CSE 206A: Lattice Algorithms and Applications Fall 2019

Random Lattices and Lattice-Based Cryptography

Instructor: *Daniele Micciancio* UCSD CSE

Lattice cryptography studies the construction of cryptographic functions whose security is based on the conjectured intractability of computationally hard lattice problems, like (variants of) the approximate closest vector problem (CVP_γ) and approximate shortest vector problem (SVP_γ). However, cryptographic constructions are typically expressed in terms of the following two problems that make to direct reference to lattices. The problems are parametrized by integers n, m, q and a set of short vectors $X \subseteq \mathbb{Z}^m$, e.g., the set $X = \{\mathbf{x} \mid \|\mathbf{x}\| \leq \beta\}$ of all vectors of euclidean norm bounded by β.

Definition 1 *The* Short Integer Solution *problem* **SIS***, on input a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$*, asks to find a nonzero vector* $\mathbf{x} \in X$ *such that* $\mathbf{Ax} = \mathbf{0} \pmod q$.

Definition 2 *The* Learning With Errors *problem* **LWE***, on input a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and a vector* $\mathbf{b}$*, asks to find a vector* $\mathbf{s} \in \mathbb{Z}_q^n$ *such that* $\mathbf{b} - \mathbf{A}^t\mathbf{s} \in X$.

Several variants of these problems exist, e.g., the following inhomogeneous version of **SIS**.

Definition 3 *The* Inhomogeneous SIS *problem* **SIS′***, on input a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and a vector* $\mathbf{b} \in \mathbb{Z}_q^n$*, asks to find a vector* $\mathbf{x} \in X$ *such that* $\mathbf{Ax} = \mathbf{b} \pmod q$.

Notice that $\mathbf{x}$ is required to be nonzero in **SIS** to avoid the trivial solution $\mathbf{A0} = \mathbf{0}$, but no such restriction is required in **SIS′**.

In cryptography, **SIS**, **LWE**, **SIS′** are usually studies (and used) as average-case problems, which requires to define approximate distributions on input instances. Usually,

- matrix $\mathbf{A}$ is chosen uniformly at random from $\mathbb{Z}_q^{n\times m}$
- In **LWE**, vector $\mathbf{b}$ is selected by picking $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random, $\mathbf{e} \leftarrow D_X$ from an appropriate distribution on X, and setting $\mathbf{b} = \mathbf{A}^t\mathbf{s} + \mathbf{e}$.
- In **SIS′**, vector $\mathbf{b}$ is set to $\mathbf{Ax}$ for a randomly chosen $\mathbf{x} \leftarrow D_X$.

The goal of these lecture is to show how these problems give raise to very simple cryptographic functions, how the problems relate to each other, and how they relate to lattices. As we will see, there are deep connections between these problems and lattices. For now, we will only interpret **SIS**, **LWE**, **SIS′** as average-case lattice problems on certain randomly distributed lattices. In a later lecture, we will explore the connection between **SIS**, **LWE**, **SIS′** and worst-case lattice problems.

1 Function Families

We begin by recalling the definition of some of the simplest cryptographic primitives: one-way functions and collision resistant hadh functions.

Most cryptographic schemes are typically described by *functions families*, i.e., collections of functions $f_k: X \to Y$ indexed by a set of keys $k \in K$. In cryptographic applications, these are usually required to be *one-way functions*, i.e., easy to compute (given the function key k and input x), but computationally hard to invert (given the key k and target value $y = f(x)$).

Definition 4 *A function family is a collection $\mathcal{F} = \{f_k: X \to Y\}_{k \in K}$ of functions with common domain X and range Y, indexed by a set of keys K, together with a (typically uniform) probability distribution over the keys $k \leftarrow K$.*

Often, the domain X is also endowed with a probability distribution $x \leftarrow X$. In order to formalize the notion of efficient computation in the asymptotic setting, one needs to consider sequences of function families $\mathcal{F}_n$ indexed by a security parameter n, where the domain X_n, codomain Y_n and key space K_n depend on n. Efficient computation is then identified with computations that can be carried out in time polynomial in n, and infeasible computations are those taking superpolynomial or exponential time in n. For simplicity, in what follows we fix the value of the security parameter n, and consider a single function family $\mathcal{F} = \{f_k: X \to Y\}_{k \in K}$, leaving the dependency on the security parameter n implicit.

We always require function families $\mathcal{F}$ to be efficiently computable, in the sense that the key distribution $k \leftarrow K$ is efficiently samplable, and there is an efficient evaluation algorithm that on input $k \in K$ and $x \in X$, outputs $f_k(x)$. We also assume that testing membership in the domain X, and sampling input values $x \leftarrow X$ can be done efficiently.

One can define several security properties for function families. Two of the most fundamental properties are collision resistance and one-wayness.

Definition 5 *A function family $\mathcal{F} = \{f_k: X \to Y\}$ is collision resistant if for any probabilistic polynomial time algorithm $\mathcal{A}$,*

$$\Pr\{x_1, x_2 \in X \wedge x_1 \neq x_2 \wedge f_k(x_1) = f_k(x_2) \mid k \leftarrow K, (x_1, x_2) \leftarrow \mathcal{A}(k)\} \leq \epsilon$$

for some negligible[1] *$\epsilon = n^{-\omega(1)}$.*

Injective functions satisfy the above definition trivially. So, the above definition is interesting only when $|X| > |Y|$, and therefore f_k cannot be injective.

Definition 6 *A function family $\mathcal{F} = \{f_k: X \to Y\}$ is one-way with respect to (efficiently samplable) input distribution X if for any probabilistic polynomial time algorithm $\mathcal{A}$,*

$$\Pr\{x' \in X \wedge f_k(x') = y \mid k \leftarrow K, x \leftarrow X, y = f_k(x), x' \leftarrow \mathcal{A}(k, y)\} \leq \epsilon$$

for some negligible function $\epsilon = n^{-\omega(1)}$.

[1] A function $\epsilon(n)$ is negligible if $\epsilon = n^{-\omega(1)}$. Typically, in cryptography, $\epsilon = 2^{-\Omega(n)}$ is exponentially small in the security parameter n.

Notice that in the definition of one-wayness the value x' output by the adversary $\mathcal{A}$ is not required to be the same as the one x chosen to compute y. This is to avoid trivial functions (e.g., the constant function $f(x) = 0$) to be considered one-way. Of course, when the functions f_k are injective over X, then the adversary $\mathcal{A}$ succeeds only if it recovers the original $x' = x$.

One last important security definition, that comes up when studying cryptographic primitives providing secrecy, is pseudorandomness.

Definition 7 *A function family $\mathcal{F} = \{f_k : X \to Y\}$ is a pseudorandom generator, with input distribution X, if the two distributions $D_0 = \{(k, f_k(x)) \mid k \leftarrow K, x \leftarrow X\}$ and $D_1 = \{(k, y) \mid k \leftarrow K, y \leftarrow Y\}$ are ϵ-indistinguishable for some negligible ϵ, i.e., for any probabilistic polynomial time algorithm $\mathcal{A}$,*

$$|\Pr\{\mathcal{A}(x) \mid x \leftarrow D_0\} - \Pr\{\mathcal{A}(x) \mid x \leftarrow D_1\}| \leq \epsilon$$

The **SIS′** and **LWE** problems are easily described as the problem of inverting a one-way function family. In both cases, the functions are indexed by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.

- The **SIS** function $f_{\mathbf{A}} : X \to \mathbb{Z}_q^n$ is defined as

  $$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \pmod{q}$$

  with input distribution D_X.

- The **LWE** function $g_{\mathbf{A}} : \mathbb{Z}_q^n \times X \to \mathbb{Z}_q^m$ is defined as

  $$g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \pmod{q}$$

  with input distribution $\mathcal{U}(\mathbb{Z}_q^n) \times D_X$.

Then, **SIS′** is precisely the problem of inverting one-way function $f_{\mathbf{A}}$, and **LWE** is the problem of inverting one-way function $g_{\mathbf{A}}$. Finally, for any set X, the **SIS** over $X' = X - X = \{\mathbf{x} - \mathbf{x}' \mid \mathbf{x}, \mathbf{x}' \in X\}$ is the problem of finding collisions to the SIS function $f_{\mathbf{A}}$ with domain X. In fact, by linearily, any $\mathbf{x} \neq \mathbf{x}' \in X$ such that $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{x}')$ also give a nonzero vector $\mathbf{x} - \mathbf{x}' \in X'$ such that $f_{\mathbf{A}}(\mathbf{x} - \mathbf{x}') = \mathbf{0}$.

2 Random Lattices and Duality

We have described **SIS**, **LWE**, **SIS′** as standard cryptographic problems of inverting a one-way function or breaking collision resistance. We now turn to connecting these problems as geometric problems on certain classes of random lattices.

Fix positive integers $n \leq m \leq q$, where n (and/or $m - n$) serves as the main security parameter. Typically m is a small multiple of n (e.g., $m = O(n)$ or $m = O(n \log n)$) and q is

a small prime with $O(\log n)$ bits. Notice that q is very small, not at all like the large primes (with $O(n)$ bits) used in number theoretic cryptography.[2] For any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ define

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\}$$

$$\Lambda_q(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x} = \mathbf{A}^T\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}_q^n\}.$$

Intuitively, $\Lambda_q(\mathbf{A})$ is the lattice generated by the rows of $\mathbf{A}$ modulo q, while $\Lambda_q^\perp(\mathbf{A})$ is the set of solutions of the system of n linear equations modulo q defined by the rows of $\mathbf{A}$. It is easy to check that $\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ are subgroups of $\mathbb{Z}^m$, and therefore they are lattices. In fact, they are full dimensional lattices because the contain m linearly independent vectors $(0, \ldots, 0, q, 0, \ldots, 0)$, with q ranging over all m coordinates.

Notice that the matrix $\mathbf{A}$ used to represent them is not a lattice basis. A lattice basis $\mathbf{B}$ for the corresponding lattices can be efficiently computed from $\mathbf{A}$ using linear algebra, but it is typically not needed: cryptographic operations are usually expressed and implemented directly in terms of $\mathbf{A}$.

Exercise 1 *Give an efficient algorithm that on input a matrix* $\mathbf{A}$*, computes a basis for the lattice* $\Lambda_q(\mathbf{A})$*.* (Hint: consider the integer matrix $[q\mathbf{I}, \mathbf{A}^t]$ and its HNF.)

Exercise 2 *Give an efficient algorithm that on input a matrix* $\mathbf{A}$*, computes a basis for the lattice* $\Lambda_q^\perp(\mathbf{A})$*.* (Hint: Use duality. If still unsure about what to do, read ahead, and then come back to this problem.)

Then, picking $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ uniformly at random defines two random class of m-dimensional integer lattices, denoted $\Lambda_q(m, n)$ and $\Lambda_q^\perp(m, n)$.

The relation between these lattices and the **SIS**, **LWE** cryptographic problems should be clear from the definition:

- **SIS** is the problem of finding a short nonzero vector in a random lattice selected according to $\Lambda_q^\perp(m, n)$.
- **LWE** is the problem of finding a lattice vector in a random lattice from $\Lambda_q(m, n)$ close to a given target $\mathbf{b}$.

In both problems, "short" or "close" is defined by the set X. If $X = \{\mathbf{x} \in \mathbb{Z}^m \mid \|\mathbf{x}\| \leq \beta\}$, then the goal is to find vectors of norm at most β, or at distance at most β from the target. Notice that these may not be the shortest of closest lattice vector (better solutions may exist), and this is why we referred to them simply as "short" and "close".

So, let us try to get a better understanding of these lattices. First of all, it immediately follows from the definition that these lattices are q-ary, i.e., they are periodic modulo q: one

[2] In fact, q is not even required to be prime, but for simplicity in these notes we will assume q is a small prime number.

can take the finite set $Q = L \cap [0, \ldots, q)^m$ of lattice points with coordinates in $\{0, \ldots, q-1\}$, and recover the whole lattice by tiling the space with copies $Q + q\mathbb{Z}^n$. This is a nice property to have, as it allows to describe the lattice as a finite set of points, a useful property in cryptographic applications. In fact, most (but not all) cryptographic operations based on these lattices can be implemented using finite arithmetics modulo q.

Next, we observe that $\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ are dual to each other, up to a scaling factor q.

Exercise 3 *Show that* $\Lambda_q(\mathbf{A}) = q \cdot \widehat{\Lambda_q^\perp(\mathbf{A})}$ *and* $\Lambda_q^\perp(\mathbf{A}) = q \cdot \widehat{\Lambda_q(\mathbf{A})}$. *In particular,* $\det(\Lambda_q(\mathbf{A})) \cdot \det(\Lambda_q^\perp(\mathbf{A})) = q^m$. *Moreover, for any* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, *we have* $\det(\Lambda_q^\perp(\mathbf{A})) \leq q^n$ *and* $\det(\Lambda_q(\mathbf{A})) \geq q^{m-n}$.

To better understand the relation between $\Lambda_q(m, n)$ and $\Lambda_q^\perp(m, n)$, it is convenient to define two auxiliary distributions. Let $\tilde{\Lambda}_q^\perp(m, n)$ be the conditional distribution of a lattice chosen according to distributions $\Lambda_q^\perp(m, n)$, given that the lattice has determinant exactly q^n. Similarly, let $\tilde{\Lambda}_q(m, n)$ be the conditional distribution of a lattice chosen according to $\Lambda_q(m, m-n)$, given that the determinant of the lattice is q^{m-n}. In both cases, when q is a prime, the condition is equivalent to requiring that the rows of $\mathbf{A}$ are linearly independent modulo q. An equivalent condition (valid for any q) is that $\mathbf{A}$ is primitive, i.e., it columns generate the whole space $\mathbf{A}\mathbb{Z}_q^m = \mathbb{Z}_q^n$. How much do these conditional distributions differ from the original ones? Not much.

Exercise 4 *Prove that for any* q *and* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, *the following conditions are equivalent:*

1. $\det(\Lambda_q^\perp(\mathbf{A})) = q^n$
2. $\det(\Lambda_q(\mathbf{A})) = q^{m-n}$
3. $\mathbf{A}$ *is primitive, i.e.,* $\mathbf{A}\mathbb{Z}_q^m = \mathbb{Z}_q^n$

Exercise 5 *Show that if* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *is chosen uniformly at random, then* $\Pr\{\det(\Lambda_q^\perp(\mathbf{A})) = q^n\} = \Pr\{\det(\Lambda_q(\mathbf{A})) = q^{m-n}\} = \Pr\{\mathbf{A}\mathbb{Z}_q^m = \mathbb{Z}_q^n\} \geq 1 - 1/q^{m-n}$. *Moreover, the conditional distributions* $\tilde{\Lambda}_q^\perp(m, n) = \tilde{\Lambda}_q(m, m-n)$ *are identical.*

So, for typical settings of the parameters (e.g., $m \geq 2n$), lattices chosen according to $\Lambda_q^\perp(m, n)$ or $\Lambda_q(m, m-n)$ have determinant q^n except with negligible probability $\epsilon \leq q^{-n}$, and the distributions $\Lambda_q^\perp(m, n)$ and $\Lambda_q(m, m-n)$ are almost identical because they are both statistically close to $\tilde{\Lambda}_q^\perp(m, n) = \tilde{\Lambda}_q(m, m-n)$.

We can now relate the **LWE** and **SIS′** problem. We claim that they are essentially different formulations of the same problem. For simplicity consider these problems with matrix $\mathbf{A}$ restricted to primitive matrices. (You can make this even easier by considering primitive matrices of the form $[\mathbf{I}, \bar{\mathbf{A}}]$.) Assume also that X is the set of vectors of norm bounded by β.

We have seen that **LWE** is the problem of finding a lattice point within distance at most β from a given target $\mathbf{b}$. Now, let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be the matrix definiting the LWE lattice

$\Lambda_q(\mathbf{A})$. We know that this lattice can also be represented as $\Lambda_q^\perp(\mathbf{H})$ for some $\mathbf{H} \in \mathbb{Z}_q^{(m-n)\times m}$. Consider the **SIS′** instance $(\mathbf{H}, \mathbf{Hb})$. A solution to this problem is precisely a vector $\|\mathbf{x}\| \leq \beta$ such that $\mathbf{Hx} = \mathbf{Hb}$. But then $\mathbf{H}(\mathbf{b}-\mathbf{x}) = 0 \pmod q$, i.e., the vector $\mathbf{v} = \mathbf{b}-\mathbf{x}$ belongs to $\Lambda_q^\perp(\mathbf{H}) = \Lambda_q(\mathbf{A})$. So, we have found a vector $\mathbf{v} \in \Lambda_a(\mathbf{A})$ in the LWE lattice within distance $\|\mathbf{b}-\mathbf{v}\| = \|\mathbf{x}\| \leq \beta$ from the target $\mathbf{b}$.

Exercise 6 *Give a similar reduction from* **SIS** *to* **LWE**. (Hint: use the fact that a generic solution to a inhomogeneous system $\mathbf{Hx} = \mathbf{b}$ can be written as the sum $\mathbf{x}_0 + \mathbf{x}_1$ between any specific solution $\mathbf{Hx}_1 = \mathbf{b}$, and a generic solution to the corresponding homogeneous system $\mathbf{Hx}_0 = \mathbf{0}$.)

This shows that (for appropriate parameters) **SIS′** and **LWE** are essentially the same problem, just using different notation. In fact, in cryptography, the two problems are typically used with very different parameters. Specifically, **LWE** typically use a much smaller value (relative to the lattice determinant) of β than **SIS′**, and this gives very different flavors to the two problems. In order to understand this difference we need to get a better understanding of the geometry of these random lattices, their successive minima $\lambda_1, \ldots, \lambda_m$ and covering radius μ.

Clearly, for any lattice $\Lambda_q(\mathbf{A})$ (or $\Lambda_q^\perp(\mathbf{A})$, we always have

$$1 \leq \lambda_1 \leq \lambda_m \leq q$$

and $\mu \leq \sqrt{m}q$ because Λ is an integer lattice with m linearly independent vectors of norm q. Moreover, from Minkowski's Theorem and Exercise 3, we know that $\lambda(\Lambda) \leq \sqrt{m}q^{n/m}$ for any $\Lambda \in \Lambda_q^\perp(m,n)$. This upper bound is essentially tight.

Exercise 7 *Prove that there is a constant $\delta > 0$ such that if Λ is chosen according to $\Lambda_q^\perp(m,n)$, then*

$$\Pr\{\lambda(\Lambda) < \delta\sqrt{n}q^{n/m}\} \leq 1/2^m.$$

[Hint: consider all integer vectors of norm at most $\delta\sqrt{m}q^{n/m}$ and use a union bound.]

What about the other parameters λ_m, μ? Also these parameters are very close to Minkowski's upper bound with high probability.

Exercise 8 *Prove that if Λ is chosen according to $\Lambda_q^\perp(m,n)$, then*

$$\Pr\{\frac{1}{\delta} \cdot \sqrt{m} \cdot q^{n/m} \leq 2\mu(\Lambda)\} \leq 1/2^m.$$

[Hint: Prove the bound for $\Lambda_q(m, m-n) \approx \Lambda_q^\perp(m,n)$ instead, and use duality and the transference theorems.]

In summary, all these distributions are essentially the same

$$\Lambda_q^{\perp}(m,n) \approx \tilde{\Lambda}_q^{\perp}(m,n) = \tilde{\Lambda}_q(m,m-n) \approx \Lambda_q(m,m-n)$$

and when a random lattice is chosen according to any of these distributions all the parameters $\lambda_1, \ldots, \lambda_m, \mu$ are within a (small) constant factor from Minkowski's bound $\sqrt{m}q^{n/m}$ with overwhelming probability.

We now see that depending on the value of β (relative to Minkowski's bound $\sqrt{m}q^{n/m}$) we get two very different types of problems:

- If $\beta \leq (\delta/2)\sqrt{m}q^{n/m}$, then there exists a most one solution, becuase β is less than half the minimum distance of the lattice. This is called the *Bounded Distance Decoding* problem BDD, and it is the setting typically used by **LWE**.
- If $\beta \geq (1/2\delta)\sqrt{m}q^{n/m}$, then there is always a solution, no matter how $\mathbf{b}$ is chose, because β is above the covering radius of the lattice. This is called the *Absolute Distance Decoding* problem (ADD, sometimes also called the Guaranteed Distance Decoding, GDD), and it is the setting most commonly described with **SIS** and **SIS**$'$.

In terms of function families, for the above setting of β, the **SIS** function $f_{\mathbf{A}}$ is surjective, while the **LWE** function $g_{\mathbf{A}}$ is injective, in both cases with high probability over the choice of $\mathbf{A}$.

3 Arbitrary Lattices

It is instructive to see how the **SIS**/**LWE** function can be generalized to use arbitrary lattices. Consider functions indexed by full dimensional lattices (e.g., represented by a basis), with the lattice dimension serving as the security parameter. Given a lattice basis $\mathbf{B} \in \mathbb{R}^{n \times n}$, one can define the function

$$f_{\mathbf{B}}(\mathbf{x}) = \mathbf{x} \bmod \mathbf{B} = \mathbf{B}(\lfloor \mathbf{B}^{-1}\mathbf{x} \rfloor).$$

In other words, $f_{\mathbf{B}}(\mathbf{x})$ rounds $\mathbf{x}$ to the corner of the fundamental parallelepiped $f_{\mathbf{B}}(\mathbf{x}) + \mathcal{P}(\mathbf{B})$ containing $\mathbf{x}$. As defined, $f_{\mathbf{B}}$ is a function with domain $X \subseteq \mathbb{R}^n$ and codomain $\mathcal{P}(\mathbf{B})$. Notice that without any restriction on the domain X, the function $f_{\mathbf{B}}(\mathbf{x}) = \mathbf{y}$ is easy to invert because $f_{\mathbf{B}}(\mathbf{y}) = \mathbf{y}$, so $\mathbf{y}$ itself is a valid preimage of $\mathbf{y}$. However, if we let the domain X of the function be a set of small vectors (say, the set $X = \mathcal{B}(r)$ of all vectors of length at most r), then inverting $\mathbf{y} = f_{\mathbf{B}}(\mathbf{x})$ corresponds to finding a lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ within distance r from $\mathbf{y}$, i.e., solving some version of the closest vector problem:

- In one direction, if there is an adversary $\mathcal{A}(\mathbf{B}, \mathbf{y}) = \mathbf{x}'$ that outputs a valid preimage of $\mathbf{y}$ (i.e. $\mathbf{x}' \in X$ and $f_{\mathbf{B}}(\mathbf{x}') = \mathbf{y}$), then we can easily compute a lattice vector $\mathbf{v} = \mathbf{y} - \mathbf{x}' \in \mathcal{L}(\mathbf{B})$ within distance $\|\mathbf{y} - \mathbf{v}\| = \|\mathbf{x}'\| \leq r$ from $\mathbf{y}$.
- Conversely, if we can find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ within distance r from $\mathbf{y}$, then we also have a valid preimage $\mathbf{x}' = \mathbf{y} - \mathbf{v}$ because $\|\mathbf{x}'\| = \|\mathbf{y} - \mathbf{v}\| \leq r$ and $f_{\mathbf{B}}(\mathbf{x}') = (\mathbf{y} - \mathbf{v}) \bmod \mathbf{B} = \mathbf{y}$.

We can think of $\mathbf{y} = f_{\mathbf{B}}(\mathbf{x})$ as the target from the coset $\mathbf{x}+\mathcal{L}(\mathbf{B})$ that makes the inversion problem computationally hardest. This is because given an arbitrary $\mathbf{y}' \in \mathbf{x}+\mathcal{L}(\mathbf{B})$, one can efficiently recover $\mathbf{y}$ by computing $\mathbf{y} = f_{\mathbf{B}}(\mathbf{y}') = f_{\mathbf{B}}(\mathbf{y}) = f_{\mathbf{B}}(\mathbf{x})$. So, if one can efficiently recover $\mathbf{x}$ given $\mathbf{y}$, then one can also recover it given $\mathbf{y}'$.

Notice also that while the definition of $f_{\mathbf{B}}$ depends on a basis $\mathbf{B}$, we can think of $f_{\mathbf{B}}(\mathbf{x})$ as computing a standard representative of the coset $\mathbf{x} + \Lambda$ under a known (but arbitrary) representation of a lattice $\Lambda = \mathcal{L}(\mathbf{B})$. This is so because given any $\mathbf{y} = f_{\Lambda}(\mathbf{x}) \in \mathbf{x} + \Lambda$, one can efficiently compute $f_{\mathbf{B}}(\mathbf{x}) = f_{\mathbf{B}}(\mathbf{y})$ for any specific basis $\mathbf{B}$, as long as $\mathbf{B}$ is known.

So, more abstractly, we can think of our function family as being defined as $f_{\Lambda}(\mathbf{x}) = \mathbf{x}+\Lambda$, where the key Λ is an n-dimensional lattice. Other representations are possible. For example, an equivalent way to define a concrete representation of these functions is to represent the lattice Λ by a dual basis $\mathcal{L}(\mathbf{D}) = \hat{\Lambda}$, and let $f_{\mathbf{D}}(\mathbf{x}) = \mathbf{D}^T\mathbf{x} \pmod 1 \in [0,1)^n$.

The **SIS** and **LWE** problems are special cases of this general construction, where the lattice is chosen according to distribution $\Lambda_q^{\perp}(m,n) \approx \Lambda_q(m, m-n)$, and represented by the matrix $\mathbf{A}$.

The typical use cases of **LWE** and **SIS** correspond to the following parameter settings respectively:

- $r < \lambda(\Lambda)/2$: this setting guarantees that there is at most one point within distance r from the target, making the function $f_{\Lambda}\colon \mathcal{B}(r) \to \mathbb{R}^n/\Lambda$ injective. In this regime of parameters, any point within distance r is necessarily the lattice point closest to the target. This is a restricted version of CVP called the Bounded Distance Decoding (BDD) problem: given a lattice Λ and target $\mathbf{y}$ within distance $r < \lambda/2$ from Λ, find the lattice point closest to $\mathbf{y}$.

- $r > \mu(\Lambda)$: this setting guarantees that there is always at least one lattice point within distance r from the target, making the function $f_{\Lambda}\colon \mathcal{B}(r) \to \mathbb{R}^n/\Lambda$ surjective. This is also an important version of CVP, called the Absolute Distance Decoding (ADD) problem: given a lattice Λ, target $\mathbf{y}$ and bound $r \geq \mu(\Lambda)$, find a relatively close lattice point within distance r from $\mathbf{y}$.