Lattice-based signature schemes

Sannidhya Shukla, Krishna Acharya, Dheeraj Pai

July 11, 2017

Abstract

Presently, most of the cryptographic schemes are based on the hardness of integer factorization or the discrete-log problem. It is assumed that there exists no polynomial-time algorithm which can factorize integers or find the discrete-log of an element in a finite field given its primitive root. However, in 1994, Shor presented a quantum algorithm which can solve both of these problems in polynomial time. Once large-scale quantum computers are built, all the major cryptographic schemes will become vulnerable. In particular, RSA (the standard for public-key cryptography, based on factorization problem) and DSA (the standard for digital signatures, based on discrete-log problem) will be affected.

However, there are some problems which would still remain intractable even in quantum domain. Some of these problems are based on lattices. There is a lot of current research on cryptographic primitives based on lattice problems. This project was an attempt to study and implement some of these primitives, in particular, signature-schemes.

Acknowledgement

This project was done under supervision of Prof.SenGupta (Indian Statistical Institute, Kolkata), we are grateful for his guidance. We also thank Prof.Rishiraj Bhattacharyya(NISER, Bhubaneswar) for helping us understand the technicality of the proofs involved in BLISS. We would also like to thank Microsoft Research India for supporting the program and the RC Bose centre for Crytpology and Security for hosting us at ISI-Kolkata.

1 Definitions and Propositions

This section will give formal definitions of terms related to lattices and some relevant results. It is based on the introductions to lattices given in [1] and [2].

Definition. An n-dimensional lattice, $\mathcal{L}$ is a discrete, additive subgroup of $\mathbb{R}^n$.

1. Additive subgroup: $\mathbf{0} \in \mathcal{L}$ and $\forall\ \mathbf{x}, \mathbf{y} \in \mathcal{L}$, we have, $-\mathbf{x}, \mathbf{x}+\mathbf{y} \in \mathcal{L}$.
2. Discrete: $\forall \mathbf{x} \in \mathcal{L}, \exists\ \epsilon > 0$, such that, $\mathcal{L} \cap \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{x}-\mathbf{y}\| < \epsilon\} = \{\mathbf{x}\}$

Another equivalent definition is:

Definition. Let $B = \{\mathbf{b}_i \in \mathbb{R}^n : 1 \leq i \leq n\}$ be a set of n linearly independent vectors. An n-dimensional lattice, $\mathcal{L}$ having basis B is the set of all the integer linear combinations of the elements of B. Thus,

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} a_i \mathbf{b}_i : a_i \in \mathbb{Z} \right\}$$

A consequence of this definition is that any lattice in $\mathbb{R}^n$ is isomorphic to $\mathbb{Z}^n$. Note that we shall denote the basis of a lattice by the set of basis vectors or by the matrix whose rows are basis vectors, depending on the context.

Proposition. If B and C are two basis of an n-dimensional lattice. Then, they are related by a unimodular integer matrix. That is, $C = AB$, where $A \in \mathbb{Z}^{n\times n}$ and $\det A = \pm 1$.

Proof. Let the set of basis vectors for B and C be $\{\mathbf{b}_i : 1 \leq i \leq n\}$ and $\{\mathbf{c}_i : 1 \leq i \leq n\}$ respectively. Then, we can write each $\mathbf{c}_i$ as an integer linear combination of the basis vectors of B, $\mathbf{c}_i = \sum_{j=1}^{n} a_{ij}\mathbf{b}_j$, where $a_{ij} \in \mathbb{Z}$, $0 \leq i, j \leq n$. In matrix notation, $C = AB$, where $A = [a_{ij}]_{n\times n}$.

Similarly, $\mathbf{c}_i = \sum_{j=1}^{n} a_{ij}\mathbf{b}_j$, where $a'_{ij} \in \mathbb{Z}$, $0 \leq i, j \leq n$. In, matrix notation, $B = A'C$, where, $A = [a'_{ij}]_{n\times n}$. Clearly, $A' = A^{-1}$.

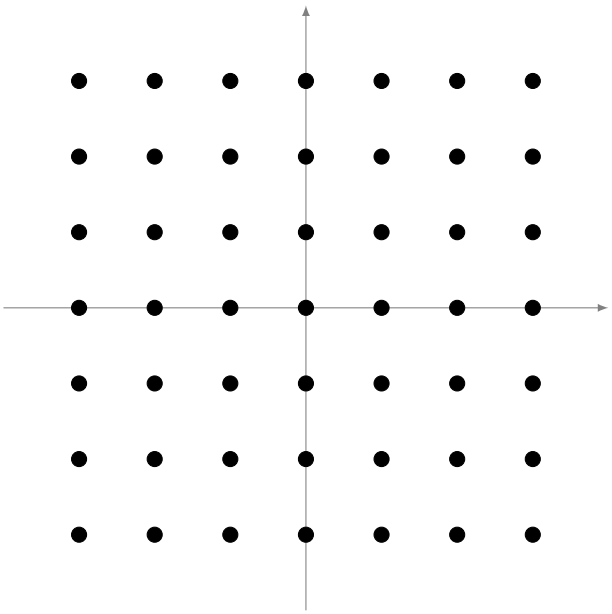
Now, $(\det A)(\det A^{-1}) = \det(AA^{-1}) = \det I = 1$. But, both A and A^{-1} have integer entries, which means, both $\det A$ and $\det A^{-1}$ are integers. Hence, $\det A = \pm 1$. QED.

A visual inspection of a lattice indicates that it is a periodic and infinite repetition of a "simpler structure". We formalize this structure as follows.

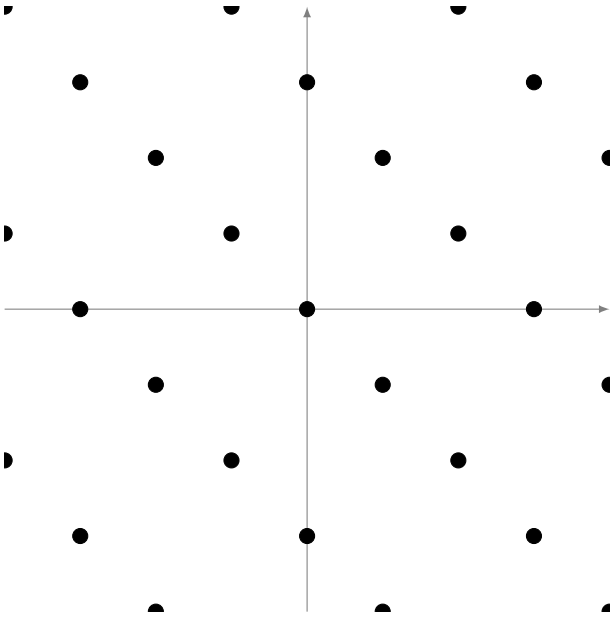
Definition. Let $\mathcal{L}$ be an n-dimensional lattice with basis $B = \{\mathbf{b}_i : 1 \leq i \leq n\}$. The fundamental domain (or fundamental parallelopiped) of $\mathcal{L}$ corresponding to basis B is the set:

$$\mathcal{F}(B) = \left\{ \sum_{i=1}^{n} a_i \mathbf{b}_i : a_i \in [0, 1) \right\}$$

Example. $\mathbb{Z}^2$ is the simplest lattice in $\mathbb{R}^2$ (having the standard basis I_2).



Example. A lattice in $\mathbb{R}^n$ with the basis $\{(1,2),(2,1)\}$.



Proposition. Let $\mathcal{L}$ be an n-dimensional lattice with basis $B = \{\mathbf{b}_i : 1 \leq i \leq n\}$ and let $\mathcal{F}$ be a fundamental domain of $\mathcal{L}$. Then, for all, $\mathbf{w} \in \mathbb{R}^n$, there are unique vectors $\mathbf{u} \in \mathcal{F}$ and $\mathbf{v} \in \mathcal{L}$, such that, $\mathbf{w} = \mathbf{u} + \mathbf{v}$.

Proof. Since, B is a basis, we can write, $\mathbf{w}$ as an integer linear combination of

all $\mathbf{b}_i$. Let, $\mathbf{w} = \sum_{i=1}^{n} a_i \mathbf{b}_i$, where, $a_i \in \mathbb{R}$. Then,

$$\mathbf{w} = \sum_{i=1}^{n} \lfloor a_i \rfloor \mathbf{b}_i + \sum_{i=1}^{n} (a_i - \lfloor a_i \rfloor) \mathbf{b}_i$$

Put $\mathbf{u} = \sum_{i=1}^{n} (a_i - \lfloor a_i \rfloor) \mathbf{b}_i$ and $\mathbf{v} = \sum_{i=1}^{n} \lfloor a_i \rfloor \mathbf{b}_i$. Clearly, $\mathbf{v} \in \mathcal{L}$ and since $0 \leq (a_i - \lfloor a_i \rfloor) < 1$, we have, $\mathbf{u} \in \mathcal{F}$ (by definition of $\mathcal{F}$). QED.

Definition. Let $\mathcal{L}$ be an n-dimensional lattice and $\mathcal{F}$ be one of its fundamental domains. The determinant of $\mathcal{L}$ (denoted by $\det \mathcal{L}$) is defined as the n-volume of $\mathcal{F}$.

We now state an important result, which is useful for finding the n-volume of a fundamental domain. The proof involves multivariable calculus. Refer [2].

Proposition. Let $\mathcal{L}$ be a lattice having basis B. Then, $\det \mathcal{L} = |\det B|$.

Corollary. Let $\mathcal{L}$ be a lattice. Every fundamental domain for $\mathcal{L}$ has same n-volume. Hence $\det \mathcal{L}$ is an invariant of the lattice, independent of the basis.

Proof. Let B and C be two basis for the lattice $\mathcal{L}$, then,

$$\begin{aligned} \det \mathcal{L} &= |\det B| && \text{(using Proposition 3.)} \\ &= |(\det A)(\det C)| && \text{(from Proposition 1.)} \\ &= |(\pm 1)(\det C)| && \text{(since } \det A = \pm 1\text{)} \\ &= |\det C| \end{aligned}$$

QED.

It is obvious that the volume of a fundamental domain is maximum when its basis vectors are pairwise orthogonal. We formalize this result through the following inequality.

Proposition (Hadamard's Inequality). Let $\mathcal{L}$ be a lattice. Let $B = \{\mathbf{b}_i : 1 \leq i \leq n\}$ be one of its basis. Then,

$$\det \mathcal{L} \leq \prod_{i=1}^{n} \|\mathbf{b}_i\|$$

The ratio $\dfrac{\det \mathcal{L}}{\prod \|\mathbf{b}_i\|}$, called the Hadamard's ratio, indicates how orthogonal the basis is. The closer this ratio is to 1, the closer the basis is to being orthogonal.

2 Lattice Problems

In this section we shall look at some of the hard problems based on lattices. The security proof of any lattice-based cryptosystem involves reduction to these problems.

Definition. The minimum distance of a lattice $\mathcal{L}$ is the shortest non-zero vector in the lattice.

$$\lambda_1(\mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|v\|$$

Or more generally, the n^{th} successive minimum, λ_n is the radius of the smallest n-sphere that contains n linearly independent vectors.

2.1 Problems

Most of the lattice problems involve minimizing the norm between two lattice points or between a lattice and a non-lattice point. All the norms expressed here are Euclidean Ł$_2$-norms unless stated otherwise.

Shortest Vector Problem (SVP)
Given a lattice $\mathcal{L}$, find a shortest lattice vector $\mathbf{v} \in \mathcal{L}$, *ie.*, find a lattice vector $\mathbf{v} \in \mathcal{L}$, such that, $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$.
Note that there can be more than one shortest vector in a lattice. For example, in $\mathbb{Z}^2$, $(\pm 1, 0)$ and $(0, \pm 1)$ are all the shortest vectors. Thus, there are four solutions to SVP in this case.

Approximate Shortest Vector Problem (SVP$_\gamma$)
Given an n-dimensional lattice $\mathcal{L}$, and γ (a function of n), find a non-zero lattice vector, $\mathbf{v} \in \mathcal{L}$, such that, $\|\mathbf{v}\| \leq \gamma(n)\lambda_1(\mathcal{L})$, where γ is a function of the dimension n of the lattice.

Closest Vector Problem (CVP)
Given a lattice $\mathcal{L}$, and a non-lattice vector $\mathbf{w} \in \mathbb{R}^n \setminus \mathcal{L}$, find a lattice vector closest to $\mathbf{w}$, *ie.*, find a lattice vector $\mathbf{v}$, such that, $\|\mathbf{v} - \mathbf{w}\|$ is minimum.

Approximate Closest Vector Problem (CVP$_\gamma$)
Given a lattice $\mathcal{L}$, and a non-lattice vector $\mathbf{w} \in \mathbb{R}^n \setminus \mathcal{L}$, find a lattice vector $\mathbf{v} \in \mathcal{L}$, such that,

$$\|\mathbf{v} - \mathbf{w}\| \leq \gamma(n) \min_{\mathbf{u} \in \mathcal{L}} \|\mathbf{u} - \mathbf{w}\|$$

where γ is a function of the dimension n of the lattice.

Shortest Integer Solution problem (SIS)
Given a matrix $A \in \mathbb{Z}_q^{n \times m}$. Find a vector $\mathbf{x}$, such that $A\mathbf{x} = 0$ and $0 < \|\mathbf{x}\| \leq \beta$.

Without constraints on norm of the solution, $\|\mathbf{x}\|$, it is easy to find one using Gaussian elimination. Further, we require that $\beta < q$, otherwise, $(q, 0, 0, ...) \in \mathbb{Z}^m$ will be a possible trivial solution. Using Pigeonhole principle it is easy to show that for existence of a solution, $\beta \geq \sqrt{\lceil n \log q \rceil}$ and $m \geq \lceil n \log q \rceil$.

2.2 Complexities

Both the SVP and CVP are known to be computationally intractable under certain assumptions. CVP is known to be NP-hard. The general SVP is known to be NP-hard under randomized reductions [6]. The SVP for L_∞-norm directly reduces to the subset-sum problem and hence is NP-hard [7].

3 Theorems on Short Vectors

The study of lattice is an old one, with works dating back to early 19^{th} century. With the most important being those of Minkowski and Hermite. We state and prove (non-rigorously) some of these theorems.

3.1 Blichfeldt's Theorem

This theorem can be thought of as an extension of *Pigeonhole principle*.
Theorem. Let $\mathcal{L}$ be a lattice in $\mathbb{R}^n$. Let S be a bounded subset of $\mathbb{R}^n$, such that, n-volume of S is greater than $\det \mathcal{L}$. Then, there exist two vectors $\mathbf{x}$ and $\mathbf{y}$ in S, such that their difference is a non-zero lattice vector, *ie.*, $\mathbf{x} - \mathbf{y} \in \mathcal{L}$.

3.2 Minkowski's Convex Body Theorem

Theorem. Let $\mathcal{L}$ be a lattice in $\mathbb{R}^n$, and S be a *bounded*, *convex*, *symmetrical* subset of $\mathbb{R}^n$, such that, n-volume of S atleast $2^n \det \mathcal{L}$, then, S contains atleast one non-zero lattice vector.

Proof. Case 1 (vol $S < \det \mathcal{L}$): We define $\frac{1}{2}S$ as $\{\frac{1}{2}\mathbf{a} : \mathbf{a} \in S\}$. Clearly, $vol(\frac{1}{2}S) = \frac{1}{2^n} vol(S) \geq \det \mathcal{L}$. Applying Blichfeldt's Theorem, we get, $\mathbf{x}$, $\mathbf{y} \in \frac{1}{2}S$, such that, $\mathbf{x} - \mathbf{y} \in \mathcal{L} (\mathbf{x} \neq \mathbf{y})$. By definition of $\frac{1}{2}S$, we have, $2\mathbf{x}$, $2\mathbf{y} \in S$. Since S is symmetric, $-2\mathbf{y} \in S$. Finally, since S is convex, the midpoint of line segment joining any two points in S is also in S. Thus,

$$\frac{2\mathbf{x} + (-2\mathbf{y})}{2} = \mathbf{x} - \mathbf{y} \in S$$

.

Case 2 (vol $S = \det \mathcal{L}$). We expand the set S, by the factor $1 + \frac{1}{k}$, $k \geq 1$, the resultant set will have n-volume greater than $2^n \det \mathcal{L}$. We then use the result from the case 1, to find a non-zero lattice vector $\mathbf{v}_k$, such that,

$$\mathbf{v}_k \in \left(1 + \frac{1}{k}\right) S$$

Each of the vector $\mathbf{v}_k$ is in the bounded set $2S$. Since $\mathcal{L}$ is discrete, there are only finite number of distinct $\mathbf{v}_k$. We choose some vector $\mathbf{v}$ which occurs infinitely often the sequence $\mathbf{v}_k$. Thus, we have found a non-zero lattice vector in the intersection

$$\bigcap_{k=1}^{\infty} \left(1 + \frac{1}{k}\right) S = S$$

QED.

Minkowski's theorem can be used to find the upper bound on the shortest vector in a lattice.

Example. Given an n-dimensional lattice $\mathcal{L}$, $\lambda_1(\mathcal{L}) \leq \sqrt{n} \det \mathcal{L}^{\frac{1}{n}}$.

Proof. Let S be $\mathbf{0}$ centered hypercube in $\mathbb{R}^n$, having side length $2a$, thus, $S = \{(x_1, x_2, ..., x_n) \in \mathbb{R}^n : -a \leq x_i \leq a, 1 \leq i \leq n\}$.

Clearly, S is symmetric and convex. The volume of S is $\text{vol}S = (2a)^n$. Put $a = \det \mathcal{L}^{\frac{1}{n}}$, then $\text{vol}S = 2^n \det \mathcal{L}$. Using Minkowski's theorem, we find a non-zero lattice vector $\mathbf{v} \in S \cap \mathcal{L}$. Let $\mathbf{v} = (v_1, v_2, ..., v_3)$. Because of how we have defined S, each of v_i, will satisfy $|v_i| \leq a$. Thus, $\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{n} v_i^2} \leq \sqrt{\sum_{i=1}^{n} a^2} = \sqrt{n}a = \sqrt{n} \det \mathcal{L}^{\frac{1}{n}}$. Hence, $\lambda_1(\mathcal{L}) \leq \sqrt{n} \det \mathcal{L}^{\frac{1}{n}}$.

This is a special case of the Hermite's theorem, which states that every n-dimensional lattice $\mathcal{L}$ has a vector $\mathbf{v}$, satisfying $\|\mathbf{v}\|^2 \leq \gamma_n \det \mathcal{L}^{\frac{2}{n}}$, where the constant γ_n is obtained by applying Minkowski's theorem on various convex, symmetric subsets of $\mathbb{R}^n$. In the case of hypercube, we obtain $\gamma_n = n$. We can further tighten this bound by applying Minkowski's theorem on hypersphere.

Example. Given an n-dimensional lattice $\mathcal{L}$, $\lambda_1(\mathcal{L}) \leq \sqrt{\frac{2n}{\pi e}} \det \mathcal{L}^{\frac{1}{n}}$.

Proof. Let $S \subset \mathbb{R}^n$ be the hypersphere of radius r. Then,

$$\text{vol}(S) = \frac{\pi^{n/2} r^n}{\Gamma(1 + n/2)}$$

For sufficiently large values of n, we can use Stirling's approximation to get

$$\text{vol}(S) \approx \left(\frac{2\pi e}{n}\right)^{n/2} r^n$$

Put $r = \sqrt{\frac{2n}{\pi e}} \det \mathcal{L}^{\frac{1}{n}}$, then,

$$\text{vol}(S) \approx \left(\frac{2\pi e}{n}\right)^{n/2} \left(\sqrt{\frac{2n}{\pi e}} \det \mathcal{L}^{\frac{1}{n}}\right)^n = 2^n \det \mathcal{L}$$

By Minkowski's theorem, S contains atleast one lattice vector $\mathbf{v}$, such that, $\|\mathbf{v}\| \lessapprox \sqrt{\frac{2n}{\pi e}} \det \mathcal{L}^{\frac{1}{n}}$.

3.3 Gaussian heuristic

In the last section we saw how the upper bound on the shortest vector in a lattice can be tighten. However, the exact bounds for the shortest vector for large dimensions are unknown. But, the size of the shortest vector can be estimated by a simple probabilistic argument: the number of lattice points in a *large* hypersphere is approximately equal to the n-volume of the hypersphere divided by the determinant of the lattice.

Let $S \in \mathbb{R}^n$ be a hypersphere of radius r and $\mathcal{L} \in \mathbb{R}^n$ be a lattice. Assuming the dimension n and the radius r to be sufficiently large, we can make the following estimation about the number of lattice points in the hypersphere:

$$\#\{S \cap \mathcal{L}\} \approx \frac{\text{vol}(S)}{\det \mathcal{L}}$$

Applying Stirling's approximation to the volume of a hypersphere, we get

$$\#\{S \cap \mathcal{L}\} \approx \frac{\left(\frac{2\pi e}{n}\right)^{n/2} r^n}{\det \mathcal{L}}$$

The shortest vector will be contained in the hypersphere which has just one lattice point, *ie.*, $\#\{S \cap L\} = 1$. Solving for r, we get:

$$r = \sqrt{\frac{n}{2\pi e}} \det \mathcal{L}^{\frac{1}{n}}$$

This is the expected length of the shortest vector in $\mathcal{L} \subset \mathbb{R}^n$ and is called *Gaussian expected shortest length* and is denoted by $\sigma(\mathcal{L})$. The Gaussian heuristic says that the length of the shortest vector in a *'randomly chosen lattice'*, with *sufficiently large* n, will satisfy

$$\lambda_1(\mathcal{L}) \approx \sigma(\mathcal{L})$$

4 Solving CVP

Consider an n-dimensional lattice $\mathcal{L} \subset \mathbb{R}^n$ having having the basis $\{\mathbf{b}_i : 1 \leq i \leq n\}$, such that the basis vectors are pairwise orthogonal, *ie.*, $\mathbf{b}_i \cdot \mathbf{b}_j = 0 \; \forall \; i \neq j$.

Suppose we want to solve CVP for a vector

$$\mathbf{w} = \sum_{i=1}^{n} w_i \mathbf{b}_i$$

Let $\mathbf{v} \in \mathcal{L}$ be one possible solution, where

$$\mathbf{v} = \sum_{i=1}^{n} v_i \mathbf{b}_i, v_i \in \mathbb{Z}$$

Then solving CVP, would mean minimizing $\|\mathbf{v} - \mathbf{w}\|$.

$$\|\mathbf{v} - \mathbf{w}\|^2 = \sum_{i=1}^{n} (v_i - w_i)^2 \|\mathbf{b}_i\|^2$$

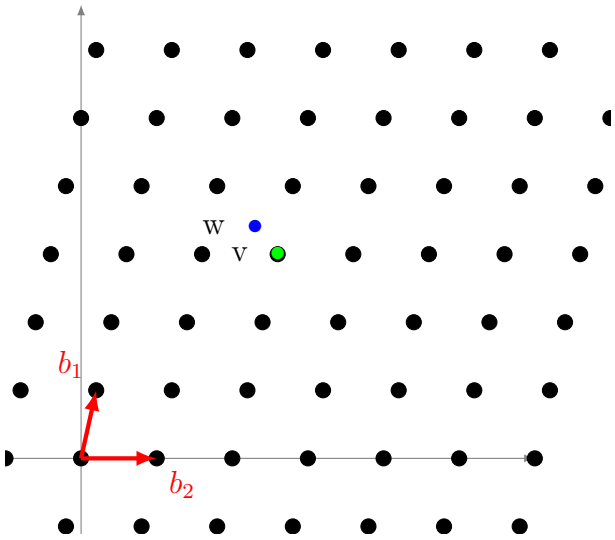
Since, $v_i \in \mathbb{Z}$, $\|\mathbf{v} - \mathbf{w}\|^2$ will be minimum when v_i are chosen to be the integers nearest to w_i. This procedure is simple but is applicable only when the basis is orthogonal, or *near* orthogonal. This procedure was proposed by Lazslo Babai in 1982, and is known as Babai's algorithm. We now formally describe this algorithm. When this procedure is applied on a basis which is not orthogonal, it leads to erroneous results. We shall call the bases which are pairwise orthogonal or *near* orthogonal as *'good'* bases and those which are far from being pairwise orthogonal as *'bad'* bases. As noted earlier, how close a basis is to being orthogonal can be measured from its Hadamard ratio.

Algorithm 1: Babai's algorithm

procedure SolveCVP
Input: A *good* basis $B = \{\mathbf{b}_i : 1 \leq i \leq n\}$, a non-lattice point $\mathbf{w} \in \mathbb{R}^n$
Output: A point $\mathbf{v} \in \mathcal{L}$, $\|\mathbf{v} - \mathbf{w}\|$ is minimum
    Express $\mathbf{w}$ in basis B: $\sum_{i=1}^{n} w_i \mathbf{b}_i \leftarrow \mathbf{w}$
    $\mathbf{v} \leftarrow \sum_{i=1}^{n} \lfloor w_i \rceil \mathbf{b}_i$
    return $\mathbf{v}$

Example. Solving CVP using *good* basis $\{(1, 0), (0.2, 0.9)\}$ for point $(2.3, 3.06)$ (expressed in standard basis) using Babai's algorithm.

Using Gaussian elimination, we first express the vector $\mathbf{w} = (2.3, 3.06)$ in the basis B. We find that $\mathbf{w} = 1.62(1, 0) + 3.4(0.2, 0.9)$. Using Babai's algorithm, the nearest lattice point (expressed in basis B) is $(\lfloor 1.62 \rceil, \lfloor 3.4 \rceil) = (2, 3)$. In standard basis, $\mathbf{v} = 2(1, 0) + 3(0.2, 0.9) = (2.6, 2.7)$ (marked by green in the plot).

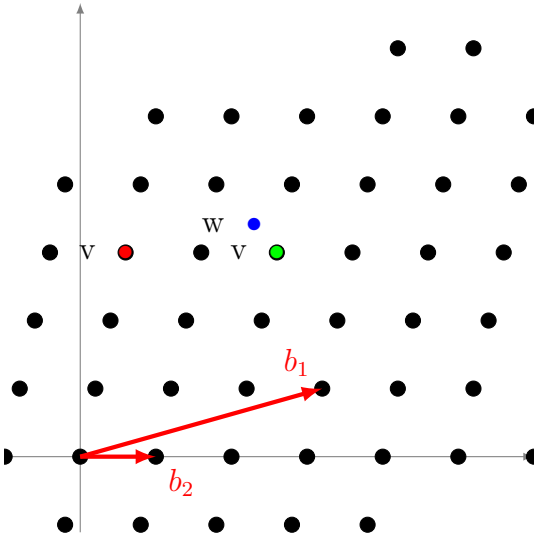


Note that in this case Babai's algorithm indeed returns the closest lattice vec-

tor because the basis is *good* (Hadamard's ratio: 0.976).

Example. Solving CVP using *bad* basis $\{(1,0),(3.2,0.9)\}$ (plotted in red) for point $(2.3,3.06)$ (expressed in standard basis) using Babai's algorithm.

Using Gaussian elimination, we first express the vector $\mathbf{w} = (2.3,3.06)$ in the basis B. We find that $\mathbf{w} = -8.58(1,0) + 3.4(3.2,0.9)$. Using Babai's algorithm, the nearest lattice point (expressed in basis B) is $(\lfloor 1.-8.58 \rceil, \lfloor 3.4 \rceil) = (-9,3)$. In standard basis, $\mathbf{v} = -9(1,0) + 3(3.2,0.9) = (0.6,2.7)$ (marked by red in the plot).



Note that in this case Babai's algorithm returns a poor solution which is far from the target vector because the basis is *bad* (Hadamard's ratio: 0.0.271).

5 GGH Cryptosystem

We shall now study the simplest and one of the earliest public key cryptosystems based on lattices, proposed by Goldreich, Goldwasser and Halevi in 1997. It is based on the approximate version of CVP.

We first choose a *good* and a *bad* basis of an n-dimensional lattice $\mathcal{L}$, say, V and W, respectively. We publish W as public key and keep V private. The key generation is formally describted below:

KGen(n)

1 : A *good* basis, $V \leftarrow_{\$} \mathbb{Z}^{n\times n}$, $\mathcal{H}(V) \approx 1$
2 : A *bad* basis, $W \leftarrow_{\$} \mathbb{Z}^{n\times n}$, $\mathcal{H}(W) << 1$
3 : Public key $\mathsf{pk} \leftarrow W$
4 : Secret key $\mathsf{sk} \leftarrow V$
5 : **return** $(\mathsf{pk}, \mathsf{sk})$

For encryption, we express the plaintext as an integer row vector, $\mathbf{m}$. Then, clearly $\mathbf{m}W \in \mathcal{L}$ is a lattice point. We then perturb this point a little by adding a small random vector $\mathbf{r}$ (such that, the absolute value of each of its components

$|r_i| \leq \delta$), the resulting vector will be the ciphertext. Thus, $\mathbf{c} = \mathbf{m}W + \mathbf{r}$, is the ciphertext. The encryption algorithm is formally described below:

$\mathsf{Enc}_\delta(\mathbf{m}, \mathsf{pk})$

1 : $\mathbf{r} \leftarrow_\$ [-\delta, \delta]^{1\times n}$

2 : $\mathbf{c} = \mathbf{m}W + \mathbf{r}$

3 : **return c**

For decryption, we use Babai's algorithm with a good basis V (which is the private key) to obtain the closest lattice vector to $\mathbf{c}$, which is $\mathbf{m}W$. We then multiply it by W^{-1} to obtain the plain text $\mathbf{m}$.

$\mathsf{Dec}(\mathbf{c}, \mathsf{sk})$

1 : $\mathbf{v} \leftarrow \texttt{SolveCVP}(V, \mathbf{c})$

2 : $\mathbf{m} \leftarrow \mathbf{v}W^{-1}$

3 : **return m**

Example. We will demonstrate GGH cryptosystem for 3-dimensions with a very simple example. (Note that, in order to achieve desirable security the dimension should be of the order of 10^2).

Let the secret key sk be the *good* basis V.

$$V = \begin{bmatrix} 100 & 0 & 0 \\ 10 & 90 & 20 \\ 0 & 15 & 95 \end{bmatrix}$$

This is a *good* basis because its Hadamard's Ratio $\mathcal{H}(V) = 0.924$ is sufficiently close to 1. Then we choose the public key pk to be the *bad* basis, W. We can generate this *bad* basis by multiplying the *good* basis with a unimodular matrix U, whose rows are long oblique vectors.

$$W = UV = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 100 & 0 & 0 \\ 10 & 90 & 20 \\ 0 & 15 & 95 \end{bmatrix} = \begin{bmatrix} 100 & 0 & 0 \\ 410 & 90 & 20 \\ 400 & 15 & 95 \end{bmatrix}$$

Suppose the plain text is $\mathbf{m} = (1, 2, 3)$, and we choose the small random random vector $\mathbf{r} = (-1, 2, 1)$. Then the cipher text is given by:

$$\mathbf{c} = \mathbf{m}W + \mathbf{r} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 100 & 0 & 0 \\ 410 & 90 & 20 \\ 400 & 15 & 95 \end{bmatrix} + \begin{bmatrix} -1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 2119 & 227 & 326 \end{bmatrix}$$

For decryption, we use the private key V, to solve CVP for $\mathbf{c}$ using Babai's algorithm. We first express the ciphertext vector in the *good* basis, we find that $(2119, 227, 326) = 20.9879(100, 0, 0) + 2.0212(10, 90, 20) + 3.0061(0, 15, 95)$. Thus the closest lattice vector (expressed) in *good* basis is $(21, 2, 3)$, or in standard basis:

$(2120, 225, 325)$. Post-multiplying this vector with W^{-1}, we recover the plain-text vector.

$$\begin{bmatrix}2120 & 225 & 325\end{bmatrix} \cdot \frac{1}{16500} \cdot \begin{bmatrix}165 & 0 & 0\\ -619 & 190 & -40\\ -597 & -30 & 180\end{bmatrix} = \begin{bmatrix}1 & 2 & 3\end{bmatrix}$$

This demonstration dealt with lattices of 3-dimensions. In practice, the number of dimensions are generally of the order of 100.

6 GGH Signature Scheme

We shall now describe a signature scheme based on approximate version of CVP, which was proposed by Goldreich, Goldwasser and Halevi in 1997.

As in the GGH encryption, the secret(signing) key sk is a *good* basis for a lattice $\mathcal{L}$, while the public(verification) key vk is a *bad* basis for the same lattice $\mathcal{L}$.

$\mathsf{KGen}(\mathrm{n})$

1 : A *good* basis, $V \leftarrow_{\$} \mathbb{Z}^{n\times n}$, $\mathcal{H}(V) \approx 1$
2 : A *bad* basis, $W \leftarrow_{\$} \mathbb{Z}^{n\times n}$, $\mathcal{H}(W) << 1$
3 : Verification key $\mathsf{vk} \leftarrow W$
4 : Signing key $\mathsf{sk} \leftarrow V$
5 : **return** $(\mathsf{vk}, \mathsf{sk})$

Let the message to be signed be a non-lattice vector $\mathbf{m} \in \mathbb{R}^n$. Then CVP for $\mathbf{m}$, in the *good* basis, using Babai's algorithm. The signature is the solution of the CVP expressed in the *bad* basis.

$\mathsf{Sig}(\mathbf{d}, \mathsf{sk})$

1 : $\mathbf{s} \leftarrow \mathtt{SolveCVP}(\mathsf{sk}, \mathbf{d})$
2 : **return** $\mathbf{s}$

Now, for verification, it just suffices to test if the signature vector is *sufficiently* close to the document vector. It is known that Babai's algorithm can almost always solve $\mathrm{CVP}_{\sqrt{n}}$ using a *good* basis, whereas, solving $\mathrm{CVP}_{\sqrt{n}}$ using a *bad* basis is difficult. Hence, we check if the distance between signature vector and the document vector is less than $\sqrt{n}\sigma(\mathcal{L})$, where, $\sigma(\mathcal{L})$ is the Gaussian expected shortest length in the lattice $\mathcal{L}$. Recall that $\sigma(\mathcal{L}) = \sqrt{\frac{n}{2\pi e}} \det \mathcal{L}^{1/n}$.

$\mathsf{Vf}(\mathbf{s}, \mathbf{d}, \mathsf{vk})$

1 : `if` $\|\mathbf{s} - \mathbf{d}\| \leq \sqrt{n}\sigma(\mathcal{L})$, `then`, **accept**

2 : `otherwise`, **reject**

Example. We demonstrate the GGH signature using the same example used for GGH encryption.

The secret (signing) key and the public (verification) key are the same as that in the previous example.

Let the document to be signed be the vector $\mathbf{d} = (368, 465, 593)$. To sign the document, we first solve CVP for $\mathbf{d}$ using the good basis V. Solving CVP for $\mathbf{m}$ using the basis V, we get the solution $\mathbf{s} = 3(100, 0, 0)+4(10, 90, 20)+5(0, 15, 95) = (340, 435, 555)$. This solution $\mathbf{s}$ is published as the signature.

To verify, we just need to need check if the distance between the document vector and the signature vector is less than $\sqrt{n}\sigma(\mathcal{L}) = \sqrt{3}\left(\frac{3}{4\pi}\right)^{1/3} \det \mathcal{L}^{1/3} = 100.77$, which is indeed the case (Note that in this case we have not used Stirling's approximation to find $\sigma(\mathcal{L})$ because the dimension is quite small).

Rejection Sampling

In the GGH signature scheme, the document-signature pair $(\mathbf{d}, \mathbf{s})$ reveals some information about the the signing key. Since, $\mathbf{s}$ is the solution of CVP for target vector $\mathbf{d}$, $\|\mathbf{d} - \mathbf{s}\|$ should be quite small. In fact,

$$\mathbf{d} - \mathbf{s} = \sum_{i=1}^{n} \epsilon_i(\mathbf{d}, \mathbf{s})\mathbf{v}_i$$

where $-\frac{1}{2} \leq \epsilon_i(\mathbf{d}, \mathbf{s}) \leq \frac{1}{2}$. An adversary who has transcript of several such document-signature pair, will also have a large number of points randomly scattered in zero-centered fundamental domain

$$\mathcal{F}' = \left\{ \sum_{i=1}^{n} a_i \mathbf{v}_i : -\frac{1}{2} \leq a_i \leq \frac{1}{2} \right\}$$

which can then be used to partially recover the *good* basis which was used as the signing key. In 2006, Nguyen and Regev proposed an algorithm which could recover signing key of an n-dimensional GGH signatures using a transcript containing roughly n^2 document-signature pair.

In 2012, Lyubashevsky proposed a lattice-based signature scheme which used rejection sampling to thwart attacks based on transcript analysis. In such schemes, Sig is a function of $\mathbf{d}$, sk and a random number r. Not all the signatures generated by Sig are published. Only those which have a certain property are published, others are rejected. In particular, this property could be their distribution. The published signature have a certain distribution, this ensures, that an adversary who has a transcript of a large number of document-signature pair can only find

a small subset of the zero-centered fundamental domain of the good basis. This works as follows: the document-signature pair $(\mathbf{d}, \mathbf{s})$ has a certain distribution, so $\mathbf{d}-\mathbf{s}$ is not distributed uniformly in the zero-centered fundamental domain, but is distributed according to some other other distribution (like discrete Gaussian)[3]. This distribution is achieved using rejection sampling. We describe one such signature scheme based on rejection sampling in the next section.

7 Lyubashevsky's Signature Scheme

In 2012, Lyubashevsky proposed a signature scheme based on SIS problem which used rejection sampling to generate signatures having discrete Gaussian distribution. Before we introduce the scheme, we first define the discrete Gaussian distribution.

Definition. The n-dimensional Gaussian distribution having standard deviation $\sigma = \frac{s}{\sqrt{2\pi}}$, is given by the probability density function, $p_s : \mathbb{R}^n \to \mathbb{R}$

$$p_s(\mathbf{x}) = \frac{1}{s^n} \exp\left(-\frac{\pi \|\mathbf{x}\|^2}{s^2}\right)$$

Definition. The discrete Gaussian distribution $D_{\mathcal{L},s}$ for a lattice $\mathcal{L}$ is the Gaussian distribution restricted to the lattice

$$D_{\mathcal{L},s}(\mathbf{x}) = \begin{cases} p_s(\mathbf{x}) & \mathbf{x} \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}$$

We are now in a position to formally describe the scheme.

KGen

1 : Signing key $\mathsf{sk} \leftarrow_\$ \{-d, ..., 0, ..., d\}^{m \times k}$

2 : Verification key $\mathsf{vk} \leftarrow (A \leftarrow_\$ \mathbb{Z}_q^{n \times m}, T = AS)$

3 : **return** $(\mathsf{vk}, \mathsf{sk})$

Sig$(\mathbf{d}, A, S)$

1 : $\mathbf{y} \leftarrow_\$ D_\sigma^m$

2 : $\mathbf{c} \leftarrow H(A\mathbf{y}, \mathbf{d})$

3 : $\mathbf{z} \leftarrow S\mathbf{c} + \mathbf{y}$

4 : $r \leftarrow_\$ [0, 1]$

5 : `if` $r < \frac{D_\sigma^m(\mathbf{z})}{M D_{S\mathbf{c},\sigma}^m}$ `return`$(\mathbf{z}, \mathbf{c})$

6 : `else goto 1`

$\mathsf{Vf}(\mathbf{d}, \mathbf{z}, \mathbf{c}, A, T)$

1 : `if` $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ `and` $\mathbf{c} = H(A\mathbf{z} - T\mathbf{c}, \mathbf{d})$, **accept**

2 : `otherwise`, **reject**

8 Bimodal Lattice Signature Scheme (BLISS)

In 2013, Ducas, Durmus, *et. al.* proposed a signature scheme called *BLISS*, based on Lyubashevsky's scheme. BLISS samples its signatures from a *bimodal* Gaussian distribution. Sampling from a bimodal Gaussian has the advantage of faster key generation (due to fewer rejections) and smaller key sizes(for the same security parameter). We did a brief complexity analysis of BLISS (integer version).

Complexity Analysis of BLISS signature. [Integer Version]

Key Generation Algorithm

(Page 38 of [4])

Input : n, m, q, α

Output : PublicKey : $A \in \mathbb{Z}_q^{n \times m}$; Secret Key $S \in \{-2^\alpha, -2^\alpha+1, \ldots\ldots, 2^\alpha\}^{m' \times n}$ such that $AS = qI_n$

The following algorithm efficiently generates the Key pairs:

KeyGen(n, m, q, α):
let $m' = m - n$
$A'_q \leftarrow \mathbb{Z}_q^{n \times m'}$
$S' \leftarrow \mathbb{Z}_q^{m' \times n}$, $S' \in \{-2^\alpha, -2^\alpha + 1, \ldots\ldots, 2^\alpha\}^{m' \times n}$
$A = (2A'_q | qI_n - 2A'_q S')\ mod\ 2q \in \mathbb{Z}_{2q}^{n \times m}$
$S = \begin{pmatrix} S' \\ I_n \end{pmatrix} \in \mathbb{Z}_{2q}^{m \times n}$
output (Public Key A and Secret key S)

We can observe that $AS = qI_n\ mod\ 2q$

Now we breifly analyze the time complexity of the key generation algorithm.

We ignore the explicit dependence of the time complexity over the modulus q as $q \ll 2^{32}$

	Time Complexity	Entropy Consumption
$A'_q \leftarrow Z_q^{n \times m'}$	$\mathcal{O}(nm')$	$nm'log(q)$
$S' \leftarrow \{-2^\alpha, -2^\alpha + 1, \ldots\ldots, 2^\alpha\}^{m' \times n}$	$\mathcal{O}(nm')$	$2\alpha nm'$
$A = (2A'_q \| qI_n - 2A'_q S')\ mod\ 2q$	$\mathcal{O}(n^2 m')$	0
Overall	$\mathcal{O}(n^2 m')$	

Signing Algorithm

input :$A \in Z_q^{n \times m}, S \in Z_q^{m \times n}$, μ

output : signature(z, c), message μ

Sign(A, S, μ):
$y \leftarrow {\mathcal{D}_\sigma}^m$
$c = H(Ay, \mu)$
$b \leftarrow \{0, 1\}$
$z = (-1)^b Sc + y$
$u \leftarrow [0, 1)$ withλ bits of precision
$\beta = \frac{1}{Mexp(-\frac{||Sc||}{2\sigma^2})cosh(\frac{<z,Sc>}{\sigma^2})}$
if $u < \beta$: output (z, c, μ)
else : repeat Sign(A, S, μ):

Complexity Analysis :

	Time	Memory	Entropy
$y \leftarrow {\mathcal{D}_\sigma}^m$	$\mathcal{O}(mlog(\sigma))$	$\mathcal{O}(m\lambda\tau\sigma)$	λm (Precision requirement)
$c = H(Ay, \mu)$	$\mathcal{O}(mn + \kappa)$	-	0
$b \leftarrow \{0, 1\}$	-	-	1bit
$z = (-1)^b Sc + y$	$\mathcal{O}(mn)$	$\mathcal{O}(m)$	0
Rejection Sampling	$\mathcal{O}(m)$	-	λ(Precision requirement)

Discrete gaussian sampling

The discrete gaussian ${\mathcal{D}_\sigma}^m$ can be sampled using various methods depending on the memory and time constraints.

The cummulative distribution table algorithm ([4]) is the most efficient technique , given we have sufficient memory.

The cummulative distribution table

In this method one precomputes the approximate cumulative distribution of the desired distribution upto λ bits of precision for a certain domain of z (say $z \in [c - \tau\sigma, c + \tau\sigma]$). While sampling, one generates $y \in [0, 1)$ uniformly at random, performs a binary search through the table to locate some $z \in Z$ such that$y \in [p_{z-1}, p_z)$ and outputs z.

Cumulative distribution table algorithm:

Compute_CD_Table$(c, \tau\sigma, \lambda)$:
$z = [c - \tau\sigma, c + \tau\sigma]$ (similar to arange$(c - \tau\sigma, c + \tau\sigma)$)
table = $\exp(-||z - c||^2/2\sigma^2)$;
output (table);

Sample_$\mathcal{D}_\sigma$(table):
PDF = table[:-1:]-table;
$u \leftarrow [0, 1)$; // with λ bits of precision
z = where(Binary_search$(u \approx$ PDF));
output z;

We believe that this method works not just with the discrete gaussian but also for any distribution whose p.d.f is monotonically decreasing.

For sampling $\mathcal{D}_\sigma$:
Precomputed bits : $\lambda\tau\sigma$
Time complexity during sampling : $log(\sigma\tau)$
Entropy : λ bits.
Overall time complexity : $M \times \mathcal{O}(mn + mlog(\sigma))$
The average time complexity of hash function H is $\mathcal{O}(m + k)$

Verification Algorithm

input : $(z, c, \mu,$ public parameters) output : "reject" or "accept"

Verify$(z, c, \mu,$ public parameters):
if $||z|| < B_2$: reject
if $||z||_\infty < q/4$: reject
if $c = H(Az - qc, \mu)$: accept
else reject

	Time
$\|\|z\\|\| < B_2$	$O(m)$
$\|\|z\|\|_\infty < q/4$	$O(m)$
$H((Az - qc) \ mod \ 2q, \mu)$	$O(m\kappa)$

Overall complexity : $O(m\kappa)$
$B_2 = \eta\sigma\sqrt{m}$
Where η is chosen according to Lemma 4.4 in [3].

References

[1] Peikert. *A Decade of Lattice cryptography.* Cryptology ePrint Archive: Report 2015/939. 2016.

[2] Hoffstein, Pipher, Silverman. *Introduction to Mathematical Cryptography.* Springer. 2014.

[3] Lyubashevsky. *Lattice Signatures Without Trapdoors.* Cryptology ePrint Archive: Report 2011/537. 2012.

[4] Ducas, Durmus, Lepoint, Lyubashevsky. *Lattice signatures and Bimodal Gaussians.* Cryptology ePrint Archive: Report 2013/383. 2013.

[5] Fleming. *Functions of Several Variables.* Springer. 1977.

[6] Ajtai. *The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions.* Proceedings of the ACM symposium on Theory of computing. 1998.

[7] van Emde Boas. *Another NP-complete problem and the complexity of computing short vectors in a lattice.* Technical Report 8104. University of Amsterdam. 1981.