Idk what my thesis title is

Bachelor's Project Mathematics

March 2023

Student: K. J. Pudowski

First supervisor: <add titles> Killicer

Second assessor: <add titles> Seri

Contents

The plan:

- introduction
  - why should we care about encryption schemes
  - introduce the classical ones
  - introduction of a problem caused by Shor
  - solution by lattices
- preliminaries
  - lattices
  - alg nt
  - complexity theory and reductions
  - shor's alg

- she & fhe
- explain how lbc fits into all of those categories
- explain some of the specific ones (like lwe, ring-lwe) and why we think they are secure (reductions)
- show couple of implementations
  - present the fhe from ring-lwe (r-lwe), lwe
  - compare all of them somehow
- show your own C/OCaml implementations (if enough time)

1 Introduction

Introduction to lattice-based cryptography: Discussing the basics of lattice-based cryptography, including the definition of a lattice, the LWE problem, and the use of lattices in cryptography.

motivate the reader why they should care.

2 Background

number rings (number fields), ideals, ring of integers, geometry then discriminant, why do they use cyclotomic. can we use any integer ring or do we need cyclotomic ones.

2.1 Notation

Most of the notation is "standard" in the field of number theory and cryptography. Nonetheless, to avoid any misunderstandings and simplify some statements, we will present the notation used throughout the text. Anything that is not mentioned here shall be defined "on the go" with definitions and such.

Any scalars $a, t, \beta, \dots$ are represented by non-bold, latin or greek letters.
Bold symbols $\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{s}, \dots$ will denote vectors.
Algorithms will be represented by <name of this font idk> font names such as Encrypt or *Evaluate*.
Traditionally, the symbols $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ shall represent the sets of integers, rationals, reals and complex numbers, respectivelly.

2.2 Lattices

Basic Definitions

We define a *lattice* as a discrete additive subgroup of $\mathbb{R}^n$. Once we fix a basis $\boldsymbol{B} = (b_1, \ldots, b_n) \in \mathbb{R}^n$ we can then describe the lattice as

$$\Lambda = \mathcal{L}(\boldsymbol{B}) = \Big\{ \sum_i z_i \boldsymbol{b}_i : z \in \mathbb{Z}^n \Big\}.$$

There are many bases for a lattice (actually infinitely many as can be proven using a little diagonalization argument [Krzys: can it tho?]), some "better" than others. This will be the foundation for some of the problems like SVP or CVP.

Example 2.1. The simplest example of a lattice is the $\mathbb{Z}^n$ itself. Taking the standard basis $\boldsymbol{B}_1 = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ we obtain

$$\mathcal{L}(\boldsymbol{B}_1) = \Big\{ \sum_i z_i \boldsymbol{e}_i : z \in \mathbb{Z}^n \Big\} = \mathbb{Z}^n.$$

More generally, Λ is a lattice of rank m in $\mathbb{R}^n$ if it is a rank m free abelian group. Recall that we call a group *free abelian group* of rank m if it can be written as $\Lambda = \mathbb{Z}\beta_1 \oplus \cdots \oplus \mathbb{Z}\beta_m$ with $\beta_1, \ldots, \beta_m$ linearly independent over $\mathbb{R}$ where $\oplus$ represents the direct sum. In this paper we will only consider lattices of full rank.

Remark 2.2. We can also view the vectors $\boldsymbol{b}_i$ as the rows of the matrix $B \in \mathbb{R}^n \times \mathbb{R}^n$ in which case, our definition becomes:

$$\Lambda = \mathcal{L}(B) = \{\boldsymbol{z}\boldsymbol{B} : \boldsymbol{z} \in \mathbb{Z}^n\}.$$

Reciprocally, any matrix $\boldsymbol{B} \in GL_n(\mathbb{R})$ spans a lattice: the set of all integer linear combinations of its rows.

Example 2.3.

1. $\mathcal{L} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ In this example $\beta_1 = \binom{1}{0}$ and $\beta_2 = \binom{0}{1}$
2. $\mathcal{L} = \{(z_1, z_2) : z_1 + z_2$ is even [Krzys: would odd work here?]$\}$
3. $\mathcal{L} = \begin{pmatrix} 13 & 21 \\ 21 & 34 \end{pmatrix}$

   [Krzys: this is useful for (ring)-lwe, not sure if i should include this at all]

Definition 2.4 (Dual)**.** For a lattice $\Lambda \subset \mathbb{R}^n$ its $\mathbb{Z}$-*dual* is

$$\Lambda^\vee = \{y \in \mathbb{R}^n : y \cdot \Lambda \subset \mathbb{Z}\}.$$

Here, the $\cdot$ means the usual dot product.

We simply require that the elements of the dual are precisely those vectors that yield an integer when "multiplied" with an element of our lattice. Note that this is different from our standard definition of a dual. Namely, it is not the orthogonal compliment of our starting space, i.e. not all of the elements of the dual have 0 dot product against the vectors of the lattice.

Example 2.5. Take $\mathcal{L} = \mathbb{Z}\binom{1}{2} + \mathbb{Z}\binom{0}{1}$ To calculate the dual of $\mathcal{L}$ we need our $y = \binom{a}{b}$ elements to satisfy $a \in \mathbb{Z}$ and $2a + b \in \mathbb{Z}$ which is equivalent to asking $a \in (1/2)\mathbb{Z}$ and so $\mathcal{L}^\vee = \binom{1/2}{0}\mathbb{Z} + \binom{0}{1}\mathbb{Z}$

Note that $\mathcal{L}^\vee$ is itself a lattice of the same dimension. [Krzys: end of dual section]

Fundamental Domain

Definition 2.6 (Fundamental Domain)**.** Let $\mathcal{L}$ be a lattice of dimension n and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be a basis for $\mathcal{L}$. The *fundamental domain* (or *fundamental parallelepiped*) for $\mathcal{L}$ corresponding to this basis is the set

$$\mathcal{F}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \{t_1\boldsymbol{b}_1 + \cdots + t_n\boldsymbol{b}_n : 0 \leq t_i < 1\}.$$

We define the *volume* of $\mathcal{F}(\boldsymbol{B})$ as the volume of the corresponding parallelepiped in $\mathbb{R}^n$. The *volume* - closely connected to the determinant - plays a very important role in our study which will become evident in later chapters. One of the advantages, of defining the fundamental domain, is that we can formalize the area (or the determinant) or any given lattice. Recall that a lattice is just a countable collection of points and therefore has no volume by itself. This, however, is resolved by introducing the following.

Definition 2.7. Let $\mathcal{L}$ be a lattice of dimension n and let $\mathcal{F}(\boldsymbol{B})$ be a fundamental domain for $\mathcal{L}$ over some basis $\boldsymbol{B}$. We define the *determinant* of that lattice as

$$\det(\mathcal{L}) = \mathrm{Vol}(\mathcal{F}(\boldsymbol{B})) = |\det(\boldsymbol{B})|$$

The next two propositions are *de facto* foundation for lattice based cryptography. The first one states that the $\det(\mathcal{L})$ does not depend on the choice of the basis for that lattice. The second, that our whole ambient space $\mathbb{R}^n$ can be described using only vectors from the lattice and the fundamental domain. We will only give an outline of the proofs for the sake of keeping this section compact. Full proofs, however, can be found in **book**, chapter 6.4.

Proposition 2.8. *The* $\det(\mathcal{L})$ *of an* n*-dimensional lattice is invariant under the choice of the basis.*

Outline of the proof. Let $\boldsymbol{B}_1, \boldsymbol{B}_2$ be two bases for a lattice $\mathcal{L}$. The crucial part of the proof is to note that any two bases are related by some unimodular matrix U (i.e. a matrix with the determinant of ± 1) s.t. $\boldsymbol{B}_1 = U\boldsymbol{B}_2$. It now easily follows to compute $|\det(\boldsymbol{B}_1)| = \det(\mathcal{L}) = |\det(U \cdot \boldsymbol{B}_2)| = |\det(U)| \cdot |\det(\boldsymbol{B}_2)| = |\det(\boldsymbol{B}_2)|$ □

From now on we will write $\mathcal{F}$ to denote the fundamental domain of the lattice without specifying the basis.

Proposition 2.9. *Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice of dimension n and let $\mathcal{F}$ be a fundamental domain for $\mathcal{L}$. Then every vector $\boldsymbol{v} \in \mathbb{R}^n$ can be written in the form*

$$\boldsymbol{v} = \boldsymbol{f} + \boldsymbol{t}$$

for $\boldsymbol{f} \in \mathcal{F}$ and $\boldsymbol{t} \in \mathcal{L}$ both unique and associated to the original $\boldsymbol{v}$.

Equivalently, the space $\mathbb{R}^n$ is spanned exactly (without overlap) by shifting the fundamental domain by the vectors from our lattice.

$$\mathbb{R}^n = \bigcup_{\boldsymbol{t} \in \mathcal{L}} \{\boldsymbol{f} : \boldsymbol{f} \in \mathcal{F}\}$$

Remark 2.10. Sometimes the *fundamental domain* is refered to as a parallelepiped or parallelotope and denoted by caligraphic $\mathcal{P}$. If we take a matrix $\boldsymbol{B}$ to represent our lattice $\mathcal{L}$, then $\mathcal{P}_{1/2}(\boldsymbol{B}) = \{\boldsymbol{xB}, \boldsymbol{x} \in [-1/2, 1/2]^n\}$ can also represent the (shifted by a half) fundamental domain of $\mathcal{L}$ (like for example in **gentry**).

We will now present two results that give us an upper bound on the length of the shortest vector in a lattice. This will later on be useful to determine the security of our schemes. These theorems are due to Hermite (1822 - 1901) and Minkowski (1864 - 1909).

Theorem 2.11 (Hermite's Theorem)**.** *Every lattice $\mathcal{L}$ of dimension n contains a nonzero vector $v \in \mathcal{L}$ satisfying*

$$\|v\| \leq \sqrt{n} \det(\mathcal{L})^{\frac{1}{n}}.$$

Remark 2.12.

2.3 Algebraic Number Theory

Algebraic number theory is the study of *number fields*, *rings of integers* and *finite fields*. In this section we will provide all the necessary background needed to understand and verify the results presented in the cryptographic schemes later in the text. Most results will be stated without proof however all of them can be found in the book **Number Fields** by **Daniel A. Marcus algebra** after which this sections is modelled.

Number Fields

In general, a *number field* is defined as a subfield of $\overline{\mathbb{Q}}$ having finite degree (the dimension as a vector space) over the rationals $\mathbb{Q}$. Throughout this section, we fix the imaginary numbers $\overline{\mathbb{Q}} = \mathbb{C}$ as an algebraic closure of the rationals.

[Pinar: Degree of a field over $\mathbb{Q}$. Define the monic minimal polynomial.]

Definition 2.13 (Algebraic integer)**.** An element $\alpha \in \mathbb{C}$ is an *algebraic integer* if it is a root of some monic polynomial with coefficients in $\mathbb{Z}$.

In fact, the set of algebraic integers forms a ring (under the usual addition and multiplication operations in K). To see this, we make use of the following lemma [Krzys: this seems like unnecessarily addition, can't this be proven more simply?].

Lemma 2.14. *Any $\alpha \in \mathbb{C}$ is an algebraic integer if and only if, the additive group of the ring $\mathbb{Z}[\alpha]$ is finitely generated.*

Proof. ($\Rightarrow$): If α is a root of a monic polynomial over $\mathbb{Z}$ of degree n, then the additive group $\mathbb{Z}[\alpha]$ is generated by $1, \alpha, \cdots, \alpha^{n-1}$.
($\Leftarrow$): [Krzys: TODO] □

Definition 2.15 (Ring of Integers)**.** We define the *ring of integers* (sometimes also called *number ring*) $\mathcal{O}_K$ of a number field K as the largest ring in the intersection:

$$\mathcal{O}_K = K \cap \overline{\mathbb{Z}} = \{x \in K : \ x \text{ is an algebraic integer}\}.$$

The *degree* of the number field is defined as the degree of the

Example 2.16. The field $K = \mathbb{Q}$ is a number field of degree 1.

Theorem 2.17. *In general, for a quadratic field $K = \mathbb{Q}(\sqrt{d})$ with square-free $d \in \mathbb{Z}$, $\mathcal{O}_K$ is $\mathbb{Z}[\sqrt{d}]$ or $\mathbb{Z}[(1+\sqrt{d})/2]$ depending on $d \mod 4$.*

Example 2.18. Its ring of integers $\mathcal{O}_K$ are the ordinary integers $\mathbb{Z}$. The field

$$K = \mathbb{Q}(\sqrt{-1}) = \{a + b\sqrt{-1} \mid a, b \in \mathbb{Q}\}$$

has degree 2 since $x^2 + 1$ is the minimal polynomial of $\sqrt{-1}$ over $\mathbb{Q}$ and with $\mathcal{O}_K = \mathbb{Z}[\sqrt{-1}]$.

for $K = \mathbb{Q}(\sqrt{5})$ the ring of integers is $\mathcal{O}_K = \mathbb{Z}[(1+\sqrt{5})/2]$.

Cyclotomic fields

Definition 2.19 (Roots of unity)**.** Given a field K and a positive integer n, an element $\zeta \in K$ is called *primitive n-th root of unity* if ζ has order n in the multiplicative group $K^\times$. (In other words, $\zeta^n = 1$ and $\zeta^m \neq 1$ for $1 \leq m < n$).

The minimal polynomial Φ_n of ζ over $\mathbb{Q}$ is called the n-th cyclotomic polynomial.

Embeddings in $\mathbb{C}$

Let $K = \mathbb{Q}(\alpha)$ be a number field of degree n for some α. Then there are exactly n canonical embeddings (injective ring homomorphisms) of K in $\mathbb{C}$. These are easily described by observing that α can be sent to any one of its n conjugates over $\mathbb{Q}$. Each conjugate β determines a unique embedding ($\sigma_i : K \to \mathbb{C}$ and every embedding must arise in this way since α must be sent to one of its conjugates).

Example 2.20. The quadratic field $\mathbb{Q}[\sqrt{d}]$, d squarefree, has two embeddings in $\mathbb{C}$: The identity mapping, and also the one which sends $a + b\sqrt{d}$ to $a - b\sqrt{d}$ (a, $b \in \mathbb{Q}$), since $\sqrt{d}$ and $-\sqrt{d}$ are the two conjugates of $\sqrt{d}$. The n-th cyclotomic field has $\varphi(n)$ embeddings in $\mathbb{C}$, the $\varphi(n)$ automorphisms where $\sigma_i(\zeta) = \zeta^i$.

[Pinar: what do we want to consider as a cyclotomic poly? is $n = 2^a$? is n a prime? and why?]

[Pinar: Maximal orders (ring of integers) are dedekind domains, embedding of $\mathbb{Q}(\alpha)$ to $\mathbb{C}$ hence embedding of the ideals. Properties of ideals in dedekind domains, operations, unique factorization and so on. all the necessary info.]

2.4 Complexity Theory and hard problems

[Pinar: will wait] In this section we will briefly introduce what it means for a problem to be considered *hard* and provide couple of examples

The best know examples

2.4.1 Shor's Algorithm

I'm not sure if that is supposed to be in a section about preliminaries but I also don't want to include it in the introduction coz its a bit long

3 Homomorphic Encryption

Fully Homomorphic Encryption (FHE) has been referred as the "holy grail" of modern cryptography as it was one of the most sought goals for the past couple of decades. First formally introduced by Rivest, Adleman and Dertouzos in **primal** (at the time called "privacy homomorphism"), shortly after the discovery of public key cryptography, it has been an open and elusive problem. Only "recently", in 2009, Craig Gentry proposed first FHE in his PhD thesis **gentry_phd**. Quote from **impl_gentry**:" Gentry later proved [5] that with an appropriate key-generation procedure, the security of that scheme can be (quantumly) reduced to the worst-case hardness of some lattice problems in ideal lattices."

Simply stated, in homomorphic encryption we want our data to be secure but we also want to perform calculations on it. This is useful when you need a third party (e.g. someone with more computational power) to perform operations on your data while still retaining privacy. Alice can store her data somewhere on external server (the cloud) and ask to perform computations on it. For example query searches without the engine knowing what is actually being searched for.

Remark 3.1. In general, an encryption scheme $\mathcal{E}$ is a tuple of $\mathsf{KeyGen}_{\mathcal{E}}$, $Encrypt_{\mathcal{E}}$ and $Decrypt_{\mathcal{E}}$ (representing the key-generation, encryption and decryption respectively), all of which we require to be *efficient* - i.e. run in time poly(λ), polynomial in the

security parameter λ that represents the bit-length of the keys (see for example **katz** or **book** for more details). A homomorphic encryption scheme has a fourth algorithm - $\mathsf{Evaluate}_{\mathcal{E}}$ which we associate with some set of *permitted functions*. Adopting the notation from **easy_fhe** we denote by $\mathcal{F}_{\mathcal{E}}$, the set of such functions.

Therefore we would like our encryption scheme to satisfy the following. Say the ciphertexts c_i's decrypt to messages m_i's. Then we want

$$\mathsf{Decrypt}_{\mathcal{E}}(c_1 + c_2) = m_1 + m_2, \qquad \mathsf{Decrypt}_{\mathcal{E}}(c_1 * c_2) = m_1 * m_2$$

Equivalently, we want $\mathsf{Decrypt}$ to be a homomorphism with respect to both addition and multiplication. Fully Homomorphic Encryption (in big simplification) means that whenever f is a composition of (few) additions and multiplications, then $\mathsf{Decrypt}_{\mathcal{E}}(f(c_1, \ldots, c_n)) = f(m_1, \ldots, m_n)$.

3.1 Somewhat Homomorphic Encryption

Before we introduce the solution presented by Gentry, we will start with something slightly simpler, introduced in **int_scheme** by van Dijk et al. Their scheme works over the integers rather than lattices but relies on a similar assumption. Namely, that finding the greatest common divisor of many "noisy" multiples of a number is computationally difficult. We will come back to this problem later. To keep the exposition compact, we will avoid specifying most parameter choices.

We begin with the **symmetric** key scheme. We take our message to be a bit $m \in \{0, 1\}$. The private key is an odd integer chosen from some interval $p \in [2^{\eta-1}, 2^{\eta})$. To encrypt our message m, we choose integers q and r at random (from some other interval and s.t. the magnitude of $2r$ is smaller than $p/2$). We obtain the ciphertext c by computing:

$$c = pq + 2r + m.$$

If we now want to decrypt our message, simply compute $(c \mod p) \mod 2$.
Let's say we have two messages c_1 and c_2. Then we can compute:

$$c_1 + c_2 = m_1 + m_2 + 2(r_1 + r_2) + p(q_1 + q_2),$$

$$c_1 * c_2 = m_1 * m_2 + 2(m_1 r_2 + m_2 r_1 + 2 r_1 r_2) + p(m_1 q_2 + m_2 q_1 + 2(r_1 q_2 + r_2 q_1) + p q_1 q_2)$$

where we can see that the noise grows with each operation and the message becomes impossible to decrypt after we do too many of them. If we can assure that $2(m_1 r_2 + m_2 r_1 + 2 r_1 r_2)$ is small enough - i.e. smaller than $p/2$ - we can assure that $\mathsf{Decrypt}(c_1 * c_2)$ evaluates correctly to the starting $m_1 * m_2$. Notice that $\mathsf{Decrypt}$ removes all the noise. This will be useful later for bootstrapping.

This simple encryption scheme is thus somewhat homomorphic as per definition by Gentry in **gentry_phd** – namely, it can be used to evaluate low-degree polynomials over encrypted data. Further on in §6 of **int_scheme**, van Dijk et al. use the techniques (called bootstrapping and squashing) to lift it to a Fully Homomorphic Scheme.

The **public** key scheme is build very similarly. The private key p stays the same. For the public key, sample $x_i = pq_i + 2r_i$ for $i = 0, 1, \ldots, t$ where the q_i and r_i stay as before. The x_i may be viewed as encryption of 0 under the symmetric key scheme. The x_i are now taken s.t. x_0 is the largest, odd and $x_0 \mod p$ is even. To now encrypt a message $m \in \{0, 1\}$, chose a random subset $S \subseteq \{1, 2, \ldots, t\}$ and a random integer r, and output

$$c = (m + 2r + 2\sum_{i \in S} x_i) \mod x_0.$$

To decrypt, we again output $m = (c \mod p) \mod 2$

The security of this preliminary SH scheme relies on the *Approximate GCD Problem*. In the simplest case, Euclid has shown us, that given two integers c_1 and c_2, it is easy to compute their gcd. However, suppose now that $c_1 = p \cdot q_1 + r_1$ and $c_2 = p \cdot q_2 + r_2$ are "near" multiples of p, where r_1 and r_2 is some small noise sampled at random. This turns out to be much more difficult. In fact, if we pick our values appropriately (see **easy_fhe** §3.4 and **int_scheme** §3 for details) we do not know any efficient (running in polynomial time) algorithm even if we are given arbitrarily many samples $c_i = r_i + p \cdot q_i$.

3.2 Fully Homomorphic Encryption

We will now present the main idea introduced in Gentry's PhD thesis **gentry_phd**. Namely, the initial "bootstrapping" result, our SHE from previous section and a technique to "squash the decryption circuit" to allow bootstrapping. At the end, we will finally be left with *Fully Homomorphic Encryption* scheme. As a basis, we will be using the SHE scheme from previous section.

Boostrapping

We are faced with a problem. Because our method relies on some error being added to the message, it builds up after we perform operations on our data. The scheme $\mathcal{E}$ can handle functions in a limited set $\mathcal{F}_\mathcal{E}$ until the noise becomes too large.

The basic idea behind bootstrapping is to use a homomorphic decryption circuit to "refresh" the ciphertext, so that it can be used for further homomorphic operations. The process involves encrypting the decryption circuit itself, evaluating it on the

ciphertext, and then re-encrypting the result to obtain a new ciphertext that can be used for further homomorphic operations.

To simplify our construction assume that our scheme $\mathcal{E}$ is "circula

4 Lattice based cryptography

Gentry's work was a true breakthrough. It not only presented the first, fully homomorphic encryption scheme, but also gave researchers a very powerful tool the *bootstrapping* is. From now on, all we need to construct another FHE scheme, is some suitable (one requirement would be to use a scheme based on ring rather than a group) SHE method, apply appropriate "squashing" to obtain the bootstrapping and we are done. In the following years this is exactly what happened in the academia and the industry.

This section will mostly serve as a survey of the main developments towards more efficient fully homomorphic encryption using lattices and their security based on computational hardness of the underlying problems. We adopt chronological narrative of the sections, starting with the oldest (but still relevant) GGH algorithm, progressing through works on (ring-)LWE and eventually arriving at the work of Gentry **gentry_phd** on ideal lattices and FHE.

4.1 The GGH public key cryptosystem

We will start this section with a somewhat simpler cryptosystem that was developed by Goldreich, Goldwasser and Halevi in late-1990s **ggh** called the GGH cryptosystem. This scheme, rather than using ideal lattices (i.e. lattices that are also ideals in the ring of integers), relies on general properties of lattices. Namely, the hardness of the SVP and CVP (see section 2.4).

Idea behind the scheme

The basic GGH cryptosystem, as mentioned before, is based on problem of finding the closest vector in the lattice $\mathcal{L}$ to a given point in the ambient space $\mathbb{R}^n$. We are given two bases, call them $\boldsymbol{B}_{good}$ and $\boldsymbol{B}_{bad}$. The $\boldsymbol{B}_{bad}$ will be our public key and $\boldsymbol{B}_{good}$ the secret key. Our secret message $\boldsymbol{m}$ is represented as a binary vector which we will use to form a linear combination $\boldsymbol{s} = \sum m_i \boldsymbol{v}_i^{bad}$ of the vectors in $\boldsymbol{B}_{bad}$. We now add some small and random[1] error $\boldsymbol{e}$ to obtain the ciphertext $\boldsymbol{c} = \boldsymbol{s} + \boldsymbol{e} = \sum m_i \boldsymbol{v}_i^{bad} + \boldsymbol{e} \in \mathbb{R}^n$ - some point that is not in the lattice, but rather, very close to a point in it. To decrypt, we can use our good basis $\boldsymbol{B}_{good}$ to represent $\boldsymbol{c}$ and Babai's algorithm[2] to

[1] some small note about the "randomness" of this e

[2] Simply stated, if the vectors of the basis are sufficiently orthogonal to one another, then this algorithm solves CVP. However, if the Hadamard ratio is less than <fill in>, the algorithm fails to find the closest vector **babai**

find $\boldsymbol{v}$ and represent it in terms of the basis $\boldsymbol{B}_{good}$ [Krzys: i think i found a mistake in the book by Hoffstein et al coz it says there "bad" instead of "good"] to recover $\boldsymbol{m}$. On the other hand, any eavesdropping adversary that is trying to learn our secret, is left with some bad basis that will be of no help in solving the CVP.

GGH construction - concretely

KeyGen:

- Pick a basis $\boldsymbol{B} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\} \subset \mathbb{Z}^n$ such that they are reasonably orthogonal to one another - i.e. with small Hadamard ratio. We will associatie the vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ as the n-by-n matrix V and let $\mathcal{L}$ be the lattice generated by these vectors. This is our good basis $\boldsymbol{B}_{good}$ - the **private key**.
- Pick an n-by-n matrix U with integer coefficients and determinant ± 1 and compute $W = UV$. The row vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n$ of W are the bad basis $\boldsymbol{B}_{bad}$ of $\mathcal{L}$ - the **public key**.

[Krzys: TODO: finish the algo]

4.2 Learning With Errors

Classical SIS and LWE

Ring-LWE

ring-lwe Toward basing fully homomorphic encryption on worst-case hardness

4.3 Fully Homomorphic Encryption Using Ideal Lattices

explain here how we can construct a really nice homomorphic encryption scheme using ideal lattices **gentry**. present the

4.4 On Ideal Lattices and Learning With Errors Over Rings

this is somewhat too difficult for me i think so ill just present main findings without proofs and details **regev**,
First explain what lattices are.
How do lattices relate to LWE? The secret key is associated with a random vector. then show how ring-lwe satisfies both of our requirements **ring-lwe**, namely, the believed hardness for quantum computers (SVP or approximate SVP) and FHE. Show also the problem with ring-LWE because the lattices that are used there are ideal lattices which obviously possess more structure than "normal" lattices.

5 Implementations

Implementations of lattice-based cryptography: ggh, ring-lwe, ibm, the ggh public key cryptosystem

6 Comparison

Comparison with traditional cryptography: Compare lattice-based cryptography with traditional cryptographic systems, including RSA and Elliptic Curve Cryptography (ECC), in terms of security, efficiency, and implementation.

7 Conclusions

a quote from **intro_cryp**: "In real world scenarios, cryptosystems based on N P-hard or N P-complete problems tend to rely on a particular subclass of problems, either to achieve efficiency or to allow the creation of a trapdoor. When this is done, there is always the possibility that some special property of the chosen subclass of problems makes them easier to solve than the general case"

References

algebra Daniel A. Marcus. *Number fields*. Universitext. Springer-Verlag, New York-Heidelberg, 1977, pp. viii+279. ISBN: 0-387-90279-1.

book Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An introduction to mathematical cryptography*. Second. Undergraduate Texts in Mathematics. Springer, New York, 2014, pp. xviii+538. ISBN: 978-1-4939-1710-5; 978-1-4939-1711-2. DOI: 10.1007/978-1-4939-1711-2. URL: https://doi.org/10.1007/978-1-4939-1711-2.

easy_fhe Craig Gentry. "Computing Arbitrary Functions of Encrypted Data". In: *Commun. ACM* 53.3 (Mar. 2010), pp. 97–105. ISSN: 0001-0782. DOI: 10.1145/1666420.1666444. URL: https://doi.org/10.1145/1666420.1666444.

gentry Craig Gentry. "Fully Homomorphic Encryption Using Ideal Lattices". In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC '09. Bethesda, MD, USA: Association for Computing Machinery, 2009, pp. 169–178. ISBN: 9781605585062. DOI: 10.1145/1536414.1536440. URL: https://doi.org/10.1145/1536414.1536440.

gentry_phd Craig Gentry. "A Fully Homomorphic Encryption Scheme". AAI3382729. PhD thesis. Stanford, CA, USA, 2009. ISBN: 9781109444506.

ggh Oded Goldreich, Shafi Goldwasser, and Shai Halevi. "Public-key cryptosystems from lattice reduction problems". In: *Advances in Cryptology — CRYPTO '97*. Ed. by Burton S. Kaliski. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 112–131. ISBN: 978-3-540-69528-8.

impl_gentry Craig Gentry and Shai Halevi. "Implementing Gentry's Fully-Homomorphic Encryption Scheme". In: *Advances in Cryptology – EUROCRYPT 2011*. Ed. by Kenneth G. Paterson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 129–148. ISBN: 978-3-642-20465-4.

int_scheme Marten van Dijk et al. "Fully Homomorphic Encryption over the Integers". In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by Henri Gilbert. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 24–43. ISBN: 978-3-642-13190-5.

intro_cryp Jill Pipher Jeffrey Hoffstein and Joseph H. Silverman. *An Introduction to Mathematical Cryptography*. Springer New York, NY, 2016. DOI: https://doi.org/10.1007/978-1-4939-1711-2.

katz Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. 2nd. Chapman & Hall/CRC, 2014. ISBN: 1466570261.

primal Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. "On Data Banks and Privacy Homomorphisms". In: 1978.

regev Oded Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". In: *J. ACM* 56.6 (Sept. 2009). ISSN: 0004-5411. DOI: 10.1145/1568318.1568324. URL: https://doi.org/10.1145/1568318.1568324.

ring-lwe Vadim Lyubashevsky, Chris Peikert, and Oded Regev. *On Ideal Lattices and Learning with Errors Over Rings*. Cryptology ePrint Archive, Paper 2012/230. 2012. URL: https://eprint.iacr.org/2012/230.