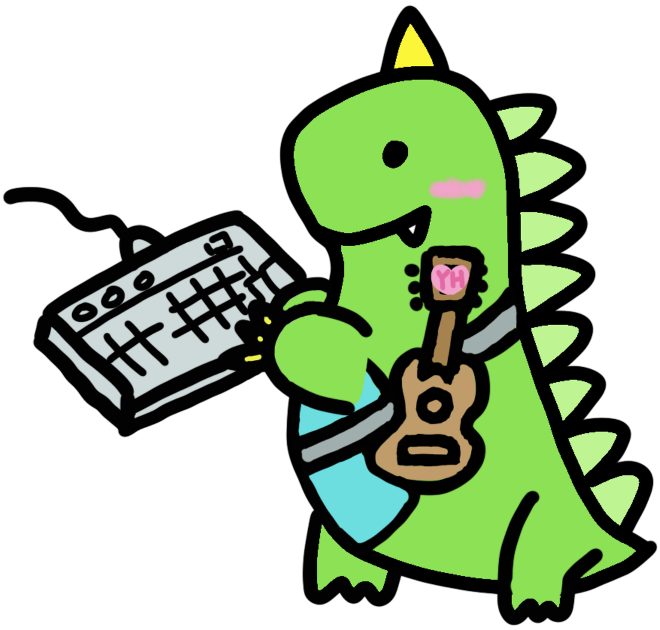
Theorem You Must Know

Some theorem proofs related to Cryptography or computational number theory.

@fffmath



Contents

1 Introduction **5**
1.1 How to Use this Book 5

2 Lattice Based Cryptography **7**
2.1 Basic of lattice 7
2.1.1 The two definitions of lattice are equivalent 7
2.1.2 Complexity of LLL-algorithm 8

3 Zero-Knowledge Proof **11**

4 Multi-Party Computation **13**

5 Quantum Complexity **15**

About the Author **17**

Bibliography **19**

1 Introduction

The original TeX was created by the famous computer scientist Donald Knuth (Knuth and Bibby, 1984), and added to by Leslie Lamport to make LaTeX (Lamport, 1985).

1.1 How to Use this Book

These are the main ways you can use this material:

- Lattice
- Zero-Knowledge Proof
- You can use it as a template. All of the source files used to make this book are freely available in GitHub at `https://github.com/dwiddows/ebookbook` and Overleaf. The source files are laid out in a way that should make it easy to clone the project and adapt it for your own book.

2 Lattice Based Cryptography

2.1 Basic of lattice

2.1.1 The two definitions of lattice are equivalent

Definition 1 (Lattice)**.** *Given n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice generated by them is defined as*

$$\mathbb{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \;\middle|\; x_i \in \mathbb{Z} \right\}$$

Definition 2. *A lattice $\mathbb{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.*

Theorem 1. *The two definitions of lattice are equivalent.*

Proof. We will first show that Definition 1 $\Rightarrow$ Definition 2.

Assume $\mathbb{L}$ is a lattice defined as the set of all integer combinations of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ which are linearly independent (Definition 1). Then, clearly L is an additive subgroup of $\mathbb{R}^n$. In addition, $\forall \mathbf{x}, \mathbf{y} \in L$, $\mathbf{x} - \mathbf{y} \in L$. Therefore, from the lower bound on a shortest lattice vector,

$$\|\mathbf{x} - \mathbf{y}\| \geq \lambda_1(\mathbb{L}) \geq \min_{i=1,\ldots,n} \|\tilde{\mathbf{b}}_i\|.$$

In other words, the length of any lattice vector must be greater than the length of a shortest lattice vector. Therefore, we can let $\varepsilon = \lambda_1$. So, both properties of Definition 2 are satisfied (L is a discrete additive subgroup of $\mathbb{R}^n$).

We show that Definition 2 $\Rightarrow$ Definition 1. Given a discrete additive subgroup L of $\mathbb{R}^n$, we construct a set of basis using the algorithm below.

We will use the following definition of a closed parallelepiped:

Given n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, their closed fundamental parallelepiped is defined as

$$\overline{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \;\middle|\; x_i \in \mathbb{R}, 0 \leq x_i \leq 1 \right\}$$

Pick $\mathbf{y} \in \mathbb{L}$ such that there is no lattice vector between the zero vector and $\mathbf{y}$. Let $\mathbf{b}_1 = \mathbf{y}$. Iterate for all i, $1 \leq i < n$: Assume we have already chosen $\mathbf{b}_1, \ldots, \mathbf{b}_i$. Choose $\mathbf{y}$ not in the span of $\mathbf{b}_1, \ldots, \mathbf{b}_i$. Consider a $\overline{P}(\mathbf{b}_1, \ldots, \mathbf{b}_i, \mathbf{y})$ (See Figure-1 for an example). Now, $\overline{P}$ contains at least one lattice point (namely $\mathbf{y}$) and it contains finitely many lattice points. Now, choose a vector $\mathbf{z} \in \overline{P}(\mathbf{b}_1, \ldots, \mathbf{b}_i, \mathbf{y}) \setminus \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$ such that $\mathrm{dist}(\mathbf{z}, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_i))$ is the smallest.

Note that we can do this since we have only finitely many points to choose from. Let $\mathbf{b}_{i+1} = \mathbf{z}$.

We will now show that the above algorithm returns a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ for the lattice. Clearly, all $\mathbf{b}_i$s are in $\mathbb{R}^m$ and they are linearly independent by the algorithm that we used. We are left to show that $L \subseteq \{\sum x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$.

Let $\mathbf{z} = \sum z_i \mathbf{b}_i$ be an arbitrary lattice vector (where $z_i \in \mathbb{R}$). Let $\mathbf{z}_0 = \sum b_z^i \mathbf{c}_i$ be an element of L. Then, $\mathbf{z} - \mathbf{z}_0 = \sum (z_i - b_z^i c_i)\mathbf{b}_i$ is in L. We will show that all coefficients z_i must be integers. Express

$$\mathbf{z} - \mathbf{z}_0 = (z_n - \lfloor z_n \rfloor)\mathbf{b}_n + \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}) = (z_n - \lfloor z_n \rfloor)\mathbf{b}_n + \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}).$$

In other words, vector $\mathbf{z} - \mathbf{z}_0$ is in the span of $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$ plus a multiple of $\tilde{\mathbf{b}}_n$ with coefficients $0 \leq b_z^n c_n < 1$.

Now,

$$\mathrm{dist}(\mathbf{z} - \mathbf{z}_0, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})) = (z_n - \lfloor b_n \rfloor)\|\tilde{\mathbf{b}}_n\|.$$

This follows because the distance is defined as the orthogonal component of $\mathbf{z} - \mathbf{z}_0$ to the span $\mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$, which is precisely $(z_n - b_z^n c_n)\|\tilde{\mathbf{b}}_n\|$. Similarly,

$$\mathrm{dist}(\mathbf{b}_n, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})) = \|\tilde{\mathbf{b}}_n\|.$$

In addition, since $0 \leq (z_n - b_z^n c_n) < 1$,

$$\mathrm{dist}(\mathbf{z} - \mathbf{z}_0, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})) < \mathrm{dist}(\mathbf{b}_n, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})).$$

But since $\mathbf{b}_n$ was chosen as the closest vector to $\mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$, $\mathbf{z} - \mathbf{z}_0$ must be linearly dependent on $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$. Therefore, $z_n - \lfloor b_n \rfloor = 0$ and so $z_n \in \mathbb{Z}$.

By recursively repeating the above argument for $\mathbf{z} = \mathbf{z} - \mathbf{z}_i \in \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$ for all $1 < i \leq n$, we obtain that all coefficients z_j for $1 \leq j \leq n$ must be integers.

□

2.1.2 Complexity of LLL-algorithm

Theorem 2. *Given an integer n-dimensional lattice basis with vectors of Euclidean norm less than B in an n-dimensional space, the LLL algorithm outputs a reduced basis in $O(n^4 \log B \cdot M(n \log B))$ bit operations, where $M(k)$ denotes the time required to multiply k-bit integers.*

Proof. Our analysis consists of two steps. First, we bound the number of iterations. Second, we bound the running time of a single iteration.

We show that the overall running time of the algorithm is polynomial in the input size. A rough lower bound on the latter is given by $N := \max(n, \log(\max_i kb_i))$ (because each of the n vectors requires at least one bit to represent, and a vector of norm r requires at least $\log r$ bits to represent).

In the following, we show that the running time of the algorithm is polynomial in M. Moreover, the LLL algorithm outputs a reduced basis in $O(n^4 \log B \cdot M(n \log B))$ bit operations, where $M(k)$ denotes the time required to multiply k-bit integers.

Algorithm 1: δ-LLL Algorithm

Data: Lattice basis $b_1, \ldots, b_n \in \mathbb{Z}^n$
Result: δ-LLL-reduced basis for $L(B)$

Compute $\tilde{b}_1, \ldots, \tilde{b}_n$;
for $i = 2$ *to* n **do**
  for $j = i - 1$ *to* 1 **do**
    $c_{i,j} \leftarrow \frac{d \cdot \langle b_i, \tilde{b}_j \rangle}{\|\tilde{b}_j\|^2}$;
    $b_i \leftarrow b_i - c_{i,j} \cdot b_j$;
  end
  Compute $\tilde{b}_i$;
  if $\exists i$ *s.t.* $\delta k \tilde{b}_i^2 > k \mu_{i+1,i} \tilde{b}_i + \tilde{b}_{i+1}^2$ **then**
    Swap $b_i \leftrightarrow b_{i+1}$;
    goto Start;
  end
end
Output $b_1, \ldots, b_n$

If the LLL algorithm terminates, it is clear that the output basis is LLL-reduced. What is less clear a priori is why this algorithm has a polynomial-time complexity. A standard argument shows that each swap decreases the quantity $\Delta = \prod_{i=1}^{n} \|b_i^*\|^2 (n - i + 1)$ by at least a factor $\delta < 1$. On the other hand, we have that $\Delta \geq 1$ because the b_i's are integer vectors and Δ can be viewed as a product of squared volumes of lattices spanned by some subsets of the b_i's. This proves that there can be no more than $O(n^2 \log B)$ swaps, and therefore loop iterations, where B is an upper bound on the norms of the input basis vectors.

It remains to estimate the cost of each loop iteration. This cost turns out to be dominated by $O(n^2)$ arithmetic operations on the basis matrix and GSO coefficients $\mu_{i,j}$ and $r_{i,i}$, which are rational numbers of bit-length $O(n \log B)$. Thus, the overall complexity of the LLL algorithm described can be bounded by $O(n^4 \log B \cdot M(n \log B))$. $\square$

3 Zero-Knowledge Proof

4 Multi-Party Computation

5 Quantum Complexity

About the Author

Hello! I'm fffmath, a Master of Mathematics student at the Chinese Academy of Sciences (CAS), with research interests in Cryptography. My personal site is https://www.fffamth.com.

My research interests include:

- Cryptography: Lattice based cryptography, Provable Security
- Theoretical Computer Science: Complexity of hard problem in lattice or other algebraic structure

And you can find my CV in https://www.fffmath.com/file/mycv.pdf.

In addition to my studies, I enjoy playing guitar and exploring new hobbies. I'm also fluent in English and my native language is Chinese.

Bibliography

Knuth, D. E. and Bibby, D. (1984). *The TeXbook*, volume 15. Addison-Wesley Reading.

Lamport, L. (1985). *LaTeX: A Document Preparation System*, volume 410. Addison-Wesley Professional; 2nd edition (June 30, 1994).