattice.

> **Definition**: Let $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ be a set of linearly independent vectors in $\mathbb{R}^n$. The *lattice* generated by the basis B is defined as
>
> $$\mathcal{L}(B) = \left\{ \sum_{i=1}^{n} c_i \mathbf{b}_i : c_i \in \mathbb{Z} \right\}$$
>
> B can also be represented as a matrix $\begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \ldots & \mathbf{b}_n \end{pmatrix}$ whose columns form the basis.

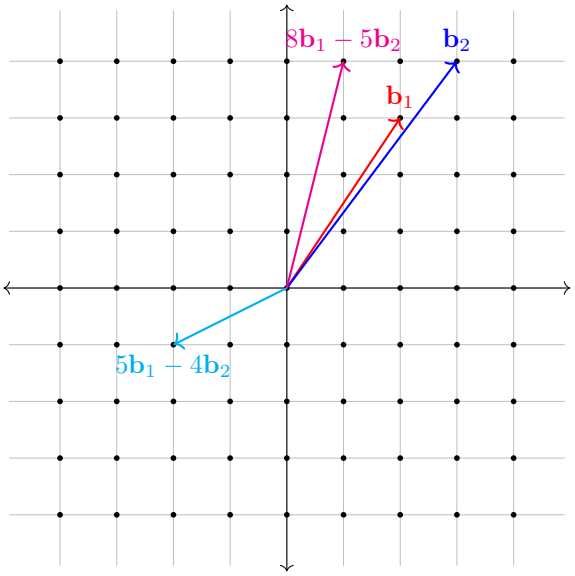
Note that choosing $c_i \in \mathbb{R}$ instead of $c_i \in \mathbb{Z}$ results in a vector space. In the two-dimensional case, we can visualize $\mathcal{L}(B)$ as a grid of points defined by B, and an element $\mathbf{x} \in \mathcal{L}(B)$ as a vector in this grid.

The $\mathbb{Z}^2$ lattice using $B_1 = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$



Importantly, bases are not unique. For example, in the above $\mathbb{Z}^2$ lattice, a different choice of B can still result in $\mathcal{L}(B) = \mathbb{Z}^2$.

The $\mathbb{Z}^2$ lattice using $B_2 = \left\{ \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \end{pmatrix} \right\}$



Intuitively, the first basis seems somehow better than the second. We say that a basis is *good* if its has short, nearly orthogonal basis vectors. The longer and more parallel basis vectors are, the worse the basis is. We can define a metric to quantify how orthogonal a basis is.

Definition: Let B be a basis for a lattice $\mathcal{L}(B)$. The *orthogonality defect* of B is the product of the basis vector lengths divided by the area/volume of the object they span:

$$\delta(B) = \frac{\prod_{i=1}^{n} \|b_i\|}{\sqrt{\det(B^T B)}}$$

We can see that $\delta(B) \geq 1$. If B is perfectly orthogonal, then $\delta(B) = 1$. The closer $\delta(B)$ is to 1, the better the basis.

We can calculate the orthogonality defects for both example $\mathbb{Z}^2$ bases, under the Euclidean norm $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}$.

$$\delta(B_1) = \frac{\sqrt{1^2 + 0^2} \cdot \sqrt{0^2 + 1^2}}{\sqrt{\det \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}} = 1$$

$$\delta(B_2) = \frac{\sqrt{2^2 + 3^2} \cdot \sqrt{3^2 + 4^2}}{\sqrt{\det \begin{pmatrix} 13 & 18 \\ 18 & 25 \end{pmatrix}}} = 5\sqrt{13} \approx 18.03$$

This confirms that the orthonormal basis B_1 is better than B_2. An interesting question is, how hard is it to find the best basis for an arbitrary lattice? If this isn't feasible, can we find a good enough basis?

Result (Lattice Reduction): Let B be a basis for a lattice $\mathcal{L}(B)$.

1. It is NP-hard to find the best basis B_0 having minimum $\delta(B_0)$.
2. The polynomial-time Lenstra–Lenstra–Lovász (LLL) algorithm can find a basis B_L such that $\delta(B_L) \leq 2^{n^2/2}$.
3. The block Korkine-Zolotarev (BKZ) algorithm can find a basis B_K such that $\delta(B_K) \leq n^n$. We do not have a good bound for its time complexity, but it is worse than LLL.

Lattice Problems

In this simple, two-dimensional case, it is easy to see that the lattice is $\mathbb{Z}^2$. However, in general, it is quite difficult to describe all the points in a lattice, other than noting that points in the lattice lattice are generated by the basis.

This property allows for lattice problems that are surprisingly hard to solve, both for conventional computers and quantum computers.

> **Problem** (Shortest Vector Problem): Let B be a basis for a lattice $\mathcal{L}(B)$. What vector in the lattice is closest to the origin, aside from the zero vector?

> **Problem** (Closest Vector Problem): Let B be a basis for a lattice $\mathcal{L}(B)$. What vector in the lattice is closest to a given point in $\mathbb{R}^n$?

While somewhat simple in two dimensions, the higher the dimension is, the harder SVP and CVP are to solve. Also, interestingly, the quality of a basis also impacts the difficulty of solving these problems.

> **Observation**: The SVP and CVP are much more difficult to solve with a bad basis than a good basis.

Some more advanced problems are currently used in major lattice-based schemes.

> **Problem** (Short Integer Solution): Let B be a basis for a lattice $\mathcal{L}(B)$. Let $A \in \mathbb{Z}_q^{n \times m}$ be a matrix, and let t be a small positive integer. Find a nonzero vector $\mathbf{x} \in \mathbb{Z}^m$ such that
>
> $$A\mathbf{x} = 0 \quad \text{and} \quad \|\mathbf{x}\| \leq t$$
>
> For some given norm $\|\cdot\|$.

> **Problem** (Learning With Errors): Let B be a basis for a lattice $\mathcal{L}(B)$. Let $A \in \mathbb{Z}_q^{n \times m}$ be a matrix. Given equation-solution pairs $(A, \mathbf{b}_i)$, decide whether
>
> $$\mathbf{b}_i = A\mathbf{s} + \mathbf{e}_i \quad \text{or} \quad \mathbf{b}_i \leftarrow \mathbb{Z}_q^n$$
>
> For some secret $\mathbf{s}$ and error $\mathbf{e}_i$. The notation $\mathbf{b}_i \leftarrow \mathbb{Z}_q^n$ means "$\mathbf{b}_i$ is uniformly random from $\mathbb{Z}_q^n$".

> **Result**: Computationally, solving LWE in the average case is as hard as solving SVP in the worst case.

Some other difficult lattice problems: GapSVP, GapCVP, Shortest Independent Vectors Problem (SIVP), Bounded Distance Decoding (BDD) problem.

GGH Cryptosystem

The GGH cryptosystem is a scheme based on the CVP. Although broken, it is still a good example of a simple lattice-based cryptosystem and is useful to look at.

Let B_1 be a good basis and let $B_2 = UB_1$ be a bad basis for the same lattice $L = \mathcal{L}(B_1) = \mathcal{L}(B_2)$, where U is some matrix. Both bases are in matrix form.

GGH Cryptosystem

1. Bob publishes B_2 as the public key, and keeps B_1 as the private key.
2. Alice selects a secret plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}^n$ that she wants to send to Bob. She also selects a small error vector $e \in \mathbb{Z}^n$.
3. Alice calculates $v = mB_2$. Note that $v \in L$. She then calculates her ciphertext $c = v + e$ and sends it to Bob.
4. Bob receives c from Alice and calculates $u = cB_1^{-1}$, which is
$$u = cB_1^{-1} = (mB_2 + e)B_1^{-1} = mB_2B_1^{-1} + eB_1^{-1} = mUB_1B_1^{-1} + eB_1^{-1} = mU + eB_1^{-1}$$
5. Bob removes the eB_1^{-1} error term by rounding to the closest lattice point (solving the CVP), which works since he knows the good basis. After rounding u to get $u' = mU$, he computes $m = u'U^{-1}$.

The key idea of GGH is that any individual aside from Bob only knows the bad basis. Hence, the CVP becomes difficult to solve. However, Bob knows the good basis, so he can round c to remove the error term.

Let us go back to our $\mathbb{Z}^2$ lattice example and walk through GGH. Note that the bad basis B_2 can be written in terms of the good basis B_1, since $B_1 = I_{2\times 2}$.

$$B_2 = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix} B_1, \quad U^{-1} = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}^{-1} = \begin{pmatrix} -4 & 3 \\ 3 & -2 \end{pmatrix}$$

1. Bob publishes the bad basis B_2 as the public key, while keeping the good basis B_1 as the private key.
2. Alice selects $m =$ "HI" $= (7, 8)$ as her message. Using the bad basis B_2, she calculates $v = mB_2$
$$v = (7, 8) \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix} = (38, 53)$$
3. Alice chooses a small error, say $e = (0.2, -0.3)$, calculates $c = v + e = (38.2, 52.7)$, and sends this ciphertext to Bob.
4. Bob receives c and calculates $u = cB_1^{-1}$
$$u = (38.2, 52.7) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} = (38.2, 52.7)$$
5. This is not a lattice point, so Bob rounds this to the closest lattice point, which is $u' = (38, 53)$. He then computes the plaintext $m = u'U^{-1}$
$$m = (38, 53) \begin{pmatrix} -4 & 3 \\ 3 & -2 \end{pmatrix} = (7, 8) = \text{"HI"}$$

Regev's LWE Public-Key Cryptosystem

The Regev cryptosystem is a more advanced scheme, and is based on the LWE problem. It is also quantum-safe, and thus is a great example of a straightforward, practical lattice-based cryptosystem.