University of Athens

$\mu\prod\lambda\forall$

Graduate Program in Logic, Algorithms and Computation

Fully Homomorphic Encryption

Author:
Christos Litsas

A Thesis Presented in Partial Fulfillment of the Requirements for the Degree of Master of Science.

December 13, 2010

preface

In this thesis we are interesting on fully homomorphic encryption. The problem of constructing a fully homomorphic public key had been open for more than three decades and was first solved by Craig Gentry. Here we follow Craig Gentry's construction that came out in 2009 in his breakthrough paper [1]. We focus on presenting the scheme and give the mathematical background that the whole construction uses. Furthermore, we make some instantiations on the scheme's subroutines, algorithms and parameters in the interest of how could it be implemented for real use.

First we present the problem and some possible and useful applications of it. After we present three classic partial homomorphic schemes. The half part of the final construction is the crucial property of bootstrappability which is examined next. Bootstrappability, is a special property of encryption schemes that as we prove if we find a leveled homomorphic bootstrappable scheme then we can build a fully homomorphic one. Bootstrappability also gives a hint on where and how to search for a homomorphic scheme. The direction that bootstrapping points to is encryption schemes with swallow decryption circuits, so integer lattices come into play, since previous schemes that are based on lattices have this property. Integer lattices, though, usually produce huge public keys something that is fixed with a special category of lattices, the ideal lattices. After the basic background and the tools that we need we give an abstract scheme that has only limited homomorphic capabilities. Then, our main goal is to modify this scheme and create a fully homomorphic one by bootstrapping it.

Acknowledgements

Special thanks to:

The fantastic program of MPLA for providing my brain with better and faster decryption algorithms.

Gregory Kounadis for driving me through bits and bytes.

Andreas Toumasis for giving me some nice explanations about various algebraic notions.

Manos Androulakis for playing the role of ‘deux ex machina‘.

Finally, I would be unfair not to mention my respect to Alice and Bob.

Contents

Part I

Preliminaries

Chapter I

Preliminaries

THE first task that has to be treated in texts that involve mathematics is the language, its symbols and some useful theory for what that follows. In this chapter we come out with some basic knowledge of algebra and group theory. Through all this book we use the conventional symbols $\forall, \exists$ for universality and existence respectively. We also use the operators $+, -, \cdot, /$ to define the usual operations unless otherwise stated. We also follow the common notation for some standard sets, $\mathbb{N}$ for natural numbers, $\mathbb{Z}$ for integers, $\mathbb{Q}$ for rationals and $\mathbb{R}$ for real numbers. For sets we use $\cap, \cup$ as the intersection and the union respectively. Often when when $a, b \in \mathbb{Z}$ and $a < b$ we use the symbolism $[a, b]$ to denote the set of integers containing a, b and every intermediary integer. We follow the same symbolism and for closed sets $[x, y] = \{u : x \leq u \leq y\}$.

Next we don't give proofs of the theorems and lemmas that we present since this is not the purpose of this thesis. We suggest that the reader refer to texts [19], [20] and [21] for further analysis on these subjects.

I.I Algebra

I.I.I Number Theory

Number theory has become maybe the most common used field of mathematics in cryptography so far. Its advantage over the other fields is the simplicity in representing every single integer number. Furthermore, there are mathematical structures that have been developed through all these years using as footing the natural numbers and their properties. We start by giving some key properties of integers.

Definition 1. *Let $d, n \in \mathbb{Z}$ we say that d* divides *n and write $d \mid n$ if there exist an integer c so that $n = c \cdot d$ if no such a number exist we say that d* does not divide *n and write $d \nmid n$.*

The following properties come after the previous definition:

1. $\forall n \in \mathbb{Z}$ $n \mid n$, $1 \mid n$ and $n \mid 0$.
2. $\forall n, m, d \in \mathbb{Z}$ if $d \mid n$ and $n \mid m$ then $d \mid m$.
3. $\forall n, m, d, a, b \in \mathbb{Z}$ if $d \mid n$ and $d \mid m$ then $d \mid (a \cdot m + b \cdot n)$.
4. $\forall n, d \in \mathbb{Z}$ with $n \neq 0$ if $d \mid n$, then $|d| \leq |n|$.
5. $\forall n, d \in \mathbb{Z}$ if $d \mid n$ and $n \mid d$, then $d = n$.

Theorem 1 (Integer Division)**.** *Let $a, b \in \mathbb{Z}$, $b \neq 0$ then there are, two unique, integers q, r such that*

$$a = b \cdot q + r, \text{ with } 0 \leq r < r$$

Theorem 2 (Greatest Common Divisor)**.** *Let $a, b \in \mathbb{Z}$, we call the natural number d* greatest common divisor *of a, b and write $d = \gcd(a, b)$ if:*

1. *$d \mid a$ and $d \mid b$.*
2. *$\forall d' \in \mathbb{N}$ such that $d' \mid a$ and $d' \mid b$ it also holds that $d' \mid d$.*

Definition 2. *Let $p \in \mathbb{N}$, $1 < p$ we call p* prime number *if it has exactly two distinct natural number divisors: 1 and p.*

Two numbers $a, b \in \mathbb{N}$ called relational primes *if $\gcd(a, b) = 1$.*

Theorem 3 (Fundamental Theorem of Arithmetic)**.** *Every natural number $a \in \mathbb{N}$, $a > 1$ can be written as a unique product of prime numbers.*

The property of a number being prime is called *primality.* Finding prime numbers is a prickly task, until today there are no known fast algorithms to implement this. However, prime numbers are needed by numerous cryptographic schemes, consequently there are a bunch of algorithms that trying to produce random prime numbers. For most of them assurance sacrificed at the altar of efficiency, thereafter the practical algorithms are probabilistic (i.e. produce prime numbers with a high probability).

Definition 3. *Denote by $\varphi(n)$ the number of positive integers less than or equal to n. $\varphi(n)$ is named after Swiss mathematician Leonhard Euler* Euler's Totient Function.

Now we present two fundamental theorems of number theory. They are both parts of many cryptosystems in the sense of providing low time computation of big exponents in particular algebraic structures.

Theorem 4 (Fermat's Little Theorem)**.** *Every natural number $p \in \mathbb{N}$, if p is a prime number, then for any integer a, $a^p - a$ id divided by p.*

Theorem 5 (Eulers's Theorem)**.** *For every natural number $n \in \mathbb{N}$, then for any integer a, $a^{\varphi(n)} - 1$ is divided by n.*

I.I.II Groups, Rings, Ideals, Quotient Rings, Fields

One of the most fundamental mathematic tools in cryptography is the sense of the *group*

Definition 4. *Let G be a set and $\diamond$ an operation that maps a pair of elements of G to a single element of G the we cal the set $\{G, \diamond\}$ a group if the following conditions are satisfied:*

Closure *$\forall a, b \in G$, the result of the operation, $a \diamond b$, is also in G*

Associativity *$\forall a, b, c \in G$, it holds that $(a \diamond b) \diamond c = a \diamond (b \diamond c)$*

Identity element *$\exists 0_G \in G$, : $\forall a \in G$, the equation $0_G \diamond a = a \diamond 0_G = a$ holds.*

Inverse element *$\forall a \in G$, $\exists b \in G$ such that $a \diamond b = b \diamond a = 0_G$.*

In a group $\{G, \diamond\}$ does not hold that $\forall a, b \in G$ $a \diamond b = b \diamond a$. Take as example G to be the set of all inversed matrices of $\mathbb{R}^{n \times n}$ and $\diamond$ to be the matrix multiplication operation, then $\{G, \diamond\}$ is a group but generally it holds that $A \cdot B \neq B \cdot A$.

Now, if $\{G, \diamond\}$ is a group with the additional property $\forall a, b \in G$ $a \diamond b = b \diamond a$ then we say that $\{G, \diamond\}$ is an *Abelian* or *Commutative Group*.

Definition 5. *Let $\{G, \diamond\}$ be a group and $H \subseteq G$, then if $\{H, \diamond\}$ forms a group we say that $\{H, \diamond\}$ is a subgroup of $\{G, \diamond\}$ and we write $\{H, \diamond\} \leq \{G, \diamond\}$.*

Definition 6 (Group Order)**.** *For a group $\{G, \diamond\}$ we denote by $|G|$ the number of elements of $\{G, \diamond\}$. $|G|$ also called the* order *of the group.*

We also denote by a^i the operation $a \diamond a \diamond \cdots \diamond a$ where a appears i times.

One of the most fundamental theorems in group theory is *Lagrange's theorem* which combines the number of elements of a group and of its subgroups.

Theorem 6 (Lagrange's Theorem)**.** *For any finite group G, and every subgroup H of G it holds that $|H| \mid |G|$.*

Of special importance are groups that are generated only by a single element of the group. Their importance lies at the fact that the one needs only one element and the operation and has the whole group.

Definition 7. *We say that $\{G, \diamond\}$ is a cyclic group if $\exists a \in G$ such that $\forall b \in G\ \exists i \in \mathbb{N} : b = a^i$.*

Example 1. *Consider the set $\mathbb{Z}$ with the operation $+$, then $\{\mathbb{Z}, +\}$ is an abelian group of infinite order.*

Example 2. *For an example of a set and an operation that are not form a group take the set $\mathbb{N}$ with the operation $+$. Then, as one can see, there is no inverse element for the elements of $\mathbb{N}$.*

Example 3. *The set $2\mathbb{Z}$, that is the of the multiples of 2, and the operation $+$. The set $\{2\mathbb{Z}_n, +\}$ is an abelian group of infinite order. Also, the set $\{4\mathbb{Z}_n, +\}$ is an abelian group of infinite order and it holds that $\{4\mathbb{Z}_n, +\} \leq \{2\mathbb{Z}_n, +\}$*

Next we give the notion of the ring, they play a key role in cryptography and also the majority of schemes base their mathematical support on them.

Definition 8. *Let R is a set and $\diamond$, $\circ$ be two operators that map pairs of elements of R to a single element of R, we say that the triple $\{R, \diamond, \circ\}$ is a ring if the following properties hold.*

Closure *$\forall a, b \in R$, the result of the operations, $a \diamond b$ and $a \circ b$ is also in R.*

Associativity *$\forall a, b, c \in G$, it holds that $(a \diamond b) \diamond c = a \diamond (b \diamond c)$ and also $(a \circ b) \circ c = a \circ (b \circ c)$.*

Identity element of $\diamond$ *$\exists 0_R \in R, : \forall a \in R$, the equation $0_R \diamond a = a \diamond 0_R = a$ holds. (The same property does not hold for operator $\circ$ in general but when holds we denote the identity element of $\circ$ with 1_R)*

Inverse element for $\diamond$ *$\forall a \in R$, $\exists b \in R$ such that $a \diamond b = b \diamond a = 0_R$.*

Commutativity of $\diamond$ *$\forall a, b \in R$, it holds that $a \diamond b = b \diamond a$.*

Distributive Law *$\forall a, b, c \in R$, it holds that $(a \diamond b) \circ c = a \circ c \diamond b \circ c$.*

Distributive Law *$\forall a, b, c \in R$, it holds that $(a \circ b) \diamond c = a \diamond c \circ b \diamond c$.*

If in ring $\{R, \diamond, \circ\}$ holds that $\forall a, b \in R$ $a \circ b = b \circ a$ we call this ring a *commutative* ring.

Remark 1. *It is easy and also useful to see that if $\{R, \diamond, \circ\}$ is a ring then $\{R, \diamond\}$ is a group.*

Among all these algebraic structures there is also a stronger one which is fundamental for the further work that we present here. This structure called *ideal* and its definition is:

Definition 9. *Consider the ring $\{R, \diamond, \circ\}$ and $I \subseteq R$, $I \neq \emptyset$. I named ideal of R and we denote this relation $I \lhd R$ if the following two conditions hold:*

- *$\forall a, b \in I$ holds that: $a \diamond b \in I$.*
- *$\forall a \in I$, $\forall r \in R$ $r \circ a \in I$ and also $a \circ r \in I$.*

One can see that the set $\{0_R, \diamond, \circ\}$ is an ideal of $\{R, \diamond, \circ\}$ this called the *trivial ideal*, also the whole ring is an ideal of it self. An ideal of $\{0_R, \diamond, \circ\}$ different than previous two called a *genuine ideal*.

Let $\{R, \diamond, \circ\}$ be a ring and $S \subseteq R$. The intersection of all the ideals of R that contain the subset S is the smaller ideal of $\{R, \diamond, \circ\}$ which contains the set S. We refer to this as the *ideal produced* by the set S and we write $\langle S \rangle$. That is, $\langle S \rangle = \bigcap \{I : I \lhd R, S \subseteq I\}$.

We say that an ideal I of ring $\{R, \diamond, \circ\}$ is *principal* if it is generated by a single element $s \in R$. Such an ideal is denoted $I = \langle s \rangle$.

For two ideals I_1, I_2 of ring $\{R, \diamond, \circ\}$ denote the set $\{a + b : a \in I_1, b_i n I_2\}$ by $I_1 + I_2$.

Definition 10. *Let $\{R, \diamond, \circ\}$ be a ring and I, J two ideals are* relatively prime *if $I + J = R$.*

Definition 11. *Consider the ring $\{R, \diamond, \circ\}$, we say that $\{R, \diamond, \circ\}$ is a* ring of principle ideals *if $\forall I \lhd R$, I is a principle ideal.*

Definition 12. *Let $\{R, \diamond, \circ\}$ be a ring. If $\exists a \neq e, b \in R : a \circ b = 0_R$ then we call a a divisor of zero of the ring $\{R, \diamond, \circ\}$.*

Definition 13. *We define a relation in ring $\{R, \diamond, \circ\}$ which depends on an ideal $I \lhd R$ as follows: we say that two elements a, b of ring $\{R, \diamond, \circ\}$ are equivalent and write $a \equiv b \mod I$ iff $a - b \in I$.*

The relation that defined just above is an equivalence relation. Also the the equivalence class of an element $a \in R$ denoted by $a + I = \{a + y : y \in I\}$.

Definition 14. *A mapping is a rule which maps each element x of one set X to a, unique, element y in another set Y. This procedure expressed as the function, say f, and denoted by $f(x) = y$. There can only be said to be a mapping from X to Y if $\forall x \in X\ \exists y \in Y : f(x) = y$ and $\forall x \in X$ if $(f(x) = y \wedge f(x) = y' | y, y' \in Y) \Rightarrow y = y'$.*

A mapping f is said to be *surjective* or *onto* if its image is equal to codomain. An *injective function* is a mapping f such that $\forall x, x' \in R\ f(x) \neq f(x') \Rightarrow x \neq x'$. The set X then called the *domain* of the mapping and the set Y is the *codomain* of the mapping or the *image* of the function.

Here we insist on emphasizing the fundamental notion of *injective functions* since their special property induces an important cryptographic requirement. In every cryptographic scheme we need to be able to inverse the encrypted data in order to decrypt them, since cryptography established on using functions the use of injective functions is straightforward.

Throughout this book the most fundamental and scrutinized studied concept is the notion of homomorphism. In this initial phase we can provide the notion of homomorphism since we have defined all that we need for this purpose.

Definition 15. *Let $\{R, \diamond, \circ\}$, $\{S, \diamond', \circ'\}$ be rings. A mapping $f : R \to S$ called* ring homomorphism *iff $\forall a, b$ holds that $f(a \diamond b) = f(a) \diamond' f(b)$ and also $f(a \circ b) = f(a) \circ' f(b)$*

If the function f is $1-1$ (i.e. f is injective function) then we say that f is *monomorphism.* And when f is an surjective function we call f *epimorphism*

Definition 16. *We say that $\{G_1, \diamond\}$ is a bilinear group if there exists a group $\{G_2, \diamond'\}$ and a bilinear map as below.*

- *$\{G_1, \diamond\}$ and $\{G_2, \diamond'\}$ are two cyclic groups of finite order n.*
- *g is a generator of $\{G_1, \diamond\}$.*

- *e is a* bilinear map $e : G_1 \times G_1 G_2$ *. In other words, $\forall u, v \in G_1$ and $\forall a, b\mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{a \cdot b}$. We also require that $e(g, g)$ is a generator of $\{G_2, \diamond'\}$.*

Definition 17. *Let $\{R, \diamond, \circ\}$ be a ring. If $\exists a \neq e, b \in R : a \circ b = 0_R$ then we call a a divisor of zero of the ring $\{R, \diamond, \circ\}$.*

Definition 18. *Consider the commutative ring $\{R, \diamond, \circ\}$ with $1_R \in R$. We call $\{R, \diamond, \circ\}$ integrity domain when it has no nontrivial divisors of zero.*

Definition 19. *Let $\{R, \diamond, \circ\}$ be a commutative ring with $1_R, 0_R \in R$, $1 \neq 0_R$ and $\forall a \in R\ \exists b \in R : a \circ b = 1_R$ then we say that $\{R, \diamond, \circ\}$ is a field.*

Theorem 7. *Every finite integral domain is a field.*

Example 4. *Consider the set $\mathbb{Z}$ with the operations $+, \cdot$, then $\{\mathbb{Z}, +, \cdot\}$ is a commutative ring. Also one can prove that $\{\mathbb{Z}, +, \cdot\}$ is an integrity domain.*

Furthermore if we consider the set $\{2\mathbb{Z}\}$ we can prove that is an ideal of $\{\mathbb{Z}, +, \cdot\}$, that is $\{2\mathbb{Z}, +, \cdot\} \lhd \{\mathbb{Z}, +, \cdot\}$. Also the ideal $\{2\mathbb{Z}\}$ is principle since it produced by the element 2, that is $\{2\mathbb{Z}\} = \langle 2 \rangle$.

Example 5. *Take the set $\mathbb{R}$ with the operations $+, \cdot$, then $\{\mathbb{Z}, +, \cdot\}$ is a field.*

The set $\{\mathbb{R}, +, \cdot\}$ is maybe the most classic example for a field. For our purposes though, this is a bad example of field. Since cryptography implemented through computers and not all elements of $\mathbb{R}$ can be represented by finite precision. Last fact is a huge disappointment because fields are a powerful mathematical tool to be used by cryptography since they offer multiply and addition inverses which is a fundamental part in cryptography by the cause of the necessity to reverse operations that produced encrypted data.

Due to this scragginess of space $\mathbb{R}$ we present an example of a finite field.

Example 6. *Consider the set $\mathbb{Z}_p$, where $p \in \mathbb{N}$ is a prime number, with the operations $+, \cdot$, then $\{\mathbb{Z}_p, +, \cdot\}$ is a field.*

The field $\mathbb{Z}_p$ is a perfect example of a base to build cryptosystems on it. It offers accuracy in the representation of every of its elements, also it offers addition and multiplication inverses and predominantly this field has an astonishing property of being fully described only by the element p.

I.I.III Polynomials

Definition 20. *Let R be a ring with 1, then there is a ring $\mathcal{R}$ with the same unity witch contains R as a subring and also has the following properties:*

- *There is an element $x \in \mathcal{R}$ such that $\forall r \in R \; r \cdot x = x \cdot r$.*
- *Every element of $\mathcal{R}$ has a representation of the form $r_0 + r_1 x + \cdots + r_n x^n$, where $n \in \mathbb{N}$ and $r_i \in R$, $\forall i \in [0, n]$.*
- *If $r_0 + r_1 x + \cdots + r_n x^n = s_0 + s_1 x + \cdots + s_m x^m$, where $n, m \in \mathbb{N}$, $n \leq m$ and $r_i \in R$, $\forall i \in [0, n]$, and $s_i \in R \; \forall i \in [0, m]$, then $r_i = s_i$, $\forall i \in [0, n]$ and $s_i = 0_R$, $\forall i \in [n+1, m]$*

We denote this ring with $\mathcal{R}[x]$ and we call its elements (univariable) polynomials. The elements of $\mathcal{R}[x]$ denoted by $f(x), g(x), h(x), \ldots$.

We call *mononym* every element of a ring $\mathcal{R}[x]$ which is of the form rx^i. We define the *degree* of a polynomial $f(x)$ to be the biggest exponent of the $x_i's$. If r is the coefficient of the term $x^{deg(f(x))}$ then we say that r is the *leading coefficient* of $f(x)$. A polynomial which leading coefficient is 1_R called *monic*. And a polynomial is *constant* if its degree is zero and also the coefficient of term x^0 is not the 0_R.

Theorem 8 (Polynomial Division)**.** *Consider the ring $\mathcal{R}[x]$, let $f(x), g(x) \in \mathcal{R}[x]$, $g(x) \neq 0_R$ then there are, unique, polynomials $\pi(x), \upsilon(x) \in \mathcal{R}[x]$ such that $f(x) = \pi(x) \cdot g(x) + \upsilon(x)$, with $\upsilon(x) = 0$ or $deg(\upsilon(x)) < deg(g(x))$.*

Definition 21. *A polynomial $f(x) \in \mathcal{R}[x]$ is* irreducible *over the ring R when*

$$\forall g(x), h(h) : f(x) = g(x) \cdot h(x) \Rightarrow (g(x) = 1_R \vee h(x) = 1_r)$$

Theorem 9. *The ring $\mathcal{R}[x]$ is a ring of principle ideals.*

Definition 22 (Greatest Common Divisor)**.** *For polynomials $f(x), g(x) \in \mathcal{R}[x]$ with at least one different than 0_R we call a polynomial $d(x) \in \mathcal{R}[x]$* greatest common divisor *of $f(x), g(x)$, symbolized by* $\gcd(f(x), g(x))$*, if it satisfies the following conditions:*

1. *$d(x)|f(x)$ and $d(x)|g(x)$, that is, $d(x)$ is a common divisor for $f(x), g(x)$.*
2. *$d(x)$ is monic.*
3. *If $d'(x) \in \mathcal{R}[x]$ and $d'(x)|f(x)$ and $d'(x)|g(x)$ then $d'(x)|d(x)$.*

When for the polynomials $f(x), g(x) \in \mathcal{R}[x]$ holds that $gcd(f(x), g(x)) = 1_R$ we asy that they are *relatively prime.*

Theorem 10. *Let $f(x), g(x) \in \mathcal{R}[x]$ be polynomials with at least one of them different than 0_R. Then there is always a polynomial $d(x) \in \mathcal{R}[x]$ which is the greatest common divisor of $f(x), g(x)$. Also, there are $a(x), b(x) \in \mathcal{R}[x]$: $gcd(f(x), g(x)) = a(x) \cdot f(x) + b(x) \cdot g(x)$.*

I.I.IV Linear Algebra

Definition 23. *Let $\mathbb{F}$ be a field and V a, non-empty, set with two binary operations $\diamond, \circ$, $\{V, \diamond, \circ\}$ is a* vector *or* linear space *over the field $\mathbb{F}$ if the following conditions are satisfied:*

1. $\diamond : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$, $\circ : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$.
2. $\forall a, b \in V \Rightarrow a \diamond b \in V$, $\forall a \in V, l \in \mathbb{F} \Rightarrow l \circ a \in V$.
3. $\forall a, b, c \in V \Rightarrow (a \diamond b) \diamond c = a \diamond (b \diamond c)$.
4. $\exists 0_V \in V \Rightarrow \forall a \in V : a \diamond 0 = 0 \diamond a = a$.
5. $\forall a \in V \exists b \in V \Rightarrow a \diamond b = b \diamond a = 0_V$, *we denote such a b by* $-a$.
6. $\forall a, b \in V \Rightarrow a \diamond b = b \diamond a$.
7. $\forall a, b \in V, l \in \mathbb{F} \Rightarrow l \circ (a \diamond b) = l \circ a \diamond l \circ b$.
8. $\exists 1_{\mathbb{F}} \in \mathbb{F} : \forall a \in V \Rightarrow a \circ 1_{\mathbb{F}} = 1_{\mathbb{F}} \circ a = a$.
9. $\forall k, l \in \mathbb{F}, a \in V \Rightarrow (k \diamond l) \circ a = k \circ a \diamond l \circ a$.
10. $\forall k, l \in \mathbb{F}, a \in V \Rightarrow (k \circ l) \circ a = k \circ (l \circ a)$.

We call the elements of V *vectors* and those of $\mathbb{F}$ *scalars.* The operation $\circ$ is also called *scalar* operation.

Definition 24. *A nonempty subset W of a vector space V that is closed under the operations $\diamond$ and $\circ$ is called a subspace of V.*

Subspaces of V are vector spaces. The intersection of all subspaces containing a given set S of vectors is called its *span*, and is the smallest subspace of V containing the set S. Expressed in terms of elements, the span is the subspace consisting of all the linear combinations of elements of S, we call it *the space spanned by S.*

Definition 25. *Let a vector space V with the operations $\diamond$ and $\circ$ over the field $\mathbb{F}$. Consider the finite subset of V, $S = \{a_1, a_2, ..., a_n\}$. This family of vectors is* linearly independent *if there exist $l_1, l_2, ..., l_n \in \mathbb{F}$ with at least one of them different than 0_V, such that $l_1 \circ a_1 \diamond l_2 \circ a_2 \cdots l_n \circ a_n = 0_V$. If $\forall l_1, l_2, ..., l_n \in \mathbb{F}$ such that $l_1 \circ a_1 \diamond l_2 \circ a_2 \cdots l_n \circ a_n = 0_V$ implies $l_1 = l_2 = ... = l_n = 0_V$ then we say that S is* linearly dependent.

In expressions of the form $l_1 \circ a_1 \diamond l_2 \circ a_2 \cdots l_n \circ a_n = a$, where $a \in V$ we say that a is a *linear combination* of $\{a_1, a_2, ..., a_n\}$.

Definition 26. *In a vector space V defined as above consider a set of elements of V, $S = \{a_1, a_2, ..., a_n\}$, we say that S is a* basis *of V if every element of V can be written as a linear combination of the elements of S.*

One can see that a basis is a linearly independent spanning set.

Definition 27. *A subset $S = \{a_1, a_2, ..., a_n\}$ of a vector space V called* maximum linearly independent subset *of V when $\forall a \in V \Rightarrow$ the set $S \cup \{a\}$ is linearly independent.*

Theorem 11. *Let V be a vector space with the operations $\diamond$ and $\circ$ over the field $\mathbb{F}$. And let $S = \{a_1, a_2, ..., a_n\} \subseteq V$. S is a basis of V iff is a maximum linearly independent subset of V.*

Theorem 12. *Every possible basis of a vector space V have the same number of elements.*

The number of elements of the bases of a vector space is called the *dimension* of the space. When a space has n elements in its bases we say that this space is of *n-dimensions*.

Here we can define a common used function for the elements of a vector space V over field $\mathbb{F}$ fitted with the operations $\diamond$ and $\circ$.

Definition 28. *A* norm *is a mapping $\|\cdot\| : V \to \mathbb{R}$ with the following properties:*

Separates Points $\forall a \in V, a \neq 0_V \Rightarrow \|a\| > 0$ *and if* $a = 0_V$ *then* $\|a\| = 0$.

Positive Homogeneity $\forall a \in V, l \in \mathbb{F} \Rightarrow \|l \circ a\| = |l| \cdot \|a\|$.

Triangle Inequality $\forall a, b \in V \Rightarrow \|a \diamond b\| \leq \|a\| \diamond \|b\|$.

I.I.V Bases Orthogonalization

In this subsection we need to define the sense of orthogonality for the elements of a vector space. In this direction we first define a structure of a vector space called inner product.

Definition 29. *For a real vector space V with the operations $\diamond$ and $\circ$, a scalar operation, over the field $\mathbb{F}$, an inner product is a mapping $\langle\rangle : V \times V \to \mathbb{R}$ that satisfies the following four properties:*

Let $a, b, c \in V$ and $l \in \mathbb{F}$ then:

- $\langle a \diamond b, c\rangle = \langle a \diamond c\rangle \diamond \langle b \diamond c\rangle$.
- $\langle l \circ a, b\rangle = l \circ \langle a, b\rangle$.
- $\langle a, b\rangle = \langle b, a\rangle$.
- $\langle a, a\rangle \geq 0$*, equality holds only when* $a = 0_V$.

Definition 30. *Let a, b elements of a vector space V, we say that a, b are orthogonal iff $\langle a, b\rangle = 0$.*

Definition 31. *An* orthonormal basis *of a vector space V is a subset S of V such that $\forall a, b \in S, a \neq b \Rightarrow \langle a, b\rangle = 0$ and also the space spanned by S is the whole space V.*

I.II Topology

I.II.I Metric Spaces, Sequences

Definition 32. *Let X be an arbitrary set. A mapping $\rho : X \times X \to \mathbb{R}$ called* metric in $\mathbb{R}$ *if the following conditions are satisfied:*

Non-negativity $\forall x, y \in X \Rightarrow \rho(x, y) \geq 0$.

Identity of Indiscernibles $\forall x, y\ \rho(x, y) = 0 \Leftrightarrow x = y$.

Symmetry $\forall x, y\ \rho(x, y) = \rho(y, x)$.

Triangle Inequality $\forall x, y, z \in X \Rightarrow \rho(x, y) \leq \rho(x, z) + \rho(y, z)$.

The pair (X, ρ) is called a metric space*.*

Definition 33. *For an be an arbitrary set X we define a* sequence *in X to be a mapping from $\mathbb{N}$ to X. We denote by a_i the value of the sequence when taking as parameter a natural number i.*

Theorem 13 (Cauchy-Schwarz inequality)**.** *Let $x_1, x_2, ..., x_n$ and $y_1, y_2, ..., y_n$ are all real numbers then the following inequality holds:*

$$\left(\sum_{i=1}^{n} x_i \cdot y_i\right)^2 \leq \left(\sum_{i=1}^{n} x_i^2\right) \cdot \left(\sum_{i=1}^{n} y_i^2\right)$$

Definition 34. *A subset Y of a metric space (X, ρ) is called* open *if,*

$$\forall y \in Y \, \exists \varepsilon > 0 \, \forall x \in X : \rho(x, y) < \varepsilon \Rightarrow x \in Y$$

Equivalently, Y is open if every point in Y has a neighborhood contained in Y.

A set Y is called closed *if its complement (i.e. the elements of X that are not in Y) is open.*

Definition 35. *Let (X, ρ) be a metric space and $x \in X, \varepsilon > 0$. The set*

$$\mathcal{S}(x, \varepsilon) = \{y \in X : \rho(x, y) < \varepsilon\}$$

called a ball of center x and radius ε*.*

Definition 36. *Let (X, ρ) be a metric space, a sequence (x_n) of elements of X* converges to $x \in X$ *if $\forall \varepsilon > 0 \, \exists n_0 \, \forall n > n_0 \in \mathbb{N} : \rho(x_n, \varepsilon) < \varepsilon$. We denote the convergence of a sequence (x_n) to $x \in X$ by $x_n \to x$ or $\lim_{n \to \infty} x_n = x$.*

Definition 37. *A* subsequence of a sequence $(x_n) \in X$ *is every sequence (x_{n_k}), where (n_k) is a genuinely ascending sequence of natural numbers (i.e. $m < l \Rightarrow n_m > n_l$). That is a subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements and denoted by $(y_n) \subseteq (x_n)$.*

Theorem 14. *Let (X, ρ) be a metric space and $A \subset X$. A is closed iff every convergent sequence of A converges in an element of A.*

Theorem 15. *Let (X, ρ) be a metric space, a sequence (x_n) of X converges to $x \in X$ iff every subsequence of (x_n) converges to x.*

Definition 38 (Cauchy sequence)**.** *Let (X, ρ) be a metric space and (x_n) a sequence of elements of X we say that (x_n) is a* Cauchy sequence *if*

$$\forall \varepsilon > 0 \, \exists n_0 \, \forall m, n \geq n_0 \in \mathbb{N} : \rho(x_n, x_m) < \varepsilon,$$

Definition 39 (Complete Metric Space)**.** *A metric space (X, ρ) is said to be* complete (or Cauchy) *if every Cauchy sequence $(x_n) \in X$ has a limit that is also in X. Alternatively if every Cauchy sequence in X converges in X.*

Definition 40 (Compact Metric Space)**.** *A metric space (X, ρ) called* compact *if if each of its open covers (i.e. a collection of open sets such that X is a subset of their union.) has a finite subcover. Formally, if $\{A_i\}$, $i \in U \subseteq \mathbb{N}$ a collection of open subsets of X such that $X \subseteq \bigcup_{i \in U} A_i$, there is a finite set $W \subseteq U$ such that $X \subseteq \bigcup_{i \in W} A_i$.*

Definition 41 (Compact Metric Space)**.** *A metric space (X, ρ) called* sequentially compact *if every sequence $(x_n) \in X$ has a convergent subsequence.*

Theorem 16. *Every compact metric space is sequentially compact. That is, if (X, ρ) is a metric space and (x_n) a sequence in X, then there is (y_n) converges.*

I.III Statistics and Probability theory

I.III.I Discrete Probability

Probability is a way of expressing knowledge or belief that an event will occur or has occurred. Its importance in the world of cryptography lies on the necessity for one to prove that in average case a cryptographic scheme resists at attacks from eavesdropers. That is because no one can talk about absolute security since an attacker at least can guess the meaning of an encrypted message.

Sample space Ω of a random process is the set of all possible outcomes of the process. We call an element $\omega \in \Omega$ *sample*.

Let Ω be a sample space, every subset of Ω is called an *event*.

A useful notation for the continuation is the symbol $\wp(S)$ (often replaced by 2^S), which represents the set of all subsets of an arbitrary set S. $\wp(S)$ called the *power set of S*.

Definition 42. *Let A be an event in the sample space Ω, then the* probability *of A is given by the formula $P(A) = N(A)/N$, where $N(A)$ is the number of elements of A and N the number of elements of the sample space Ω.*

For a sample space Ω the probability has the following properties:

1. *$P(A) \geq 0$ for every sample A of $\wp(\Omega)$).*
2. *$P(\Omega) = 1$.*

3. *$P(A \cup B) = P(A) + P(B)$ for every disjoint sets A, B (i.e. $A \cap B = \emptyset$) of $\wp(\Omega)$).*

In the sample space Ω we define the *conditional probability* of an event $A \in \wp(\Omega)$ assuming that event $B \in \wp(\Omega)$ has occurred, denoted $P(A|B)$, equals

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Definition 43 (probability space)**.** *We call* probability space *a triple* $(\Omega, \mathcal{A}, P)$ *consisting of the following three parts:*

1. *A* sample space Ω*, which is the set of all possible outcomes.*

2. *A set of* events $\mathcal{A}$*, where each event is a set containing zero or more outcomes.*

3. *The assignment of probabilities to the events, that is, a function P from events to probability levels.*

Random variable is a way of assigning a value to each possible outcome of a random event and is denoted by the function $X : \Omega \to \mathbb{R}$. In this way, X is a function that assigns a real value x to every single element $\omega \in \Omega$, namely $X(\omega) = x$.

Definition 44. *Let $(\Omega, \mathcal{A}, P)$ be a probability space and $X : \Omega \to \mathbb{R}$ a random variable. Consider the function F defined by the relation:*

$$F(x) = P(X \leq x) = P[\{\omega \in \Omega : X(\omega) \leq x\}]$$

we call the function F a distribution function *of the random variable X.*

Definition 45. *Let $(\Omega, \mathcal{A}, P)$ be a probability space and $X : \Omega \to \mathbb{R}$, $: \Omega \to \mathbb{R}$ two random variables. We define the* statistical distance Δ *by:*

$$\Delta[X, Y] = \frac{1}{2} \sum_{\omega \in \Omega} |P_X(\omega) - P_Y(\omega)|$$

I.IV Cryptography

We give now some basic notions of cryptography. The present section exists in order to establish a common language and symbolization of cryptography notions.

We start by building a story with a couple of protagonists. Suppose that a boy called *Bob* and a girl, *Alice* desire to exchange some secret information over an insecure channel. Briefly, we say that a communication occurs over an insecure channel if it is feasible for someone that does not participating on it to steal data from the conversation. This unknown, but existing, person since is not someone known with his name is symbolized by $\mathcal{A}$ and called *adversary*. The whole story turns around the effort of Alice and Bod to transpose their data without revealing anything to any adversary. We assume that our protagonists exchange some data consisting of bits. We call these data, *messages*.

Since Alice and Bob are informed that there is someone who tries to steal some of their data, they decide to *encrypt* them. That is, they try to map every bit or set of bits of their data to another bit or set of bits respectively so that the real meaning of their data be vanished. But don't forget that their main goal is to exchange their true data, so the previous process deflects them from it. The only way to bring about some good results is to find a way so that the mapping that each of them apply to be reversible in a standard an efficient way acknowledged only by them. If this happens then they can start transmitting their encrypted data over every insecure channel without being afraid of any *eavesdropper* since only they know how to *decrypt* a message (i.e. to reverse the mapping).

Beware of the importance of the mapping, in fact is the main tool that protects them from any adversary. In consequence, this mapping have to be treated in order to become as much easy reversible as it could be under the condition that the same does not hold when an adversary tries to reverse it.

There is one big issue related to the mapping. Since the way of reversing any of their mappings must be known to each of them, one more problem arises. They have to fabricate a contrivance to help them exchange their secret informations about the mapping, which from on will be referred as their *keys*.

One technique to overcome the presence of the adversaries when they change their keys is to do it only once but do it in person. In case they follow this strategy we say that they have construct a *private key scheme*. Maybe the above seems as the only way to deflect $\mathcal{A}$ from knowing about their secret keys. Nevertheless, there is one more trick to keep their keys secretly from $\mathcal{A}$.

The second contrivance comes with an unforeseen approach for key exchanging, by which Alice and Bob won't exchange keys. They just create their own keys consisting of two parts, labeled the *public* and the *private* keys the one of which is the reversed operation for the other. Then they publish their public keys for one another. As a consequence they encrypt their messages with the public known keys, afterward, once they have taken the encrypted messages, they can start decryption with their private keys.

To formally describe all the above we next give some definitions for some basic essences of cryptography.

Private and Public Key Cryptography

Suppose that Alice and Bob want to exchange their data over an insecure channel. We denote by $\mathcal{P}$ the set of all possible messages (*plaintexts*), and by $\mathcal{C}$ the set of all possible encrypted messages (*ciphertexts*). By $\mathcal{K}$ we refer to the space of all keys that can be used in such a process.

A *private (or symmetric) cryptosystem* consists of the following parts:

1. A function $\text{Encrypt}_{\mathcal{E}} : \mathcal{P} \times \mathcal{K} \to \mathcal{C}$ called *encryption function.*
2. A function $\text{Decrypt}_{\mathcal{E}} : \mathcal{C} \times \mathcal{K} \to \mathcal{P}$ with the additional property
$$\forall m \in \mathcal{P}, k \in \mathcal{K}\, \text{Decrypt}_{\mathcal{E}}(\text{Encrypt}_{\mathcal{E}}(m, k), k) = m.$$
called *decryption function.*

Our requirements from functions $\text{Encrypt}_{\mathcal{E}}$ and $\text{Decrypt}_{\mathcal{E}}$ are: they both must be easy computable, in addition, for $\text{Decrypt}_{\mathcal{E}}$ must hold that

$$\forall m \in \mathcal{P}, k, k' \in \mathcal{K}, k \neq k'\, \text{Decrypt}_{\mathcal{E}}(\text{Encrypt}_{\mathcal{E}}(m, k), k') \neq m.$$

A *public (or asymmetric) cryptosystem* consists of the following parts:

1. A pair of keys $sk, pk \in \mathcal{K}$, we say that sk is the *secret key* and pk is the *public key.*
2. A function $\text{Encrypt}_{\mathcal{E}} : \mathcal{P} \times \mathcal{K} \to \mathcal{C}$ called *encryption function.*
3. A function $\text{Decrypt}_{\mathcal{E}} : \mathcal{C} \times \mathcal{K} \to \mathcal{P}$ with the additional property
$$\forall m \in \mathcal{P}, k \in \mathcal{K}\, \text{Decrypt}_{\mathcal{E}}(\text{Encrypt}_{\mathcal{E}}(m, k), k) = m.$$
called *decryption function.*

We require as before $\text{Encrypt}_{\mathcal{E}}$ and $\text{Decrypt}_{\mathcal{E}}$ to be easy computable, and also $\text{Decrypt}_{\mathcal{E}}$ must has the property to be infeasible for one who knows everything but the sk to find pk' such that

$$\forall m \in \mathcal{P}, sk, pk' \in \mathcal{K}\, \text{Decrypt}_{\mathcal{E}}(\text{Encrypt}_{\mathcal{E}}(m, sk), pk') = m.$$

In both symmetric and asymmetric ciphers the functions of encryption and decryption are parameterized by the length of the key which we call *security parameter* and denote by λ. We also assume that an adversary $\mathcal{A}$ may access every part of a cryptosystem except its private key. It may appears like we are giving to much power to $\mathcal{A}$, but all these features that we provide him are those that being exposed in public during every single execution of the scheme. That's the reason that we assume that $\mathcal{A}$ knows everything except the secret keys.

Recall now that out purpose is to protect every transmission from being captured by $\mathcal{A}$. Once we have instantiated the encryption-decryption functions we say that we have a cryptographic scheme and we denote it by $\mathcal{E}$. Afterwards, we have to convince Alice and Bob that scheme $\mathcal{E}$ is secure enough to be used by them. But what is a secure cryptographic scheme? One can give many definitions in such a question, but through all these years there is one that predominates over every other definition.

For a public key cryptographic scheme $\mathcal{E}$ we define the following game:

1. A probabilistic polynomial time-bounded adversary $\mathcal{A}$ is given a public key, which it may use to generate any number of ciphertexts (within polynomial bounds).

2. $\mathcal{A}$ generates two equal-length messages m_0 and m_1, and transmits them to an independent, honest coin-flipping machine $\mathcal{CF}$ along with the public key.

3. $\mathcal{CF}$ randomly a bit $b \in \{0\} \cup \{1\}$, encrypts the message m_b under the public key, and returns the resulting ciphertext c to $\mathcal{A}$.

4. Ask from $\mathcal{A}$ to find the value of b. Denote by S the event that $\mathcal{A}$ guesses the right b.

Definition 46. *The sceme is said to be* indistinguishable under chosen plaintext attacks (IND-CPA) *if for every possible $\mathcal{A}$ holds that:*

$$P(S) \leq \frac{1}{2} + negl(\lambda)$$

$negl(\lambda)$ is a sufficient small real number depended on the security parameter λ.

Occasionally, we call every IND-CPA secure public scheme a *semantic secure* scheme.

Part II

Homomorphic Cryptography

Chapter **II**

Motivation for the Problem

IMAGINE that you are facing the following situation: you have a huge amount of digital data and you also have a personal computer with a much smaller capacity than your data size. Suppose also, there is a server that you, among others, can connect with. This server is able to store your personal data and give you back at any time every part of them that you need. You decide to send it your data but you don't trust the server and you choose to encrypt your data in order to be infeasible for an adversary to read anything from them. Thereupon, as the server holds your encrypted data its time for you to ask it give you a part of them back. And here comes the first dilemma, how this can be done? A first option is; the server sends you back the data in packages, you encrypt them one by one and hold whatever you need. Another option is to trust the server and give it your private key to encrypt your data in order to avoid the previous time and memory expensive task.

Both solutions are unconvincing since we neither want to spend so much time in downloading, bit by bit, what we need nor want to let an unknown server operate over our data by giving it the secret key. Here comes the main idea that is behind homomorphic encryption. What if one can make queries over the encrypted data that the server has and take back only the part of the encrypted data that he needs? This question was first transpired just after the publication of the RSA cryptosystem [9] which is a multiplicative homomorphic scheme. For the above story this is what we all need in order to have our data stored secretly and at any time we want we can derive every part of them back efficiently. For cryptography though, it had been one of the most knotty problems ever. Since the problem of constructing, or even proving the existence, of such a scheme had been pending open over 30 years. But in year 2009 was Graig Gentry in his breakthrough paper [1] the first to construct a fully homomorphic scheme. Just after the first construction, which uses lattices as the mathematical structure, came the [2] which simplifies the first scheme using only integers.

In this thesis we present some, classical, partial homomorphic schemes, but the main part of this turns around the fully homomorphic schemes that we know so far. For completeness we present the fundamentals of the theory of integer lattices and in particular of ideal lattices that are the core for the one construction.

II.I Defining our goal

We have already define what a cryptographic scheme is and the parts from which it is formed. We now take a second look at each procedure and also we present one additional algorithm that is needed only by homomorphic schemes. The last is the algorithm that operates over the encrypted data after being parameterized by the circuit that one want to operate.

From now on we consider a public key cryptosystem $\mathcal{E}$ which consists of four algorithms: KeyGen$_{\mathcal{E}}$, Encrypt$_{\mathcal{E}}$, Decrypt$_{\mathcal{E}}$ and Evaluate$_{\mathcal{E}}$.

- The KeyGen$_{\mathcal{E}}$ algorithm takes as a parameter a variable, which we call the security parameter of the system and we will denote by λ.This algorithm outputs a pair (sk, pk) which are the system's secret and the public key respectively. Based on pk we define the space of the plaintexts $\mathcal{P}$ and the space of the chiphertexts $\mathcal{C}$ for the system $\mathcal{E}$.

- The second algorithm Encrypt$_{\mathcal{E}}$ first of all is parameterized by the public key of $\mathcal{E}$, its input is a plaintext $\pi \in \mathcal{P}$ and output the corresponding chiphertext $\psi \in \mathcal{C}$.

- The algorithm Decrypt$_{\mathcal{E}}$ is also parameterized by the private key of $\mathcal{E}$, it receives a chiphertext ψ and outputs its corresponding plaintext π.

- Finally every homomorphic cryptographic system has to be supplied with one more algorithm witch is going to be parameterized by the system's public key and its inputs will be a circuit $C \in \mathcal{C}_{\mathcal{E}}$, with $\mathcal{C}_{\mathcal{E}}$ we denote the set of circuits for witch the $\mathcal{E}$ is homomorphic, and a tuple of chiphertexts $\Psi = \langle \psi_1, \psi_2, ...\psi_t \rangle$ every element of Ψ is an input in the corresponding wire of the circuit. What we ask from Evaluate$_{\mathcal{E}}$ is: if, Encrypt$_{\mathcal{E}}(pk, \pi_i) = \psi_i$, for $1 \leq i \leq t$ and $C(\pi_1, \pi_2, ..., \pi_t) = w$, then, if Evaluate$_{\mathcal{E}}(pk, C, \Psi) = c$, the following holds: Decrypt$_{\mathcal{E}}(sk, c) = w$.

Now one can observe that a homomorphic scheme is exactly as a classic public scheme plus the evaluation algorithm. Our requirements from Evaluate$_{\mathcal{E}}$ is to be computed without secret key, all we need is somehow return back the encrypted data that we ask for.

Now that we have made every part of the scheme clear its time to define some properties that is necessary for a homomorphic scheme to have. Naturally, some of them, are exactly the same conditions that we require from a public key scheme. In our case tough, there is a need to restrict the scheme's behavior accordingly to its homomorphic aspects.

Definition 47 (Correctness of a homomorphic encryption scheme)**.** *We say than $\mathcal{E}$ is correct for circuits in $C \in \mathcal{C}_{\mathcal{E}}$ if:*

$$\text{Encrypt}_{\mathcal{E}}(pk, \pi_i) = \psi_i, \textit{ for } 1 \leq i \leq t \textit{ and } C(\pi_1, \pi_2, ..., \pi_t) = w$$

$$\textit{implies},$$

$$\textit{if } \text{Evaluate}_{\mathcal{E}}(pk, C, \Psi) = c \Rightarrow \text{Decrypt}_{\mathcal{E}}(sk, c) = w.$$

Definition 48 (Compacteness for a homomorphic encryption scheme)**.** *We call the cryptosystem $\mathcal{E}$ compact if there is a polynomial f so that the size of the decryption circuit $D_{\mathcal{E}}$ remains less than $f(\lambda)$ for every value of the security parameter λ.*

We say that a homomorphic encryption scheme *compactly evaluates* circuits in $\mathcal{C}_{\mathcal{E}}$ if it is compact and also correct for the circuits of $\mathcal{C}_{\mathcal{E}}$.

The two definitions above provide us a sense of the problem to be solved. The first requirement (correctness) is fully anticipated as it asks the apparent property of computing correctly over encrypting data in order the decrypter to proceed with the desirable result. The second requirement maybe is not clear at first, but remember what we don't need. A system that evaluates by sending all the data back to the decrypter in order to decrypt and then apply his circuit on them, is for sure correct, but is useless. Essentially, this is what we want to avoid, so, we forbid it by requiring compactness. So we exclude the trivial solution $\text{Evaluate}_{\mathcal{E}}(pk, C, \Psi) = (C, \Psi)$.

In the next chapters we encounter cryptosystems which have the property of correct evaluating circuits only upto a specific depth. For simplification we denote them by $\mathcal{E}^i$. That is, if $\mathcal{E}$ is a cryptosystem that correctly evaluates circuits of depth at most i, $i \in \mathbb{Z}$, then we denote this cryptosystem with $\mathcal{E}^i$.

In response to the above, there must take place a couple of, modified, definitions in order to conventionally continue by the right terminology.

Definition 49 (Leveled fully homomorphic encryption)**.** *Let $\{\mathcal{E}^{(d)} : d \in \mathbb{N}\}$ be a family of homomorphic encryption schemes, we say that $\mathcal{E}^{(d)}$ is leveled fully homomorphic if for every $d \in \mathbb{N}$ all possible decryptions are computed by the same decryption circuit and also $\mathcal{E}^{(d)}$ compactly evaluates every circuit of depth at most d, moreover, the complexity of every algorithm that contains $\mathcal{E}^{(d)}$ is polynomial according to λ, d and to the size of the circuit C.*

Definition 50 (Fully homomorphic encryption scheme)**.** *We call a scheme $\mathcal{E}$ fully homomorphic if it compactly evaluates every possible circuit.*

All we did is that we named our main target. That is, we are working towards constructing a fully homomorphic scheme.

II.II Applications of Homomorphic Cryptography

The first example than demonstrates the usefulness of fully homomorphic encryption has already been presented. Even though the web runs faster and faster and points to the direction of using any computer for data storing it is not enough because who wants to store his private data in an untrusted server? The development of a fully homomorphic scheme is the first result that strengthens the idea of secure data storing on different data banks. This means that one is able to access his data from anywhere since he can store everything in an unknown server.

One more useful application that homomorphisms have to offer is that they open the way for building e-voting systems. In a case that a fully homomorphic scheme is available then every one could deposit his vote after encrypting it. Then the system is capable of counting all the votes and get the result without knowing the individual choices. An extension of e-voting is the statistical analysis that could take place in a system which gives questionnaires in a special form so that it collects only anonymous and encrypted data that can be processed only by the system.

Internet queries also could be more ”secure” after the existence of such a scheme. Since its possible to encrypt the search query and the appropriate server could be send the results of the query without knowing what has really been asked.

Chapter ***III***

Partially Homomorphic Schemes

IT took over 30 years for the researchers to come to a conclusion for the existence of a homomorphic scheme. In the meanwhile, an increasingly number of partial homomorphic schemes was developed. By the term *partial homomorphic schemes* we refer to cryptosystems which don't have the homomorphic property for every circuit but the have it for at least one circuit operation. Next, we briefly present some previous cryptographic schemes for which the homomorphic property holds at least for one of the operations of addition or for multiplication. We first describe RSA, which is multiplicative homomorphic but this version of the algorithm does not confer semantic security. Next we have ElGamal scheme which is semantic secure and is multiplicative homomorphic. The last scheme we present is the Goldwasser-Micali cryptosystem, a scheme that uses quadratic residues and evaluates homomorphically the exclusive-or operation.

III.I RSA

In 1978 Rivest, Shamir, Adleman designed the well-known public key cryptosystem RSA ([9]) named after its inventors, which is the scheme that rendered the motivation for the search of a fully homomorphic scheme.

First we describe the functions $KeyGen_{RSA}$, $Encrypt_{RSA}$ and $Decrypt_{RSA}$, that characterize the scheme:

Scheme 1 (RSA)**.**

KeyGen$_{\text{RSA}}$

1. *Choose two distinct prime numbers* p, q .
2. *Compute* $n = p \cdot q$.
3. *Compute Euler's totient function* $\phi(p \cdot q) = (p-1) \cdot (q-1)$.
4. *Choose an integer* e *such that* $1 < e < \phi(p \cdot q)$, *and*

$$gcd(e, \phi(n)) = 1.$$

5. *Determine* $d = e^{-1} \mod (p-1) \cdot (q-1)$.
6. *Return public key* $\{n, e\}$, *private key* $\{d\}$.

Encrypt$_{\text{RSA}}$

1. *Take as input a message* $m \in \mathcal{P}$.
2. *Return* $c = m^e \mod n$.

Decrypt$_{\text{RSA}}$

1. *Take as input a ciphertext* c.
2. *Return* $m = c^d \mod n$.

Observe now that if $pk = \{n, e\}$, $sk = \{d\}$ are the public and the private keys respectively. Then for messages $m_1, m_2 \in \mathcal{P}$ let $c_1 = \text{Encrypt}_{\text{RSA}}(pk, m_1) = m_1^e \mod n$ and $c_2 = \text{Encrypt}_{\text{RSA}}(pk, m_2) = m_2^e \mod n$, then $c_1 \cdot c_2 \mod n = m_1^e \cdot m_2^e \mod n = (m_1 \cdot m_2)^e \mod n$. That is, we proved that:

$$\text{Encrypt}_{\text{RSA}}(pk, m_1) \cdot \text{Encrypt}_{\text{RSA}}(pk, m_2) = \text{Encrypt}_{\text{RSA}}(pk, m_1 \cdot m_2).$$

Thus, we have shown that RSA is a multiplicative homomorphic scheme. RSA was the first partially homomorphic scheme ever and one can say that the initiation of homomorphic encryption is due to this classic scheme.

III.II Goldwasser-Micali Cryptosystem

The first scheme ever with a proof of semantic security was purposed by Shafi Goldwasser and Silvio Micali in 1982. They were the first to introduce the notion of semantic security, and their scheme was the first to use probabilistic method for the encryption.

Next we denote by $\left(\frac{x}{p}\right)$ the Jacobi symbol.

Scheme 2 (Goldwasser-Micali)**.**

KeyGen$_{\mathrm{G-M}}$

1. *Generate two prime numbers* p, q.
2. *Set* $N = p \cdot q$.
3. *Compute in integer* x *such that* $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1$.
4. *Return public key* $= \{N, x\}$, *private key* $= \{p, q\}$.

Encryption$_{\mathrm{G-M}}$

1. *Choose a random* $y \in \{0, 1, ..., N-1\}$.
2. *Calculate* $c_i = y^2 \cdot x^{m_i} \mod N$.
3. *Return* $\{c_1, ..., c_n\}$.

Decryption$_{\mathrm{G-M}}$

1. *For every* c_i *check if* c_i *is a quadratic residue.*
2. *If* c_i *is quadratic residue then* $m_i = 0$ *else* $m_i = 1$.

The Goldwasser-Micali scheme offers homomorphic evaluation over the operation $\oplus$ (i.e. exclusive-or). This can be proved if we consider two random bits b_1, b_2 encrypted over the cryptosystem as c_1, c_2 respectively. Then $\mathrm{Encrypt}_{\mathrm{G-M}}(b_1) \cdot \mathrm{Encrypt}_{\mathrm{G-M}}(b_2) = y_1^2 \cdot x^{b_1} \cdot y_2^2 \cdot x^{b_2} \mod N = (y_1 \cdot y_2)^2 \cdot x^{b_1+b_2} \mod N$.

III.III ElGammal

Seven years after the publication of the RSA cryptographic scheme, a new partially homomorphic scheme has occurred. Taher ElGammal built a scheme inspired by the classic key exchange protocol Diffie-Hellman. ElGamal is a non-deterministic encryption scheme, one plaintext has multiple ciphertext representations and that is why this is a semantically secure cryptosystem. Details about the algorithms that the scheme uses provided below.

Scheme 3 (ElGammal)**.**

KeyGen$_{\mathrm{ElG}}$

1. *Generate a multiplicative cyclic group* G *of order* q *with generator* g.

2. *Choose a random* $w \in \{0, 1, ..., q-1\}$.
3. *Compute* $h = g^w$.
4. *Return public key* $= \{G, q, g, h\}$, *private key* $= \{w\}$.

Encryption$_{\mathrm{ElG}}$

1. *Choose a random* $y \in \{0, 1, ..., q-1\}$ *and calculate* $c_1 = g^y$.
2. *Calculate* $c_1 = g^y$, $s = h^y$, $c_2 = m \cdot s$.
3. *Return* $\{c_1, c_2\}$.

Decryption$_{\mathrm{ElG}}$

1. *Calculate* $s = c_1^w$.
2. *Return* $m = c_2 \cdot s^{-1}$.

Consider now plaintexts m_1, m_2 then their corresponding ciphertexts: $\{g^{y_1}, m_1 \cdot h^{y_1}\}$ and $\{g^{y_2}, m_2 \cdot h^{y_2}\}$ respectively. Let's multiply now the components of the ciphertexts, we get $\{g^{y_1+y_2}, m_1 \cdot m_2 \cdot h^{y_1+y_2}\}$. For the decryption now we have $s = (g^{y_1+y_2})^w = h^{y_1} \cdot h^{y_2}$ which gives that $m = m_1 \cdot m_2 \cdot h^{y_1+y_2} \cdot (h^{y_1} \cdot h^{y_2})^{-1} = m_1 \cdot m_2$. Which implies:

$$\mathrm{Encrypt}_{\mathrm{ElG}}(pk_1, m_1) \cdot \mathrm{Encrypt}_{\mathrm{ElG}}(pk_2, m_2) = \mathrm{Encrypt}_{\mathrm{ElG}}(pk_1 + pk_2, m_1 \cdot m_2).$$

As in the case of RSA we have show that ELGammal is multiplicative homomorphic, but in this case the final ciphertext is encrypted neither first nor second public key, but it's encrypted under the sum of the public keys.

Chapter **IV**

Bootstrappable encryption

WITHIN this chapter we focus on a special property of a cryptosystem $\mathcal{E}$ which we call *bootstrappability.* This is a universal property and becomes interesting when the scheme is a leveled homomorphic scheme. Then *botstrappability* permits one to achieve fully homomorphic encryption only by modifying the existing leveled homomorphic scheme. Though, in order to efficient construct such a fully homomorphic scheme we have to chose carefully the scheme that we will build on. A crucial property the initial scheme must have is a shallow decryption circuit.

IV.I Intuition for Bootstrappability

Thus far, there have been presented only public schemes which are homomorphic in a limited sense. In this chapter we describe a way of constructing a fully homomorphic scheme starting by one that is partially homomorphic. This is possible if the initial scheme has one additional property, to be able to homomorphic evaluate its own decryption circuit. This self referential characteristic may seems unusual on the ground to be used since the most common use of techniques that use self reference is to prove that the algorithm being examined has poor running times or even that a problem that can't be solved. In this case though, we prove that this property suffices to permit constructing a secure public homomorphic scheme.

Before the formal description of our requirements and of the way that we can build such a scheme we give an example to improve our intuition with regard to the notion of bootstrappability.

First assume that we have a public key cryptosystem $\mathcal{E}$ which has the property of being able to homomorphically evaluate its decryption circuit $\text{Decrypt}_{\mathcal{E}}$. Imagine now that Alice holds a message m, and encrypts it as c_A under her public key sk_A, we can build a strategy for converting the ciphertext c_A to another ciphertext c_B which encrypts the message m but under Bob's public key sk_B.

First, Alice publishes her secret key but encrypted under Bob's public key pk_B, let $\overline{sk_A}$ the result of this procedure. After this Bob has sk_B, pk_B, $\overline{sk_A}$, and c_A. Now he can take advantage of the homomorphic evaluation that the scheme offers for $\text{Decrypt}_{\mathcal{E}}$ as follows: He encrypts c_A under his public key let $\overline{c_A}$ this, double, encrypted message. Then he asks for an evaluation on the circuit $\text{Evaluate}_{\mathcal{E}}(pk_B, \text{Decrypt}_{\mathcal{E}}, \overline{sk_A}, \overline{c_A})$. Last circuit's result is c_B, that is, m encrypted under Bob's public key, this happens only because of the special property that $\mathcal{E}$ by assumption has.

Now suppose that the homomorphic evaluation stands not only for the decryption circuit but also for the NAND gate. Moreover, lets assume that if we connect two copies of the $\text{Decrypt}_{\mathcal{E}}$ circuit via a NAND gate then the resulting circuit can also be homomorphically evaluated by the scheme, we denote this circuit by NAND_{DEC}^{DEC}. Then Alice and Bob can operate as above and begin with two messages m_1, m_2 encrypted under Alice's public key as c_{1A}, c_{2A} and finish with the message $m_1 \text{NAND} m_2$ encrypted under Bob's public key.

The main idea behind what follows is the above game as we will see that is the core for a technique that finally permits the homomorphic encryption under arbitrary circuits. The big issue is to find a complete set S of gates whose each member must be able to do what the NAND gate did above. We need for every gate $g \in S$ the resulting circuit after putting on its input wires copies of $\text{Decrypt}_{\mathcal{E}}$ to be evaluated homomorphically by the scheme. Then, the concept for making a scheme that can evaluate arbitrary circuits is that because of the completeness of S we can express every circuit as a combination of its elements. Thus, for a circuit C the first step is to express it as a combination of elements of S and then run the above game for each of the circuits that form C to conclude with a homomorphic evaluation on C which is formed by circuits that by assumption computed homomorphically by the scheme.

IV.II A generic construction

Continuing the above approach to the problem we formalize the steps and the requirements that permit bootstrappability. First, we give a definition of what an *augmented decryption circuit* is. This is, just for simplification for everything that follows.

Definition 51. *Let $D_{\mathcal{E}}$ be the decryption circuit of $\mathcal{E}$ and Γ a set of gates which includes the identity gate (is the one that outputs the input as it is). Also,let $g \in \Gamma$ we call a circuit consisting of g modified so that every input wire has a copy of $D_{\mathcal{E}}$ g-augmented decryption circuit. The set of all the g-augmented decryption circuits, $g \in \Gamma$, is denoted by $D_{\mathcal{E}}(\Gamma)$.*

We will see that if $\mathcal{C}_{\mathcal{E}}$ contains the augmented decryption circuit of $\mathcal{E}$ then we can construct from $\mathcal{E}$ a fully homomorphic scheme which evaluates every circuit. In fact the circuit $\mathrm{NAND}_{DEC}^{DEC}$ that we presented in previous section is an augmented decryption circuit. So the way of getting a fully homomorphic scheme starts to appear, it will be a generalized method that implements the strategy we described above.

Definition 52. *Let $\mathcal{E}$ be a leveled homomorphic scheme and Γ a set of gates. We say that $\mathcal{E}$ is bootstrappable with respect to Γ if $D_{\mathcal{E}}(\Gamma) \subseteq \mathcal{C}_{\mathcal{E}}$*

Towards working on finding a fully homomorphic scheme this property plays a lead role as we will see in next chapters. In breath, this gives us the opportunity to construct an algorithm which inputs will be a public key pk_1, a secret key encrypted under pk_1, $\overline{sk_2}$, and a ciphertext encrypted under the corresponding public key pk_2. Then the algorithm's output will be the ciphertext encrypted under pk_2. We will take advantage of it as we will construct a scheme than during the evaluation of a circuit the use of the above algorithm will refresh the ciphertext every time it comes to a new gate. An obvious usefulness of it is when we deal with encryptions that introduce errors in the plaintexts, in such occasions we need to keep the error small enough to be able to decrypt without errors which is a non trivial task since during the evaluation the error maybe grows in every gate the ciphertext is an input.

Next we describe the algorithm Recrypt that materializes all the above. But before we have to make the following clarifications: We consider two pairs of keys (pk_1, sk_1) and (pk_2, sk_2), both are outputs of $\mathrm{KeyGen}_{\mathcal{E}}(\lambda)$ algorithm, we also have $\mathrm{Encrypt}_{\mathcal{E}}(pk_1, \psi_1) = \pi$ and by $\overline{sk_{(1,j)}}$ we denote the j-th bit of the key sk_1 encrypted under pk_2 and by $\psi_{(1,j)}$ the j-th bit of ψ_1.

Algorithm 1.

```
1. $\overline{\psi_{(1,j)}} \xleftarrow{R} \text{Encrypt}_{\mathcal{E}}(pk_2, \psi_{(1,j)})$, for all $j \in [1, \ell(\psi_1)]$
2. $\psi_2 \leftarrow \text{Evaluate}_{\mathcal{E}}(pk_2, D_{\mathcal{E}}, \langle\langle\overline{sk_{(1,j)}}\rangle, \langle\overline{\psi_{(1,j)}}\rangle\rangle)$
3. return $\psi_2$
```

In order the algorithm to be more efficient we replace the second step with this: $\overline{\psi_{(1,j)}} \xleftarrow{R} \text{WeakEncrypt}_{\mathcal{E}}(pk_2, \psi_{(1,j)})$, for all $j \in [1, \ell(\psi_1)]$, where $\text{WeakEncrypt}_{\mathcal{E}}$ algorithm is a weaker encryption algorithm than $\text{Encrypt}_{\mathcal{E}}$ but its decryption circuit has less complexity. Although $\text{WeakEncrypt}_{\mathcal{E}}$ is weaker than $\text{Encrypt}_{\mathcal{E}}$ it does not hurt security because when we apply $\text{WeakEncrypt}_{\mathcal{E}}$ we have already an encrypted message.

Now, we show how the above algorithms and observations will help us construct a leveled homomorphic encryption scheme. Starting from a bootstrappable scheme $\mathcal{E}$ with respect to the gates of the set Γ we construct the scheme

$$\mathcal{E}^{(d)} = (\text{KeyGen}_{\mathcal{E}^{(d)}}, \text{Encrypt}_{\mathcal{E}^{(d)}}, \text{Evaluate}_{\mathcal{E}^{(\delta)}}, \text{Decrypt}_{\mathcal{E}^{(d)}})$$

which can evaluate homomorphically circuits of depth at most d. We denote by $D_{\mathcal{E}}(\Gamma, \delta)$ the circuits consisting of gates from the set Γ and their depth is δ augmented by $D_{\mathcal{E}}$. In details the algorithms of $\mathcal{E}^{(d)}$ are:

$\text{KeyGen}_{\mathcal{E}^{(d)}}(\lambda, d)$: sets:

$$sk_i \xleftarrow{R} \text{KeyGen}_{\mathcal{E}}(\lambda),\ i \in [0, d]$$

$$\overline{sk_{(i,j)}} \xleftarrow{R} \text{Encrypt}_{\mathcal{E}}(pk_{i-1}, sk_{(i,j)}),\ i \in [1, d], j \in [1, \ell(\lambda)]$$

where $sk_{(i,1)}, sk_{(i,2)}, ..., sk_{(i,\ell)}$ are the representations of sk_i as elements of the space $\mathcal{P}$. This algorithm outputs the secret key of the scheme $sk^{(d)} \leftarrow sk_0$ and also the public key $pk^{(d)} \leftarrow (\langle pk_i\rangle, \langle\overline{sk_{(i,j)}}\rangle)$ with $i \in [0, \delta]$ and for $\delta \leq d$.

$\text{Encrypt}_{\mathcal{E}^{(d)}}(pk, \pi)$: Its input is a $\pi \in \mathcal{P}$ and it output a ψ such that:

$$\psi \xleftarrow{R} \text{Encrypt}_{\mathcal{E}}(pk_d, \pi)$$

$\text{Decrypt}_{\mathcal{E}^{(d)}}(sk, \psi)$: With input a ciphertext ψ it computes and returns the $\text{Decrypt}_{\mathcal{E}}^{sk_0}(\psi)$.

Augment$_{\mathcal{E}^{(\delta)}}(pk^{(\delta)}, C_\delta, \Psi_\delta)$**:** The input here is a circuit C_δ with depth at most δ with gates from the set Γ and an encrypted input of it Ψ_δ. After, it augments the circuit C_δ by $D_{\mathcal{E}}$, we denote the yielding circuit by $C^\dagger_{\delta-1}$. The next move is to construct an input for the new circuit. So, every $\psi \in \Psi_\delta$ is being replaced by the tuple $\langle\langle\overline{sk_{(\delta,j)}}\rangle, \langle\overline{\psi_j}\rangle\rangle$, where $\overline{\psi_j} \leftarrow \text{WeakEncrypt}_{\mathcal{E}^{(\delta-1)}}(pk^{(\delta-1)}, \psi_j)$ but every ψ_j comes from the representation of ψ as an element of $\mathcal{P}$. Finally, the output is the pair $(C^\dagger_{\delta-1}, \Psi^\dagger_{\delta-1})$

Reduce$_{\mathcal{E}^{(\delta)}}(pk^{(\delta)}, C_\delta, \Psi_\delta)$**:** The input is $C^\dagger_\delta \in D_{\mathcal{E}}(\Gamma, \delta+1)$ and a proper input for $C^\dagger_\delta$, let $\Psi^\dagger_\delta$. It constructs a new circuit, C_δ, which is a subcircuit of $C^\dagger_\delta$ consisting only from its δ first levels. It also produces an input for the new circuit. It comes from the $\Psi^\dagger_\delta$ and we denote it by Ψ_δ. The element of Ψ_δ which corresponds to the wire w of the new circuit will be the Evaluate$_{\mathcal{E}}(pk_\delta, C^{(w)}_\delta, \Psi^{(w)}_\delta)$, with $C^{(w)}_\delta$ to be the subcircuit of $C^\dagger_\delta$ whose output is the wire w and $\Psi^{(w)}_\delta$ is the input of $C^{(w)}_\delta$).

Algorithm outputs the pair (C_δ, Ψ_δ).

Evaluate$_{\mathcal{E}^{(\delta)}}(pk^{(\delta)}, C_\delta, \Psi_\delta)$**:** The input is a circuit C_δ of depth at most δ consisting of gates taken by Γ and a tuple of ciphertexts (under pk_δ) for its input, Ψ_δ. We note here that because this algorithm runs recursively, we assume without loss of generality that all the wires of C_δ are connecting gates of consecutive levels of the circuit. If the above does not hold we add identity gates to where needed. The algorithm then operates as follows:

1. If the depth of C_δ equals to zero, then it returns Ψ_0
2. $(C^\dagger_{\delta-1}, \Psi^\dagger_{\delta-1}) \leftarrow \text{Augment}_{\mathcal{E}^{(\delta)}}(pk^{(\delta)}, C_\delta, \Psi_\delta)$
3. $(C_{\delta-1}, \Psi_{\delta-1}) \leftarrow \text{Reduce}_{\mathcal{E}^{(\delta-1)}}(pk^{(\delta-1)}, C^\dagger_{\delta-1}, \Psi^\dagger_{\delta-1})$
4. Simulates Evaluate$_{\mathcal{E}^{(\delta-1)}}(pk^{(\delta-1)}, C_{\delta-1}, \Psi_{\delta-1})$

The above construction points to a new direction for searching a fully homomorphic scheme, maybe we can start building one step by step than only trying to find an ad-hoc algorithm that operates homomorphically for every circuit. All we need is a bootstrappable scheme with a swallow decryption circuit in order to fit it into the scheme's homomorphic operations circuits and, operating as above, finally end with a fully homomorphic scheme.

IV.II.I Correctness, complexity and security of the scheme

Theorem 17. *Correctness*
Let $\mathcal{E}$ be a bootstrapable scheme with respect to the set of gates Γ. Then, $\mathcal{E}^{(d)}$ compactly evaluates every circuit of depth at most d with gates from Γ.

Proof. We denote by $D(d, w, C, \Psi)$ the value of the wire w after the decryptions of Ψ as inputs of C. Since initially we have the circuit C_d with input the ciphertext Ψ_d it suffices to show that $D(d, w_{out}, C_d, \Psi_d) = D(0, w_{out}, C_0, \Psi_0)$ for every wire w_{out} of the output, namely, of the circuit C_0. The correctness of the above comes from the following two conditions:

- $(C^\dagger_{\delta-1}, \Psi^\dagger_{\delta-1}) \leftarrow \text{Augment}_{\mathcal{E}^{(\delta)}}(pk^{(\delta)}, C_\delta, \Psi_\delta)$, so we have:
  $D(\delta, w, C_\delta, \Psi_\delta) = D(\delta - 1, w, C^\dagger_{\delta-1}, \Psi^\dagger_{\delta-1})$

- $(C_\delta, \Psi_\delta) \leftarrow \text{Reduce}_{\mathcal{E}^{(\delta)}}(pk^{(\delta)}, C^\dagger_\delta, \Psi^\dagger_\delta)$, which gives:
  $D(\delta, w, C^\dagger_\delta, \Psi^\dagger_\delta) = D(\delta, w, C_\delta, \Psi_\delta)$

□

Note here that if Γ is an universal set of gates, then the family $\mathcal{E}^{(d)}$ is leveled fully homomorphic.

Theorem 18. *Complexity For a circuit C of depth at most d and of size s (the total number of the wires) we need at most $s \cdot \ell \cdot d$ calls of* $\text{WeakEncrypt}_\mathcal{E}$ *and $s \cdot d$ calls of* $\text{Evaluate}_\mathcal{E}$ *at the circuits of $D_\mathcal{E}(\Gamma)$, during the operation of* $\text{Evaluate}_{\mathcal{E}^{(d)}}$ *for C.*

Proof. First of all, observe that our acceptance that $\text{Evaluate}_{\mathcal{E}^{(d)}}$ handles only circuits whose wires connect gates in consecutive levels maybe alters C in order to be a valid circuit. The worst case for this modifications gives us a circuit which has at most $s \cdot d$ wires. But for each of them because of the algorithm $\text{Evaluate}_{\mathcal{E}^{(d)}}$ and because of the augmentation we have calls of $\text{WeakEncrypt}_\mathcal{E}$. If $\ell(\psi) = m$, then for the expression of ψ as a concatenation of length ℓ we need ℓ elements (they are the new ψ_j) which means that for every wire we have ℓ additional calls of $\text{WeakEncrypt}_\mathcal{E}$. All the above gives us that $\text{WeakEncrypt}_\mathcal{E}$ may be called at most $s \cdot d \cdot \ell$ times.
For the second part of the theorem it suffices to observe that $\text{Evaluate}_{\mathcal{E}^{(d)}}$ is being called only from the $\text{Reduce}^{pk^{(\delta)}}_{\mathcal{E}^{(\delta)}}(C_\delta, \Psi_\delta)$ and it happens only once for each wire. □

Theorem 19. *Security (chosen plaintexts attacks)*
Let $\mathcal{A}$ be an algorithm that in time t with advantage ϵ brakes the semantic

security of $\mathcal{E}^{(d)}$, then there is an algorithm $\mathcal{B}$ which in time $t \cdot \ell$ and with advantage at least $\epsilon/(\ell(d+1))$ breaks the semantic security of $\mathcal{E}$.

Proof. From so on, for $k \in [0, d]$, we denote by *Game* k the game for the definition of the semantic security of $\mathcal{E}^{(d)}$. With the variation that the challenger defines $(sk'_i, pk'_i) \xleftarrow{R} \text{KeyGen}_{\mathcal{E}}(\lambda)$ and $sk_{(i,j)} \xleftarrow{R} \text{Encrypt}_{\mathcal{E}}^{pk_{i-1}}(sk'_{(i,j)})$. We also define the *Game* $d+1$ being exactly the same with *Game* d with the only difference that the challenger ignores the challenges π_0, π_1 and encrypts a new, random, message π. We define with ϵ_k the advantage of the adversary at *Game* k. First we can observe that in the *Game* 0 the adversary has by assumption advantage $\epsilon_0 = \epsilon$, and also $\epsilon_{d+1} = 0$ because we have fool the adversary giving him a forged message. So, in the sequence of the games starting at *Game* 0 until *Game* $d+1$ there must be to consecutive games $k, k+1$, with $k \in [0, d]$ for which holds that $|\epsilon_k - \epsilon_{k+1}| \geq \epsilon/(d+1)$. From now on we fix this k.

Next we construct an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ in order to break $\mathcal{E}^{(\ell)}$ which is by definition to be equal to the $\mathcal{E}$ but with plaintext space the set $\mathcal{P}^{\leq \ell}$. We denote by $\mathcal{C}_B$ the challenger in the game that $\mathcal{B}$ aims to break $\mathcal{E}^{(\ell)}$. Obviously in the game where the adversary is $\mathcal{A}$ the challenger is $\mathcal{B}$. We now give the operation of $\mathcal{B}$: when he receives the public key pk from $\mathcal{C}_B$ he produces keys for the *Game* k with the algorithms that we have just define but after this he replaces the value of pk_k with pk. He also produces one more pair $(sk'_{k+1}, pk'_{k+1}) \xleftarrow{R} \text{KeyGen}_{\mathcal{E}}(\lambda)$ and sends $\mathcal{C}_B$ the $\pi_0 = sk_{k+1}$ and $\pi_1 = sk'_{k+1}$. After this $\mathcal{C}_B$ chooses $b_1 \xleftarrow{R} \{0, 1\}$ and sends $\mathcal{B}$ the tuple of ciphertexts $\langle \psi_j \rangle$ which are the encryptions for the bits of $\langle \pi_{b_1,j} \rangle$. $\mathcal{B}$'s next move is to replace $\langle \overline{sk_{(k+1,j)}} \rangle$ with $\langle \psi_j \rangle$. After this we have that the values that $\mathcal{B}$ gives to $\mathcal{A}$ correspond to the *Game* $k + b_1$. $\mathcal{A}$ produces two messages π_0, π_1 and sends them to $\mathcal{B}$. For this move of $\mathcal{B}$ there are two possibilities.

1. If $k < d$, then chooses $b_2 \xleftarrow{R} \{0, 1\}$ and sends $\mathcal{A}$ the tuple consisting of the values $\psi_j \leftarrow \text{Encrypt}_{\mathcal{E}}^{pk_d}(\pi_{(b_2,j)})$.

2. If $k = d$, then constructs a random π and sends $\mathcal{C}_B$ the π_{b_2} and π after letting $\mathcal{C}_B$ to choose randomly one of them he encrypts it and after he sends $\mathcal{B}$ the corresponding ciphertext. Then $\mathcal{B}$ gives it to $\mathcal{A}$ whose answer is a $b_3 \in \{0, 1\}$. Finally, $\mathcal{A}$ answers $b_3 \oplus b_2$ to $\mathcal{C}_B$

In both occasions we have construct either the *Game* k or the *Game* $k+1$ which by hypothesis $\mathcal{A}$'s advantage is $\epsilon/(d+1)$ to distinguish between them, so this distribution passes to $\mathcal{B}$ and he is able to distinguish the respecting ciphertext. So we have proved that $\mathcal{B}$ breaks the semantic security of $\mathcal{P}^{\leq \ell}$ with advantage $\epsilon/(d+1)$. On the other hand to attack $\mathcal{E}$ we need ℓ calls of $\mathcal{E}^{\ell}$ so with probability at least $1/\ell$ there is one call where $\mathcal{A}$ distinguishes his starting messages. Namely, $\mathcal{B}$ runs an amount of $t \cdot \ell$ steps and has advantage at least $\epsilon/(\ell(d+1))$. $\square$

IV.III Fully homomorphic scheme

We have describe the way to construct a leveled homomorphic scheme, since the key of $\mathcal{E}^{(d)}$ depends on the depth, d, of the circuit that we want to evaluate. So, what we need to modify the above construction in a fully homomorphic scheme is to fix the length of the key. We now give a useful security definition:

We say that a cryptosystem is secure for key dependent attacks (KDM secure) if, for known public keys $pk_1, pk_2, ..., pk_n$ every encryption of the function $f(sk_1, sk_2, ..., sk_n)$, f is chosen by the adversary, does not reveal anything about any of the private keys sk_i.

Now, we show that if $\mathcal{E}$ is secure under key dependent attacks then we can modify this scheme in the following way without breaking its semantic security that its already proved: Instead of constructing a chain of d keys, where d is the depth of the circuit, we can construct only one pair $(sk^*, pk^*) \overset{R}{\longleftarrow} \text{KeyGen}_{\mathcal{E}}(\lambda)$ and every time algorithm Recrypt operates we refresh the encryption with the same key, we denote the new scheme by $\mathcal{E}^*$. Then, because the scheme is secure under key dependent attacks we can prove that $\mathcal{E}^*$ is semantically secure. The proof is exactly the same with the case that after every refresh of the keys we did not have same keys.

Chapter V

Introduction to Lattices

In this chapter we present the notion of a lattice. They have a key role in the construction of the homomorphic scheme and that's because they provide encryption schemes with low decryption complexity circuits. To see why this is important in our construction recall the introductory scheme and especially the augmentations that it produces. This is an operation that lead us to a scheme which executes a huge number of decryptions, so the need of a low complexity circuit for the decryption is necessary.

V.I Lattices

Here we give the definition of a lattice. We denote the Euclidean length of vector $x = (x_1, x_2, ..., x_n)$ by $\|x\|$ that is, $\|x\| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$. Also for a square matrix B we denote by $\|B\|$ the maximum length of its column vectors.

Definition 53. *Let B be an integer matrix of dimensions $m \times n$, we take the set $\mathcal{L} = \{y : y = B \cdot x \quad \forall x \in \mathbb{Z}^{1\times n}\}$, that is the set of all linear combinations of B. We call every $\mathcal{L}$ with the above properties a lattice.*

Lattices are being studied since early 80's and so far there are some problems associated with them that seem to be hard to solve. For some of them there are no known algorithms even for quantum computers. The above give us the opportunity to think of them as a power tool to be used in cryptography. We start to study in detail some properties of lattices and after we present some of the most famous lattice problems.

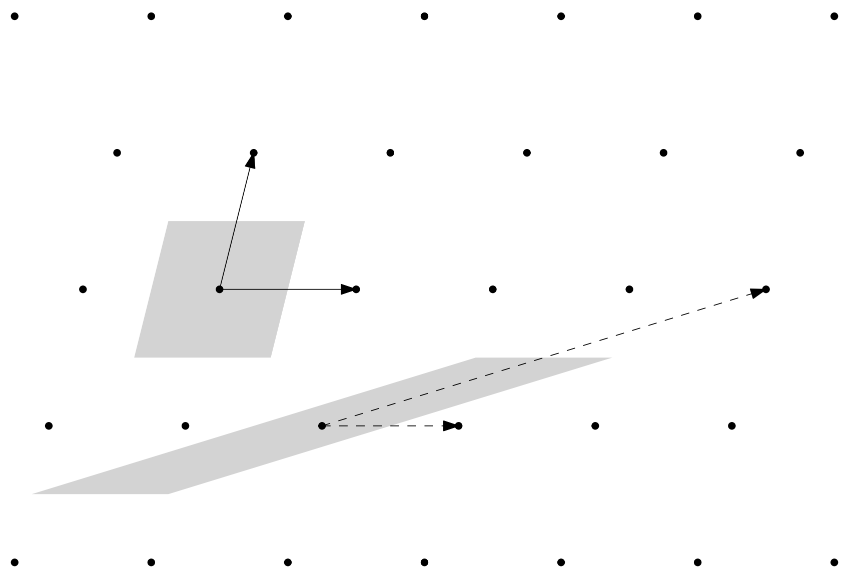
Figure V.I: A lattice and two possible bases.

First of all one must notice that a lattice is fully characterized by the matrix that generates it. In fact, matrix B contains n linear independent vectors which are the basis of the lattice. Despite, a lattice $\mathcal{L}$ does not produced only by one, unique, basis, we will show that there are to many different bases that produce the same lattice. Towards this goal we first define the notion of unimodular matrix.

Definition 54. *Let $U \in \mathbb{Z}^{n \times n}$, with the property that $|\det(U)| = 1$,then we say that B is unimodular.*

Lemma 1. *Let U is a unimodular matrix then it has inverse, the matrix denoted U^{-1}, and also this matrix is unimodular.*

Proof. We have that $det(U^{-1}) = \frac{1}{det(U)} = \pm 1$. We need to show that U^{-1} is an integer matrix. The last is true since along the Cramer's rule the (i, j) position of matrix U^{-1} has the element $\frac{(-1)^{i+j} \cdot \det(U_{(i,j)})}{\det(U)}$, where $U_{(i,j)}$ denotes the matrix U with its i-th line and j-th column deleted. □

Theorem 20. *The set of unimodular matrices of dimensions $n \times n$ is a group under matrix multiplication.*

Proof. The proof is an imidiate fact of the previous theorem and the multiplicative property of the determinant $(det(A) \cdot det(B) = det(A \cdot B))$. □

Before next theorem we start by giving a tip that helps us in next proof and also gives a better view of what a unimodular matrix is.

Proposition 1. *Suppose B is a square matrix. Then all the following operations can be performed on B only by a right-multiplication with an appropriate unimodular matrix.*

1. *Multiplication of a column by -1.*
2. *Exchange of two columns.*
3. *Addition of an integer multiple of one column to another.*

Proof. 1. Take the matrix

$$U_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

U_1 is the identity matrix except the (i,i) position where there is an -1, if we apply it with right-multiplication to a square matrix A then the resulting matrix will be the same as A but with its i-th column multiplied by -1.

2. If we have to exchange two columns then we take the following matrix.

$$U_2 = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

U_2 is almost the same as the identity matrix but in its i-th line has the 1 in the j-th position and also in the j-th li ne has its 1 in the i-th position. As in first case if we apply it with right-multiplication to a square matrix A then the resulting matrix will be the same as A but with its i-th and j-th columns exchanged.

3. In case we need to add an integer multiple of a column to another we consider the matrix U_3.

$$U_3 = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & k & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

U_3 is almost the identity matrix but it has an additional k in the position (i, j), then with right-multiplication to a square matrix A the result will be a matrix with every column is identical to corresponding A's column except the i-th column which will be the sum of the i-th and k times j-th column of A.

□

Now we prove a property that gives us a way to construct a new basis for a lattice when we already have one.

Lemma 2. *Suppose B_1 is a basis for a lattice $\mathcal{L}$, then B_2 is a matrix for the same lattice if and and only if there exists a unimodular matrix U such that $B_2 = B_1 \cdot U$*

Proof. $(\Rightarrow)$

Suppose that B_1 and B_2 define the same lattice. Then every column of B_2 is a linear combination of the columns of B_1. So, the first column of matrix B_2 is a linear combination of those of B_1, according to previous lemma there is a unimodular matrix U_1 so that the product $B_1 \cdot U_1$ is the same as B_1 but with its first column being identified to the first column of B_2. Thus the same must hold for the second column of B_2, there is a unimodular matrix U_2 so that the product $B_1 \cdot U_1 \cdot U_2$ has all the first two columns as they are in the matrix B_2 and the rest of them are as they are in B_1. Continuing in the same manner we build a matrix $B_1 \cdot U_1 \cdot U_2 \cdots U_n$ that has all the columns of B_2, namely $B_1 \cdot U_1 \cdot U_2 \cdots U_n = B_2$, and because the unimodular matrices of the same dimension are a group inner matrix multiplication we take that $U_1 \cdot U_2 \cdots U_n = U$ with U unimodular as required.

$(\Leftarrow)$

Denote by $\mathcal{L}'$ the lattice produced by B_2, we will show that $\mathcal{L} = \mathcal{L}'$. Since $B_2 = B_1 \cdot U$ and U is unimodular (integer) matrix then we take that every column of B_2 is contained in lattice $\mathcal{L}$, so $\mathcal{L}' \subseteq \mathcal{L}$. It also holds that

$B_1 = B_2 \cdot U^{-1}$, with U^{-1} be unimodular, so $\mathcal{L} \subseteq \mathcal{L}'$. Combining the above relations we take that $\mathcal{L}' = \mathcal{L}$. □

V.II Special Properties Of Lattices

In this section we present some properties of lattices that make them a useful tool to be used in cryptography. We start by giving the definition of the fundamental parallelepiped that is the base for everything that follows.

Definition 55 (Fundamental Parallelepiped)**.** *Let $\mathcal{L}$ be a lattice, and let a basis for $\mathcal{L}$ is $B = [b_1, ..., b_n]$, b_i are the column vectors of B, then we define the set $\mathcal{P}(B) = \{y : y = \sum_{i=0}^{n} x_i \cdot b_i, \quad x_i \in [-\frac{1}{2}, \frac{1}{2})\}$. We call $\mathcal{P}(B)$ the fundamental parallelepiped of $\mathcal{L}$ with respect to the basis B.*

To visualize the sense of the fundamental parallelepiped the reader could consider the figure V.I. The overshadowed areas correspond to the fundamental parallelepipeds of the bases that lie on them.

Through all these years a useful, mathematic, tool in cryptography has been the modular arithmetic. That's because a ring modulo a, prime, number offers a variety of conveniences. First of all, such a ring consists only from distinct elements, this is a necessity when we apply a cryptosystem on a computer. This ring also has a much more helpful property of being a *field* when the modulo is a prime number. Last condition is momentous since we deal with a algebraic construction which offers some of the strongest mathematical properties that we are familiar with, such as the multiplicative inverse.

But what is the role of the modular arithmetic in our construction? This just gives the inspiration for what follows. In this scheme the basic mathematical tool will be the lattices but we will utilize them in a manner that the further work with them will be in the sense of modular arithmetic. The key point towards this direction will be the fundamental parallelepiped. We will use the fundamental parallelepiped as we use the prime number in a modulus ring. Here we have to come up with a definition.

Definition 56. *For a basis B of a lattice $\mathcal{L}$ and a random chosen vector $u \in \mathcal{L}$ define $P_{B,u} = \{\mathcal{P}(B) + u\}$ then we have that the sets $P_{B,u}$ when $u \in \mathcal{L}$ form a partition of the whole space $\mathbb{R}^n$. Also,we denote by $P_{B,u}(v)$ the set $\{\mathcal{P}(B) \cap v + u\}$.*

Remark 2. *It is easy to see that* $P_{B,u}(v) = \begin{cases} v + u & \text{if } v \in \mathcal{P}(B) \\ \emptyset & \text{if } v \notin \mathcal{P}(B) \end{cases}$

Thereon, for a random vector t we will silently use the notation $t = P_{B,u}(v)$ instead of using the notation $t \in P_{B,u}(v)$ taking advantage of the last remark.

Proposition 2. *Let $\mathcal{L} \subseteq \mathbb{Z}^n$ and $b_1, ..., b_n$ be linear independent vectors of space $\mathbb{Z}^n$, then $b_1, ...b_n$ form a basis of $\mathcal{L}$ iff $\mathcal{P}(B) \cap \mathcal{L} = \{\mathbf{0}\}$, where B is the n-dimensional square matrix formed by $b_1, ..., b_n$ as columns.*

Proof. $(\Rightarrow)$

Suppose that $b_1, ..., b_n$ form a basis of $\mathcal{L}$, then

$$\mathcal{P}(B) = \{y : y = \sum_{i=0}^{n} x_i \cdot b_i, \quad x_i \in [-\frac{1}{2}, \frac{1}{2})\}.$$

That is, the only linear combination that holds inside $\mathcal{P}(B)$ is when $b_i = 0, \forall i \in [1, n]$, the point produced then is the $\mathbf{0}$ thus the only lattice point that lies inside $\mathcal{P}(B)$ is the vector $\mathbf{0}$.

$(\Leftarrow)$

Imagine now that $b_1, ..., b_n$ are linearly independent then follows that they all form a basis of $\mathbb{R}^n$. Take now the lattice point $v \in \mathcal{L} \subseteq \mathbb{R}^n$, since $b_1, ..., b_n$ are a basis of $\mathbb{R}^n$ there must be $y_1, ..., y_n \in \mathbb{R}$ so that $v = \sum_{i=0}^{n} y_i \cdot b_i$. We will show that each one of y_i is an integer, that is $y_1, ..., y_n \in \mathbb{Z}$.

First, consider the vector $u = \sum_{i=0}^{n} \lceil y_i \rfloor \cdot b_i$ which is a member of $\mathcal{L}$ since $\lceil y_i \rfloor \in \mathbb{Z}$. Since $v, u \in \mathcal{L}$ it follows that for the vector $t = v - u$ it holds that $t \in \mathcal{L}$, but $t = \sum_{i=0}^{n} (y_i - \lceil y_i \rfloor) \cdot b_i$. Note that $y_i - \lceil y_i \rfloor \in [-\frac{1}{2}, \frac{1}{2})$. However, because $\mathcal{P}(B) \cap \mathcal{L} = \{\mathbf{0}\}$ and $t = \sum_{i=0}^{n} (y_i - \lceil y_i \rfloor) \cdot b_i$ by assumption we take that $t \in \mathcal{P}(B) \Rightarrow t = \mathbf{0}$ which gives that $y_i = \lceil y_i \rfloor = 0 \Rightarrow y_i \in \mathbb{Z}$. □

Now, we briefly describe the connection between the modulus ring and a lattice. In operations of the form $a \mod b$ the result depends on the relative position of a to its nearest multiply of b that is no bigger than a. In fact the space has been partitioned by the sets $P_k = l(b) + kb$, $k \in \mathbb{Z}$, $l(b)$ denotes the half-open set $[O, b)$ and we have to find where a is placed in the set P_k that it belongs without finding which exactly is the set (i.e. the k). The analogous operation on lattices will be: given a point $\mathbf{v} \in \mathbb{R}^n$ find where $\mathbf{v}$ is placed in the $P_{B,u}$ that it belongs, formally:

Definition 57. *Suppose that $\mathcal{L}$ is a lattice defined by the n-dimensional square matrix B, and $v, v' \in \mathbb{R}^n$ then we write $v \equiv v' \mod B$ if $v - v' = P_{B,u}(\mathbf{0})$ for a vector $u \in \mathcal{L}$. We say that vector v is the* distinguished representative *for vector v' according to basis B if v is the unique point $v \in \mathcal{P}(B)$ for which holds that $v' = P_{B,u}(v)$ for a vector $u \in \mathcal{L}$.*

Proposition 3. *The relation $\equiv$ that we defined above is an* equivalence relation.

Proof. Suppose that B is a n-dimensional square matrix that generates lattice $\mathcal{L}$ and a, b, c are vectors of $\mathbb{R}^n$.

- It holds that $a \equiv a \mod B$ because $a - a = 0 = P_{B,\mathbf{0}}(\mathbf{0})$
- If $a \equiv b \mod B \Rightarrow a - b = P_{B,u}(\mathbf{0})$ for a vector $u \in \mathcal{L}$. We also have that $-(a - b) = P_{B,u'}(\mathbf{0})$ for the vector $u' = -1 \cdot u \in \mathcal{L}$ thus, $b - a = P_{B,u'}(\mathbf{0}) \iff b \equiv a \mod B$.
- Suppose that $a \equiv b \mod B$ and $b \equiv c \mod B$ then its true that $a - b = P_{B,u_1}(\mathbf{0})$ for a vector $u_1 \in \mathcal{L}$ and $b - c = P_{B,u_2}(\mathbf{0})$ for a vector $u_2 \in \mathcal{L}$. We now consider the quantity $a - c = a - b + (b - c) = u_1 + u_2 \Rightarrow a - c \in P_{B,u_1+u_2}(\mathbf{0})$, and $u_1 + u_2 \in \mathcal{L}$ since they are both members of the lattice. From the last relation we conclude that $a \equiv c \mod B$

□

Proposition 4. *For a lattice $\mathcal{L}$ generated by the n-dimensional square matrix B a* distinguished representative *of the vector $t \mod B$ is efficiently computable as $t - B \cdot \lceil B^{-1} \cdot t \rfloor$.*

Proof. Suppose that $s \equiv t \mod B$, with $s \in \mathcal{P}(B)$ we will show that

$$s = t - B \cdot \lceil B^{-1} \cdot t \rfloor \tag{V.1}$$

We have already seen that the sets $P_{B,u} = \{\mathcal{P}(B) + u\}$ form a partition of space $\mathbb{R}^n$. Now, let $t \in \mathbb{R}^n$ then t must be contained at a unique $P_{B,u}$ for some $u \in \mathcal{L}$. Also, because $s \in \mathcal{P}(B)$ and $P_{B,u}(s) = t$ then the following equation holds:

$$t = s + u \tag{V.2}$$

with $s = \sum_{i=0}^{n} x_i \cdot b_i, \quad x_i \in [-\frac{1}{2}, \frac{1}{2})$. After the above observations we can now use a part of the reduction formula for $t = s + u$. We take:

$$\lceil B^{-1} \cdot t \rfloor = \lceil B^{-1} \cdot (s+u) \rfloor = \lceil B^{-1} \cdot s + B^{-1} \cdot u \rfloor$$

but because $u \in \mathcal{L}$ holds that $\lceil B^{-1} \cdot u \rfloor = B^{-1} \cdot u$ therefore

$$\lceil B^{-1} \cdot t \rfloor = \lceil B^{-1} \cdot s \rfloor + B^{-1} \cdot u \tag{V.3}$$

Now we focus on eliminating the $\lceil \cdot \rfloor$to the term with s. In this direction we have to use the relation of s with the generator matrix B. That is, $s = \sum_{i=0}^{n} x_i \cdot b_i, \quad x_i \in [-\frac{1}{2}, \frac{1}{2})$ where b_i denotes the $i-$th column of matrix B. Denote now by b_i' the $i-$th row of matrix B^{-1}, here must be clear that $b_i' \cdot b_i = 1$ for every $i \in [1, n]$ and also that $b_i' \cdot b_j = 0$ for every $i, j \in [1, n]$ when $i \neq j$. Now we can compute the $i-$th coordinate of vector $B^{-1} \cdot s$, which equals to $b_i' \cdot \sum_{i=0}^{n} x_i \cdot b_i = x_i \in [-\frac{1}{2}, \frac{1}{2})$ after this we have $\lceil B^{-1} \cdot s \rfloor = \mathbf{0}$. Combining now the last result with equation V.3 we take

$$\lceil B^{-1} \cdot t \rfloor = B^{-1} \cdot u$$

so we can now compute the product

$$B \cdot \lceil B^{-1} \cdot t \rfloor = u$$

witch combined to the equation V.2 gives that

$$t - B \cdot \lceil B^{-1} \cdot t \rfloor = t - u = s$$

so the equation V.1 holds. $\square$

Chapter ***VI***

Lattices for Cryptography

VI.I Lattice Associated Problems

SINCE lattices are going to be used as the main mathematical tool in this construction we have to review some of the most famous lattice based algorithmic problems that, up to this date, lack of efficient algorithms. The most famous problems are the *Shortest Vector Problem*, the *Closest Vector Problem* and the *Shortest Independent Vectors Problem.* These three problems are the most common to which one can reduce a problem to prove security for a lattice-based cryptographic scheme. Next we give a brief description for each of them.

Shortest Vector Problem **(SVP):** Given a lattice basis B, find the shortest nonzero vector in $\mathcal{L}(B)$.

Input a lattice basis B.

Output $v \in \mathcal{L} : \|v\| \leq \|x\| \; \forall x \in \mathcal{L}$.

Closest Vector Problem **(CVP):** Given a lattice basis $B \in \mathbb{Z}^n$ and a vector $t \in \mathbb{Z}^n$ find the lattice point that is closest to t.

Input a lattice basis B, a vector $t \in \mathbb{Z}^n$

Output $v \in \mathcal{L} : \|v - t\| \leq \|x - t\| \; \forall x \in \mathcal{L}$

***Shortest Independent Vectors Problem* (SIVP):** Given a lattice basis B and k linearly independent lattice vectors $S = [v_1, v_2, ..., v_k]$, with $k < n$ find the shortest possible lattice vector that is independent to the vectors of S.

Input a lattice basis B, k linearly independent lattice vectors $S = [v_1, v_2, ..., v_k]$

Output $v \in \mathcal{L} : a_1 \cdot v_1 + a_2 \cdot v_2 + \cdots a_k \cdot v_k + a_{k+1} \cdot v = 0 \iff a_i = 0, i \in [1, k+1]$, $a_i \in \mathbb{Z}, i \in [1, k+1]$ with $\|v\| \leq \|v'\|$, $\forall v'$: $a_1 \cdot v_1 + a_2 \cdot v_2 + \cdots a_k \cdot v_k + a_{k+1} \cdot v' = 0 \iff a_i = 0, i \in [1, k+1]$, $a_i \in \mathbb{Z}, i \in [1, k+1]$

The majority of lattice-based cryptography schemes base their security on reductions to approximation variants of these three problems. The above problems though are not always hard to be solved, but in case we have been provided a "good" basis of a lattice then it is feasible to solve such problems with low complexity circuits. For each of these problems the arduousness or the easiness depend on the basis of the lattice.

According to the three problems presented above becomes clear that its shortest vectors are the main factor for solving hard problems. Relying on this we have to develop a symbolism for each class (according to their length) of lattice vectors.

Definition 58 (Shortest Vector)**.** *Let $\|\cdot\|$ be an arbitrary norm and $\mathcal{L}$ a lattice. We define the* shortest vector *of $\mathcal{L}$ with respect to the norm $\|\cdot\|$ to be the vector $\lambda_1 \in \mathcal{L}$ such that:*

$$\forall u \in \mathcal{L} \setminus \{0\} \Rightarrow \|\lambda_1\| \leq \|u\|$$

From now on we denote the shortest vector of a lattice with λ_1. Also, we affiliate a notation for every level of a vector's length.

Definition 59 (Successive Minima)**.** *Let $\|\cdot\|$ be an arbitrary norm and $\mathcal{L} \subseteq \mathbb{Z}^n$ a lattice. The successive minima $\lambda_1, \lambda_2, ..., \lambda_n$ of $\mathcal{L}$ with respect to the norm $\|\cdot\|$ defined recursively as:*

1. *λ_1:= the shortest vector of $\mathcal{L}$.*
2. *$\lambda_i := inf\{u \in \mathcal{L} : \{\lambda_1, ..., \lambda_{i-1}, u\}$ are linearly independent.$\}$*

Since, as before stated, the key role held by the shortest vector of a lattice, we have to prove that this vector exists and that it is not only the convergence point of a lattice points' sequence (x_n) without belonging to (x_n).For this purpose we prove the following lemma.

Lemma 3. *Let $\mathcal{L} \subseteq \mathbb{Z}^n$, and also let $e_i, i \in [1,n]$ denote the vector of $\mathbb{Z}^n$ which consists of zeros but its i-th coordinate is* 1*. Then $\exists k \in [1,n] : \|e_k\| \leq \|\lambda_1\|$.*

Proof. The vectors $e_i, i \in [1,n]$ form a basis of the space $\mathbb{Z}^n$, note also that for an arbitrary vector $u \in \mathbb{Z}^n$ holds that: $u = a_1 \cdot e_1 + a_2 \cdot e_2 + \cdots + a_n \cdot e_n$ with $a_i \in \mathbb{Z}$, for every $i \in [0,n]$. The last holds also for every point of the lattice $\mathcal{L}$ since it is a subset of $\mathbb{Z}^n$.

Consider now a $u \in \mathcal{L} \setminus \{0\}$, there must exist $a_1, a_2..., a_n \in \mathbb{Z}$ with at least one of them different than zero so that: $u = a_1 \cdot e_1 + a_2 \cdot e_2 + \cdots + a_n \cdot e_n$, then we take:

$$\|u\| = \|a_1 \cdot e_1 + a_2 \cdot e_2 + \cdots + a_n \cdot e_n\|$$

Suppose that $a_k \neq 0$, $k \in [1,n]$, for e_k also holds that:

$$\|a_k \cdot e_k\| \leq \|a_1 \cdot e_1 + a_2 \cdot e_2 + \cdots + a_n \cdot e_n\|$$

which gives that:

$$|a_k| \cdot \|e_k\| \leq \|a_1 \cdot e_1 + a_2 \cdot e_2 + \cdots + a_n \cdot e_n\|$$

But, the property of a_k to be a non-zero integer indicates that $1 \leq |a_k|$, after this it follows that:

$$\|e_k\| \leq \|a_1 \cdot e_1 + a_2 \cdot e_2 + \cdots + a_n \cdot e_n\| = \|u\|$$

We have shown that $\|e_k\|$ is a lower bound of the set

$$S = \{x \in \mathcal{L} : 0 < \|x\|\}$$

Actually, the vector λ_1 is defined as $\lambda_1 = inf\{S\}$, the last two remarks imply that $\|e_k\| \leq \|\lambda_1\|$. □

Lemma 4. *For every lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ there exist a fixed $\phi \in \mathbb{R}$ with $\phi > 0$ such that for every two vectors $u, u' \in \mathcal{L}$ holds that $\|u - u'\| < \phi \Leftrightarrow u = u'$*

Proof. From the previous lemma we have that $\exists k \in [1,n] : \|e_k\| \leq \|\lambda_1\|$, the proof follows if we set $\phi = \|e_k\|$. □

Last two lemmas point out a strong conclusion, that the shortest vector of a lattice is a member of it. The last is not trivial since λ_1 is defined as an infimum of a particular set of lattice points. It would not be surprise if λ_1 was not member of this set since, generally, is not true that the convergence point of a sequence belongs to it.

Lemma 5. *Let $\mathcal{L}$ be a lattice subset of $\mathbb{Z}^n$, then the shortest vector of $\mathcal{L}$ is a member of it. That is, $\lambda_1 \in \mathcal{L}$.*

Proof. By its definition the shortest vector is $\lambda_1 = inf\{x \in \mathcal{L} : 0 < \|x\|\}$, so there exist sequence $(x_n) \in \mathcal{L}$ consisting of lattice points, such that $\lim\limits_{n\to\infty} x_n = \lambda_1$. Consider now the closed sphere $\mathcal{B}(\lambda_1, \phi/3) = \{x \in \mathbb{R}^n : \|x - \lambda_1\| \leq \phi/3\}$. Since $x_n \to \lambda_1$ there are infinite terms of (x_n) in the sphere $\mathcal{B}(\lambda_1, \phi/3)$.

Define now, recursively, the strictly ascending sequence s_n as follows:

- $s_1 = k \in \mathbb{N} : (x_k \in \mathcal{B}(\lambda_1, \phi/3)) \wedge (\forall m : x_m \in \mathcal{B}(\lambda_1, \phi/3) \Rightarrow k \leq m)$
- $s_{i+1} = k \in \mathbb{N} : (k > s_i) \wedge (x_k \in \mathcal{B}(\lambda_1, \phi/3)) \wedge (\forall m > s_i : x_m \in \mathcal{B}(\lambda_1, \phi/3) \Rightarrow k \leq m)$

Observe that (s_n) is consisting of all the indexes for which the sequence (x_n) falls within the sphere $\mathcal{B}(\lambda_1, \phi/3)$.

Next, we construct a subsequence of (x_n) by the following formula:

$$y_n = x_{s_n}$$

Every term of the sequence (y_n) is a member of the ball $\mathcal{B}(\lambda_1, \phi/3)$. We have already assumed that $x_n \to \lambda_1$, also $(y_n) \subseteq (x_n)$, thus $y_n \to \lambda_1$.

The compactness of $\mathcal{B}(\lambda_1, \phi/3)$ implies that for every converged sequence its convergence point lies inside the sphere $\mathcal{B}(\lambda_1, \phi/3)$, this is the point λ_1. Take now two terms of (y_n) we have that $\|y_i - y_j\| \leq \|y_i\| + \|y_j\| \leq \phi/2 + \phi/3 < \phi$. According to the lemma 4, the last inequality yields that every to points of the sequence (y_n) are identical. The last implies that every the convergence point of (y_n), which is λ_1, is also identical to every term of (y_n). By hypothesis, the sequence (x_n) is a lattice points sequence, the same holds for (y_n) as it is a subsequence of (x_n), this gives that λ_1 is also a lattice point as desired.

□

In order to understand how a basis induces a hard or a easy instance of these problems in next subsection we focus on describing "good" and "bad" lattice bases.

VI.II ‘Good‘ and ‘Bad‘ Lattice Bases

We have already described the *mod* operation for a lattice $\mathcal{L} \subseteq \mathbb{Z}^n$, that involves a, random, point $t \in \mathbb{R}^n$ and a basis B that produces $\mathcal{L}$. The

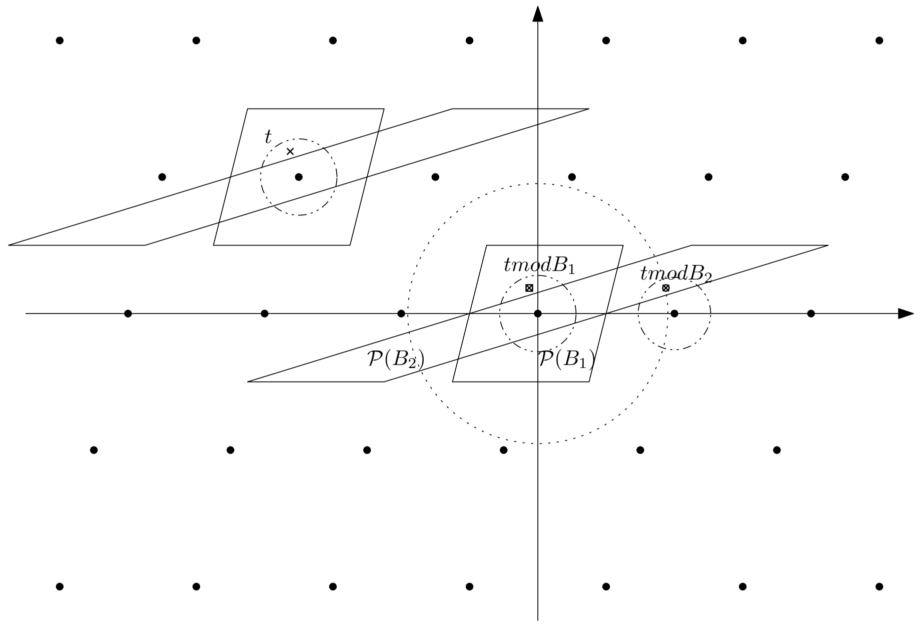

Figure VI.I: The mod operation.

reason that produces hard lattice problems is the use of this operation and its property to output equivalent but not identical points when applied for more than one bases of $\mathcal{L}$.

Proposition 5. *Let $\mathcal{L} \subseteq \mathbb{Z}^n$ be an integer lattice and B_1, B_2 two bases of $\mathcal{L}$. Consider a point $t \in \mathbb{R}^n$, then if $s_1 \equiv t \mod B_1$ and $s_2 \equiv t \mod B_2$, with s_1, s_2 being the* distinguished representatives*, it holds that $s_1 \mod B_2 \equiv s_2$ and $s_2 \mod B_1 \equiv s_1$, also the equality $s_1 = s_2$ does not always hold.*

Proof. First of all, it suffices to show that $s_2 \equiv s_1 \mod B_2$. Since $s_1 \equiv t \mod B_1$ then $t = P_{B_1,u_1}(s_1)$, thus $t = u_1 + s_1$, for some $u_1 \in \mathcal{L}$, similarly to this we take that $t = u_2 + s_2$, for some $u_2 \in \mathcal{L}$. Combining the above two relations we have $s_1 = u + s_2$, with $u = u_2 - u_1 \in \mathcal{L}$, namely $s_1 = P_{B_2,u}(s_2) \Longleftrightarrow s_1 \equiv s_2 \mod B_2 \Longleftrightarrow s_2 \equiv s_1 \mod B_2$. □

Last result stands here just to emphasize the importance of the existence of "good" and "bad" bases. In previous section we brought up the three most well-known computational problems on lattices. Observe now that every single problem of them deals with the notion of the distance, specifically with the notion of a norm. They all ask the smallest norm of a lattice vector, the smallest norm of a random vector according to a lattice vector or the smallest norm of a lattice vector under some restrictions. Essentially, they don't require the norm as a measure of $\mathbb{Z}^n$ but as a measure being produced from lattice points for lattice points. One has to understand last proposition in order to apprehend the difficulties that come out with the absence of a basis that visualizes the structure of the lattice that it describes.

Hereafter, by the term *"good" basis* we indicate a basis that its vectors are nearly orthogonal. We also will use the term *"bad" basis* for those that are consisting of vectors that are closer to being parallel than being orthogonal to each other. As a rule of thumb a basis with fatter fundamental parallelepiped is the best among different bases. Figure VI.II illustrates this relation between "good" and "bad" lattice bases. Note that the fattest fundamental parallelepiped takes up the space closer to the lattice point that it circumscribes, in contrast the thiner basis takes up space that has some parts closer to different points from the one that it circumscribes. The advantage of having a "good" basis is that after a reduction the reduced point's closest lattice vector is probably the center of the fundamental parallelepiped.

In order to substantiate the importance of better bases imagine the following game between two players $\mathcal{A}$ and $\mathcal{B}$:

$\mathcal{A}$ is aware of a "good" basis B_1 for the lattice $\mathcal{L}$ and $\mathcal{B}$ has nothing but a "bad" basis B_2 for the same lattice. Then we publish a point $t \in \mathbb{Z}^n$ and ask them to find the closest lattice vector to t. $\mathcal{A}$'s strategy could be:

1. compute $v \leftarrow t \mod B_1$

2. return $t - v$

Then $\mathcal{A}$ most likely has return the correct point, since he is aware of a "good" basis which implies that after the reduction $\mod B_1$ the resulting point has 0 as closest lattice point. After this he knows that v is the vector that shows the difference between t and its closest lattice point since after the modulo operation the orientation of v remains unchanged compared with the points around it. Thus, $\mathcal{A}$ could return $t - v$ as the closest lattice point to the target t.

The same strategy does not work for player $\mathcal{B}$ because there is no guarantee that after the reduction the resulting point's closest point would be 0 or at least a fixed lattice point. The reason that produces this deformity is the bad shape of the basis B_2 which means that it is possible for some points that lie in $\mathcal{P}(B_2)$ to have closest lattice points other than the basis "center" which is the point 0.

VI.III LLL Reduction

After the above game it must be clear that one of the most important issues for one that has to solve lattice problems is the awareness of a "good" lattice basis. We show that $\mathcal{A}$ had one kind of advantage because the basis B_1 gives him a more visualized illustration of the lattice than $\mathcal{B}$ could have. Let's try now to help $\mathcal{B}$ in order to balance the advantage between the two players. $\mathcal{B}$'s main problem is the basis that he has, if we want to help him, then we have to give him a way of creating a new, better basis that the one he already has.

First, we describe the *Gram−Schmidt* orthogonalization process, named after its inventors, this is an efficient algorithm that produces orthogonal basis starting of arbitrary ones. This process for producing orthogonal bases from arbitrary bases is one of the most famous algorithms in finding orthogonal bases.

We first define the *projection operation* for an element a onto the space spanned by vectors $S = \{a_1, a_2, ..., a_k\}$ by $proj_S(a) = \sum_{i=1}^{k} \frac{\langle a, a_i\rangle}{\langle a_i, a_i\rangle} \cdot a_i$. Now we are ready to describe the algorithm.

Algorithm 2. *(Gram-Schmidt Orthogonalization)*
Input: *An arbitrary basis* $B = \{b_1, b_2, ..., b_n\}$ *of a subspace of* $\mathbb{R}^n$.
Output: *An orthogonal basis of the same subspace.*

```
1. Compute $v_1 = b_1$, and $u_1 = \frac{v_1}{\|v_1\|}$
2. Compute $v_i = b_i - \sum_{j=1}^{i-1} proj_{b_j}(v_i)$, and $u_i = \frac{v_i}{\|v_i\|}$
3. Return $\{u_1, u_2, ..., u_n\}$
```

So, is this the hint that we are looking for? Unfortunately, the above algorithm works efficiently only in two dimensions. The Gram-Schmidt orthogonalization is effective and being used for $n < 3$, in dimensions bigger than 2 in generally orthogonalization is not an easy task and up to this day all known algorithms are lagging in time complexity.

Next we present the fastest known algorithm that can produce nearly orthogonal bases in higher dimensions. This is an algorithm that produces reduced bases which essentially are closer to a "good" basis of a lattice than an arbitrary one. We first give the formal definition of the reduced basis.

For simplicity we denote by $\mu_{i,j}$ the terms $\dfrac{\langle v_i, b_j\rangle}{\langle b_j, b_j\rangle}$ that arise in the Gram-Schmidt orthogonalization algorithm.

Definition 60 (LLL Reduced Basis)**.** *Let $\mathcal{L}$ be a lattice and B a basis of it. We denote by b_i the i-th column of the matrix B. B is said to be* LLL reduced *with parameter δ, $1/4 \leq \delta \leq 1$ if the following conditions are satisfied:*

- $|\mu_{i,j}| \leq 1/2$, $\forall i > j$.
- *For any pair of consecutive vectors b_i, b_{i+1}, $i < n$ we have that*

$$\delta \| proj_{a_j}(b_i)\|^2 \leq \|proj_{a_j}(b_{i+1}))\|^2.$$

Now that we have defined our goal we can proceed to the description of the LLL algorithm that was first introduced by A. Lenstra, H. Lenstra and L. Lovász in 1982 and still remains the best known algorithm for basis reduction.

Algorithm 3. *(LLL Reduction Algorithm)*
Input: *A lattice basis $B = \{b_1, b_2, ..., b_n\} \in \mathbb{Z}^n$.*
Output: *A δ-LLL-Reduced basis for the same lattice.*

```
1. Compute the basis $b_1^*, b_2^*, ..., b_n^*$ that the Gram-Schmidt algorithm produces.
2. Set $b_i \leftarrow b_i - \lceil \frac{\langle b_i, b_j^*\rangle}{\langle b_j^*, b_j^*\rangle} \rfloor$, $i \in [2, n]$ and $j \in [1, i-1]$.
3. If there is $i$ such that $\|b_{i+1}^* + \mu_{i+1,i} \cdot b_i^*\|^2 < \delta \|b_i^*\|^2$ then
    (a) Swap $b_i$ with $b_{i+1}$.
    (b) Run LLL Reduction Algorithm with input $\{b_1, b_2, ..., b_n\}$.
4. Return $\{b_1, b_2, ..., b_n\}$
```

Suppose now that we have an arbitrary basis B of a lattice $\mathcal{L}$, then we can take advantage of the above algorithm and produce a reduced basis B' which describes $\mathcal{L}$ also but it gives us better visualization for the lattice that the initial basis. As we already have seen for a basis that is closer to an orthogonal one the corresponding fundamental parallelepiped is fatter than others and it helps to recover closest lattice points to when given a random point of the space. Though, it does not buy us anything, because there is always the possibility the reduced point to fall within an area that it is not closer to the center of the fundamental parallelepiped. This becomes an important issue if we consider that we have no guarantee that we can find an orthogonal basis, actually the LLL algorithm just approximates the best possible basis within a factor.

Algorithm 4. *(Babai's Nearest Plane Algorithm)*
Input: *A lattice basis* $B =\in \mathbb{Z}^{m\times n}$ *and a point* $t \in \mathbb{Z}^m$.
Output: *A vector* $x \in \mathcal{L}(B)$ *such that* $\|x - t\| \leq 2^{\frac{n}{2}} \cdot dist(t, \mathcal{L}(B))$.

```
1. Run $\delta$-LLL on $B$ with $\delta = 3/4$.
2. Set $b \leftarrow t$.
3. $b \leftarrow b - \lceil \frac{\langle b, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle} \rfloor$ for $i = n$ downto $1$.
4. Output $t - b$
```

It can be seen that this algorithm runs in polynomial time in the input size, indeed, the LLL procedure runs in polynomial time and allows the last algorithm to have this property. Notice also that unlike our description of the LLL algorithm, here we consider the algorithm for arbitrary lattices that are not necessarily full-rank.

Proposition 6. *Suppose that* B *is the basis for the lattice* $\mathcal{L}$. *Then for any* $t \in \mathcal{L}$, *the output* x *of the algorithm is such that* $\|x - t\|^2 \leq \frac{1}{4}\sum_{i=1}^{n} \|b_i\|$.

Proof. Since B is LLL reduced we have that: $\forall i \in [1, n], \|b_i\| \leq 2^{\frac{n-i}{2}} \cdot b_n$. Which implies the following:

$$\|x - t\|^2 \leq \frac{1}{4}\sum_{i=1}^{n} \|b_i\|^2 \leq \frac{1}{4}\sum_{i=1}^{n} 2^{n-i} \cdot \|b_n\|^2 \leq \frac{1}{4}2^n \cdot \|b_n\|^2$$

as claimed. □

Proposition 7. *For any* $t \in \mathbb{Z}^m$, *let* $y \in \mathcal{L}(B)$ *be the closest lattice point to* t. *Then the algorithm described above finds a point* $x \in \mathcal{L}(B)$ *such that* $\|x - t\| \leq 2^{\frac{n}{2}} \cdot \|y - t\|$.

Proof. We prove by induction on the rank n that our algorithm finds a point $x \in \mathcal{L}(B)$ such that $\|x - s|\| \leq 2^{\frac{n}{2}}\|y - s\|$. This yields the claim, since

$$\|x - t\|^2 = \|s - t\|^2 + \|s - x\|^2$$

because $s-t$ and $s-x$ are orthogonal, which gives that

$$\|x-t\|^2 \leq \|s-t\|^2 + 2^n \cdot \|y-s\|^2 \leq 2^n \cdot (\|s-t\|^2 + \|y-s\|^2) = 2^n \cdot \|y-t\|^2$$

where the last equality follows because $s-t$ and $s-x$ are orthogonal.

Now distinguish two cases. If $\|s-y\| < \frac{\|b_n\|}{2}$, then $y \in c \cdot b_n + span(b_1, ..., b_{n-1})$, because all other hyperplanes are of distance at least $\frac{\|b_n\|}{2}$. Therefore, $y \in c \cdot b_n + \mathcal{L}(b1, ..., b_{n-1})$. Intuitively this means that we identified the correct translate in Step 2. So we obtain that $y' = y - c \cdot b_n \in \mathcal{L}(b_1, ..., b_{n-1})$ is the closest point to s'. Hence, by our inductive assumption.

$$\|x-s\| = \|x'-s'\| \leq 2^{\frac{n-1}{2}} \cdot \|y'-s'\| = 2^{\frac{n-1}{2}} \cdot \|y-s\| \leq 2^{\frac{n}{2}} \cdot \|y-s\|.$$

Otherwise, we must have that $\|s-y\| \geq \frac{\|b_n\|}{2}$ In this case, it is possible that we identify the . wrong translate in Step 2. However, by proposition 7, we have that $\|s-x\| \leq \dfrac{1}{2} \cdot 2^{\frac{n}{2}} \cdot \|b_n\| \leq 2^{\frac{n}{2}} \cdot \|s-y\|$

□

VI.IV Hermite Normal Form

In last two subsections we established the importance of different kind of bases for the same lattice. It should be clear at this point that in order to proceed with a lattice based cryptographic protocol we need a stepping stone for constructing "bad" bases for a lattice starting from "good" ones. Here we present the notion of the HNF for a matrix which is the key for constructing "bad" bases.

Definition 61 (Hermite Normal Form). *Consider a matrix $B \in \mathbb{Z}^{n \times m}$ the* Hermite normal form (HNF) *of B is the unique matrix $H = (h_{i,j})$ such that there is a unimodular $n \times n$ matrix U with $U \cdot B = H$, and such that H satisfies the following two conditions:*

- *H is upper triangular.*
- *There exist a sequence of integers $j_1 < \cdots < j_n$ such that for all $0 \leq i \leq n$ we have $h_{i,j} = 0$ for all $j < j_i$.*
- *for $0 \leq k < i \leq n$ we have $0 \leq h_{k,j_i} < h_{i,j_i}$.*

In some sense, the worst basis of a lattice $\mathcal{L}$ is its unique upper-triangular Hermite normal form. Given any basis B of $\mathcal{L}$, one can compute HNF($\mathcal{L}$)

efficiently. Geometrically, the HNF of a lattice basis is a basis for the same lattice but much more thinner than the starting one. Thus, HNF($\mathcal{L}$) does not "reveal" more about $\mathcal{L}$'s structure than any other basis, making HNF($\mathcal{L}$) a good choice for the public lattice basis to be included in a public key.

VI.V Ideal Lattices

For our purpose we use lattices that are defined by polynomials in polynomial rings. But, we have seen that a lattice consists of a set of vectors, so we have to map polynomials to vectors in order to achieve our goal. The mapping is the obvious one, this that takes every coefficient of a polynomial and maps it to the corresponding coordinate of a vector. Namely, let $f = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$ be a polynomial, then we map f to the vector $v_f = (a_n, a_{n-1}, ..., a_1, a_0)$. One more obstacle is the dimension of the vectors that we produce. It is clear that a lattice consists only of vectors of same dimension, but if we consider the above mapping between polynomials and vectors then the vector dimensions will differ to each other. So, how do we fix the dimension of the produced vectors? If we need, say, n dimensions for our lattice then we consider every polynomial modulo a fixed polynomial of degree n. After the first observations one now can see that a lattice does not only what a, basis, matrix produces, but it may represent other algebraic constructions. The questions that now come in mind are a) can one define mathematical operations similar to addition and multiplication for the points of a lattice? b) if the first answer is yes, then, do the points of a lattice form groups or fields under these operations? The answer is yes for the first question, the two operations will be exactly as they are defined in the ring modulo a polynomial. For the second question the answer is yes for a group and under the restriction that the polynomial used as module is irreducible then lattice points form a field.

The previous discussion has no practical aspects to our construction, but it has been made to clarify that we move forward to a strong algebraic construction and at any time we need everything in this direction lattices offer compliance and provide us with at least the basics of a useful mathematical construction. So we can use their structure when its needed in our proofs.

The study of lattices so far demonstrates that they supply a main feature for public key cryptography which is the easiness of making hard instances of problems and also calculating trapdoors for them. Nevertheless, they all have a common nature which, unfortunately, provides them with a deficient property that has not been mentioned so far. That is, cryptography schemes need fast algorithms and keys with small bit length. The second requirement, though has not been satisfied since if we use lattices as public and private keys for a scheme one needs huge amount of memory to store them. Imagine only that with classic cryptography, witch uses operations in $\mathbb{Z}_p$, the length of p starts to become prohibitive as keys expanding to resist cryptanalytic attacks, if we use lattices then only for the basis of a "small" lattice of dimensions, say 10×10, we need to store 100 numbers that each of them may be oversized. The problem of finding a family of lattices which require only a reasonable amount of memory to be stored has an answer, the solution is *ideal lattices*.

Ideal lattices offer one more desired property, they have additive and multiplicative structure. We can take advantage of it by developing a homomorphic scheme homomorphic for both operations. If we succeed in it then we have finish since the set $\{+, \times\}$ is a complete set of circuits (i.e. we can express every circuit in terms of $+$ and $\times$).

Now we have two strong reasons to study ideal lattices. Just before we define this new structure we present a way of mapping polynomials to vectors in order to start thinking of them as members of $\mathbb{Z}^n$.

We match every single element $g(x)$ of the ring $\mathbb{Z}[x]/f(x)$ to a, unique, vector v_g so that, if $deg(g(x)) = n$, then $v_P \in \mathbb{Z}^{n+1}$. And the $i-$th $(1 \leq i \leq n+1)$ coordinate of v_P is the coefficient of $x^{(n+1)-i}$.

Definition 62. *Consider the ring $\mathcal{R}[x] = \mathbb{Z}[x]/\langle f(x)\rangle$, and an ideal $I \subseteq \mathcal{R}[x]$. We call ideal lattice the set $\mathcal{L} \subseteq \mathbb{Z}^n$ such that $v_g \in \mathcal{L} \Leftrightarrow g(x) \in I$.*

From the definition given above it is obvious that if $deg(f(x)) = n$, then every single element of the ring $\mathcal{R}[x]$ is a polynomial with degree at most $n-1$, therefore all members of $\mathcal{L}$ are vectors of n dimensions.

We also notice that an ideal lattice can be described only by $f(x), I$ as they defined above. Especially in the case where $I = \langle g(x)\rangle$, i.e. when I is a principal ideal, then the corresponding ideal lattice $\mathcal{L}$ is fully defined only by $f(x), g(x)$. Hereinafter when we write $[f(x), g(x)]$ we refer to the ideal lattice $\mathcal{L}$ witch corresponds to the ideal $\langle g(x)\rangle$ in the ring $\mathbb{Z}[x]/\langle f(x)\rangle$.

Lemma 6. *If $f(x)$ is irreducible then $\mathcal{L} = \mathbb{Z}^n$, when $deg(f(x)) = n$.*

Proof. Indeed, if $f(x)$ is irreducible then $gcd(f(x), g(x)) = 1$ thus there exist $a(x), b(x) \in \mathbb{Z}[x]/\langle f(x)\rangle$ so that $a(x) \cdot f(x) + b(x) \cdot g(x) = 1$ thus, $b(x) \cdot g(x) \equiv 1$ as elements of $\mathbb{Z}[x]/f(x)$. For random $q(x) \in \mathbb{Z}[x]/f(x)$ we have that $(q(x) \cdot b(x)) \cdot g(x) \equiv q(x)$ in $\mathbb{Z}[x]/f(x)$ therefore $q(x) \in < g(x) >$, so, we have shown that: $\langle g(x)\rangle = \mathbb{Z}[x]/f(x)$ thus the corresponding vectors of the elements of $\langle g(x)\rangle$ cover all $\mathbb{Z}^n$. $\square$

Next we prove that a basis of an n-dimension ideal lattice can be described only by a vector of n elements than a $n \times n$ matrix.

Lemma 7. *The ideal lattice produced by* $[f(x), g(x)]$ *is generated by the column vectors* $\{g(x) \cdot x^i \mod f(x) : i \in [0, n-1]\}$. *Also we call this basis the* rotation basis *of the ideal lattice* $[f(x), g(x)]$.

Proof. It suffices to show that $\{g(x) \cdot x^i \mod f(x) : i \in [0, n-1]\}$ is a basis of the ideal $\langle g(x) \rangle$ as an element of $\mathbb{Z}[x]/\langle f(x) \rangle$. Take now the random polynomial $h(x) \in \langle g(x) \rangle$ from the last relation we have that

$$h(x) = p(x) \cdot g(x),\ p(x) \in \mathbb{Z}[x]/\langle f(x) \rangle.$$

Since $p(x) = a_{n-1}x^{n-1} + \cdots + a_1 x^1 + a_0 x^0$ for some $a_0, a_1, ..., a_{n-1} \in \mathbb{Z}$ we have $h(x) = \sum\limits_{i=0}^{n-1} a_i \cdot x^i \cdot g(x)$, that is, every random element $h(x) \in \langle g(x) \rangle$ depends only on the choice of $a_0, a_1, ..., a_{n-1}$ thus, the set $\{g(x) \cdot x^i \mod f(x) : i \in [0, n-1]\}$ form a basis for the ideal $\langle g(x) \rangle$. □

Lemma 8. *For an ideal lattice produced by* $[f(x), g(x)]$ *with* $f(x) = x^n - 1$ *we have that its rotation basis has columns all possible circular permutations of the coefficients' vector of* $g(x)$.

Proof. Let $g(x) = a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + ... + a_0$, then but $g(x) \cdot x^i = f(x) \cdot p(x) + r(x)$ for

$$p(x) = \sum_{j=1}^{j=i} a_{n-j} \cdot x^{i-j}$$

and

$$r(x) = \sum_{j=i}^{j=n+i} a_{n-i-j} \cdot x^{n-j+1}$$

So that $r(x)$ has the required form and for every i its coefficients are cyclically rotated. □

Chapter **VII**

An Abstract Scheme

HERE we present an example on how to operate addition and multiplication homomorphically using the space $\mathbb{Z}$. This leads us to the main idea that is behind Gentry's homomorphic encryption scheme.

Choose two primes (or coprimes) integers p and q and take q to be much smaller than p. In this scheme the "plaintext" space will be the set $\mathcal{P} = \{0, 1, 2, ..., q-1\}$ and the "ciphertext" space the set $\{0, 1, ..., p-1\}$.

The "encryption" procedure for a $x \in \mathcal{P}$ is: find a $k \in \mathbb{Z}$ such that $x \cdot k < p$ define the "ciphertext" to be $\psi = k \cdot x \mod p$. Then, one recovers x when he computes the result of the operation $(\psi \mod p) \mod q$. Also, we can define the addition and multiplication operations homomorphically. Let $x_1, x_2 \in \mathcal{P}$ and $\psi_1 = k \cdot x_1 \mod p$, $\psi_2 = l \cdot x_2 \mod p$ their corresponding "ciphertexts". If the sum $k \cdot x_1 + l \cdot x_2$ remains smaller than p then the "ciphertext" $\psi_1 + \psi_2$ is the "encryption" of $x_1 + x_2$ the same holds for the multiplication when the product $k \cdot x_1 \cdot l \cdot x_2$ is smaller than p.

The above construction seems reasonable for the homomorphic operations but there is one big issue, this is not a cryptographic scheme. The problem though is obvious, as we don't have distinguish the public and private keys. In order to made this scheme a public encryption scheme we have to modify the fake encryption procedure by replacing the $\mod p$ operation by a $\mod p'$ operation. But for a plaintext $x \in \mathcal{P}$ take the ciphertext $\psi = x \mod p'$ then must hold that $\psi \mod p = x$. From this simple example it is obvious that we can't find such a $p' \neq p$. But what if we could? Then a homomorphic scheme would begin to comes out. The real answer though, to the previous problem, is that we can construct such a scheme but without using integers or numbers modulo an integer, but with the utilization of lattices. In order to achieve such a goal we need to find out a way of equivalent modulo reduction. That is, we need to replace a, private known only, modulo ring, say R_1, with another, public, modulo ring, R_2, without R_2 revealing any information about R_1 but with R_1 to be a ring that offers the easiness of calculating with low cost circuits some stuff that is not easy to calculate in the second ring.

The answer in the above quest hides in lattices. We have already seen that a lattice can be defined by a infinite, countable, set of different bases. Also, for every basis B we defined the reduction modulo B (i.e. the operation mod B) and, furthermore, we have seen that only a few "good" bases provide easiness in computing *shortest lattice vectors* or *closest vectors* to our lattice.

VII.I A first construction

We now present a scheme that is homomorphic under addition and multiplication. This will give us a first sense in how the finally scheme will look like. First we define the problem in which is based the security of the following scheme.

Definition 63. *(Ideal Coset Problem) Let R be a ring, B_I is a basis of the ideal $I \subset R$, $Samp_1$ is an algorithm that efficiently samples from R and also let $IdealGen(R, B_I)$ is an algorithm that produces two bases of an ideal J in R which is relatively prime to I. We define the ideal coset problem to be the following game: The first player chooses a $b \overset{R}{\leftarrow} Samp_1(R)$ and two bases $(B_J^{sk}, B_J^{pk}) \overset{R}{\leftarrow} IdealGen(R, B_I)$. If $b = 0$ he chooses $r \overset{R}{\leftarrow} Samp_1(R)$ and $t \leftarrow r \mod B_J^{pk}$. If $b = 1$ he chooses randomly a t from $R \mod B_J^{pk}$. The second player has to find b given (t, B_J^{pk}).*

Now we describe every contraption that the scheme uses as black box or that requires to be secure.

We define the algorithm $IdealGen$, this algorithm takes as input a ring R and a basis of an ideal $I \subset R$ and outputs a basis for an ideal $J \subset R$ such that $I + J = R$, i.e. I and J are relatively prime.

Next, we use the algorithm $Samp$ which takes an input of the form (x, B_I, R, B_J) and outputs a random chosen polynomial from the coset $x + I$.

Here is a scheme that gives us homomorphic operations for circuits consisting of gates Add_{B_I} and $Mult_{B_I}$, i.e. the gates of addition and multiplication respectively in the space of cosets of I. Bellow we geve a detailed description of the algorithms that the scheme uses:

Scheme 4 (Abstract)**.**

KeyGen(R, B_I)**:** *the input here is a ring R and a basis B_I of the ideal $I \subset R$. It returns a pair $(B_J^{sk}, B_J^{pk}) \xleftarrow{R} IdealGen(R, B_I)$. The public key is the pair R, B_I, B_J^{pk}, the algorithm $Samp(x, B_I, R, B_J)$ is also public known. Its output is an element from a coset $x + I$ which is selected totally uniform. The private key is the basis B_J^{sk}. Also, the plaintext space is $\mathcal{P} \subset R \mod B_I$*

Encrypt(pk, π)**:** *with input the plaintext $\pi \in \mathcal{P}$ chooses randomly a $\psi' \leftarrow Samp(\pi, B_I, R, B_J)$ and returns $\psi \leftarrow \psi' \mod B_J^{pk}$.*

Decrypt(sk, ψ)**:** *the input here is a ciphertext ψ and the output a $\pi \leftarrow (\psi \mod B_J^{sk}) \mod B_I$*

Evaluate(pk, C, Ψ)**:** *the inputs are a circuit C, consisting only of gates Add_{B_I} and $Mult_{B_I}$, and a tuple of ciphertexts Ψ. This algorithm also uses the circuits $Add(pk, \psi_1, \psi_2)$ and $Mult(pk, \psi_1, \psi_2)$ that operate as follows:*

- *$Add(pk, \psi_1, \psi_2)$: returns $\psi_1 + \psi_2 \mod B_J^{pk}$*
- *$Mult(pk, \psi_1, \psi_2)$: returns $\psi_1 \times \psi_2 \mod B_J^{pk}$*

We now show that this scheme is correct. First observe that the circuit C which is an input for the algorithm $Evaluate(pk, C, \Psi)$ has to convert in a circuit which executes operations in the ring R since after the encryption we are in the whole ring R and we are not in cosets of I because of the $\mod B_I$ and $\mod B_J^{pk}$ operations and also because I, J are relatively prime.

We call the circuits that being used for this purpose *generalized* and we denote them by $g(C)$, where C is the initial circuit for operations in cosets of I and $g(C)$ is the respective circuit which operates in R.

With X_{Enc} we denote the image of *Samp* and with X_{Dec} the representative of the cosets $R \mod B_J^{sk}$.

From now on when we say *acceptable circuits* we refer to an element of the following set of circuits:

$$\mathcal{C}'_{\mathcal{E}} = \{C : \forall (x_1, ..., x_t) \in X_{Enc}^t, g(C)(x_1, ..., x_t) \in X_{Dec}\}.$$

That is, they are the circuits whose generalizations with inputs from X_{Enc} return representatives of cosets $R \mod B_J^{sk}$ which means that after operating over their inputs the output remains inside the area that decryption works correctly.

Theorem 21 (Correctness of the Evaluation)**.** *Let $\mathcal{C}_{\mathcal{E}}$ a set of acceptable circuits (that includes the identity gate). Then if ψ is a tuple of ciphertexts for C the scheme $\mathcal{E}$ is correct for inputs Ψ of C.*

Proof. We have that $\Psi = \{\psi_1, ..., \psi_t\}$ with $\psi_k = \pi_k + i_k + j_k$, where $p_k \in \mathcal{P}$, $i_k \in I$, $j_k \in J$ and $\pi_k + i_k \in X_{Enc}$ with $i \in [1, t]$.

Then the following are true:

$$Evaluate(pk, C, \Psi) = g(C)(\Psi) \mod B_J^{pk} \in g(C)(\pi_1 + i_1, ..., \pi_t + i_t) + J$$

but if C is in the set of acceptable circuits $\mathcal{C}_{\mathcal{E}}$ then it maps the elements of X_{Enc} inside X_{Dec} thus, we take:

$$Decrypt(sk, Evaluate^{pk}(C, \Psi)) = g(C)(\pi_1 + i_1, ..., \pi_t + i_t) \mod B_I$$

$$= g(C)(\pi_1, ..., \pi_t) \mod B_I = C(\pi_1, ..., \pi_t)$$

□

Remark that the hardness assumption for the ICP problem relies on the choices that the algorithm $Samp_1$ makes. If $Samp_1$ samples uniformly from R then the ICP problem does not make any sense since in both occasions (for $b = 0$ or for $b = 1$) the resulting element is uniformly sampled from R mod B_J^{pk}. Thus, all we need in order to construct a hard instance of ICP is a biased instantiation of $Samp_1$. This gives us a hint on how to reduct the security of the abstract scheme in the ICP problem.

First we instantiate the $Samp$ algorithm as:

$$Samp(x, B_I, R, B_J) = x + r \times s, \text{ with } r \xleftarrow{R} Samp_1(R)$$

Proposition 8 (Security of the Scheme)**.** *Suppose that $\mathcal{A}$ is an algorithm that efficiently breaks the semantic security of the scheme with advantage ϵ when it uses Samp. Then there is an algorithm $\mathcal{B}$ that solves efficiently the ICP with advantage $\epsilon/2$.*

Proof. Let $\mathcal{A}$ be an algorithm that breaks the semantic security of the above scheme when it uses $Samp$. Since $Samp$ does not samples uniformly from R mod B_J^{pk} a player $\mathcal{B}$ can break the ICP problem with the following strategy.

Let (x, B_J^{pk}) an output of the game for the ICP that is given to $\mathcal{B}$. Now $\mathcal{B}$ has to choose if x is randomly chosen from the ideal which basis is the matrix B_J^{pk}.

When $\mathcal{A}$ sends $\mathcal{B}$ the plaintexts π_1, π_2 then $\mathcal{B}$ randomly sets π to be π_1 or π_2 and sends back to $\mathcal{A}$ the ciphertext ψ computed as:

$$\psi \leftarrow \pi + x \times s$$

Observe now that if $x \leftarrow r \mod B_J^{pk}$ for $r \overset{R}{\leftarrow} Samp_1(R)$ then ψ is a well formed output of the *Encrypt* algorithm. In this case $\mathcal{A}$ has advantage ϵ to discover it and this also is an advantage for $\mathcal{B}$.

But, if the output of $Samp_1$, is a random element of R then t is also a random element taken by the cosets of ideal J then $\mathcal{A}$'s advantage is 0 and this gives on advantage to $\mathcal{B}$.

In first case, $\mathcal{B}$ responds with $b = 0$ and his advantage is ϵ. In second case, $\mathcal{B}$'s answer is $b = 1$ with no advantage for it. The overall advantage of $\mathcal{B}$ is then $\epsilon/2$ □

VII.II The importance of X_{Enc} and X_{Dec}

At this point, we have an almost homomorphic scheme, but none can say that this is what we want to build. Note also that suffices to focus only on the scheme's additive and multiplicative homomorphic structure since the set of gates $\{+, \times\}$ is complete. Consequently, we can start consider only addition and multiplication in order to understand the scheme's homomorphic ability. It is obvious that our construction is not fully homomorphic and that's because we have set up an upper bound on the value of a ciphertext in order to be decrypted correctly. That is, the correctness proof relies on the assumption that every ciphertext lies inside X_{Dec}.

To see why this is necessary, we cite the following trivial example:

Example 7. *Let $I = \langle 5 \rangle$ and $J = \langle 101 \rangle$, our plaintext space then is $\{0, 1, 2, 3, 4\}$ (i.e. the distinguished representatives of $\mathbb{Z}_5$) to encrypt the messages $\pi_1 = 3, \pi_2 = 1$ we call algorithm Samp, suppose that Samp maps π_1, π_2 to the ciphertexts $\psi_1 = 23$ and $\psi_2 = 36$ respectively. Then if one sum up the two ciphertexts the result is $\psi_1 + \psi_2 = 59$ which is equal to $\pi_1 + \pi_2$ after decryption (apply first $\mod 101$ and then $\mod 5$ operations). Thought, for multiplication the same issue does not hold since $\psi_1 * \psi_2 = 828$ which is $20 \mod 101$ and finally the decryption outputs 0 which, of course, is not equal to $\pi_1 \cdot \pi_2$.*

Let's try to figure out now what are the differences between $+$ and $\cdot$ operations that cause this significant discrepancy. On the one hand addition increases the ciphertext but $23 + 36$ is still smaller than X_{Dec} which in our case is 101, on the other hand multiplication produces a larger ciphertext so that it also exceeds X_{Dec}.

The above difference is the reason for this asymmetric behavior of ψ_1, ψ_2 to addition and multiplication respectively. The weakness of multiplication is that the product overheads X_{Dec} and that causes a decryption error since the $\mod 101$ operation produces a result, different than the initial product which had built as $(3 + k_1 \cdot 5) \cdot (1 + k_2 \cdot 5)$ and hopefully after the $\mod 5$ operation everything but $3, 1$ has been eliminated. But, now after multiplication there is no guarantee that the product continues to have the form $(3 + l_1 \cdot 5) \cdot (1 + l_2 \cdot 5)$ since we hold everything that gets through the $\mod 101$ operation.

From the above analysis we have now a visualized sense of procedures that may cause mistaken homomorphic operations. All we need is the X_{Enc} and X_{Dec} to be set so that every operation of elements of X_{Enc} results inside X_{Dec}. Actually, last restrictions have already been set; rearmost requirement is exactly an outline of the acceptable circuits!

In any case, it's infeasible to get fully homomorphic encryption from this abstract scheme; it offers, at most, only a few homomorphic steps of addition and multiplication. But its no so bud, we already have a somewhat homomorphic scheme. Recall now the notion of bootstrappability that we introduced in chapter IV, if we manage to modify the abstract scheme so that it become bootstrappable then our problem is over.

Now is the time that we have define our goals, and we know what we are asking in order to achieve fully homomorphic encryption. In next chapters we have to answer the following questions:

1. For fixed X_{Enc}, X_{Dec} how deep can we evaluate homomorphically in a circuit?

2. How can X_{Enc} and X_{Dec} can be initialized?

3. For how many operations does the scheme works correctly?

4. May we shorten the decryption circuit size to attain bootstrappability?

Chapter **VIII**

Modifying The Scheme

$\boxed{U}$P to this point we have take a taste of a homomoprhic scheme. Remember now that we ended chapter IV by obtaining a trick on how to build fully homomorphic encryption schemes starting by leveled ones. This is the subject of this chapter, we try to modify the scheme to fulfil the requirements that a scheme must have to be bootstrappable. There are two main tasks to complete here, first we have to increase the scheme's -homomorphic- evaluative capacity but in parallel we also have to decrease the scheme's decryption circuit size.

VIII.I Homomorphic depth

In previous section we show a somewhat homomorphic scheme which is able to operate only a limited number of homomorphic steps. Here we describe a formula that counts these steps and also show that the depth depends on the ring that we chose our lattices to based on.

Before start presenting the results, we make a redefinition of X_{Enc} and X_{Dec} in order to give them a more geometrically sense than they have.

Definition 64 (r_{Enc} and r_{Dec})**.** *Denote by r_{Enc} the smallest radius such that $X_{Enc} \subseteq \mathcal{B}(r_{Enc})$ and by r_{Dec} the largest radius such that $\mathcal{B}(r_{Dec}) \subseteq X_{Dec}$*

Define also the set of *permitted circuits* as the following set:

$$\mathcal{C}'_{\mathcal{E}} = \{C : \forall (x_1, ..., x_t) \in \mathcal{B}(r_{Enc})^t, g(C)(x_1, ..., x_t) \in \mathcal{B}(r_{Dec})\}. \qquad \text{(VIII.1)}$$

Hereinafter, our big challenge is to setup r_{Enc} and r_{Dec} to get the more evaluative capacity for arbitrary circuits that we can. As you can see from figure VIII.I if we fix r_{Enc} and r_{Dec} then the sum of two vectors $a, b \in \mathcal{B}(r_{Enc})$ lies inside r_{Dec} more often than their product does. This is because, generally, multiplication has an expansion factor that is bigger than addition's expansion witch is bounded by the triangle inequality. In the same figure we denote by γ_{Mult}, γ_{Mult} the expansion factor of multiplication and of addition respectively. Next we prove that γ_{Mult} depends on the ring that our ideal lattice is based on, for γ_{Add} though the bound only depends on the length of the added in vectors.

Next theorem characterizes the correct evaluation capacity of scheme 4 for arbitrary circuits based on their depth.

Lemma 9. *Let $r_{Enc} \geq 1$ and $r_{Dec} \geq r_{Enc}$ be the closed spheres defined above for scheme 4. Consider also a circuit C, consisting only of $+$ and $\times$ gates, with additive fan-in $\gamma_{Mult}(R)$ and multiplicative fan-in 2. Then the scheme 4 correctly evaluates circuit C implies that C's depth is at most* $\log\log r_{Dec} - \log\log(\gamma_{Mult}(R) \cdot r_{Enc})$.

Proof. Suppose that C has depth d, and denote by r_i an upper bound on the Euclidean norm of the values at level i. We also have that $r_0 = r_{Enc}$. Note now that for every two consecutive levels $i, i+1$ of circuit C it holds that the output of an addition gate of level i is at most $\gamma_{Mult}(R) \cdot r_i$ also the output of a level-i multiplication gate is at most $\gamma_{Mult}(R) \cdot r_i^2$. In either case we take that $r_{i+1} \leq \gamma_{Mult}(R) \cdot r_i^2$ and because C has d levels we have:

$$r_d \leq (\gamma_{Mult}(R) \cdot r_0)^{2^d}$$

Observe now that the scheme 4 correctly evaluates circuit C when: $(\gamma_{Mult}(R) \cdot r_{Enc})^{2^d} \leq r_{Dec}$ this gives the required relation:

$$d \leq \log\log r_{Dec} - \log\log(\gamma_{Mult}(R) \cdot r_{Enc})$$

□

Now, insomuch every circuit may be replaced by one that uses only $+$ and $\times$ gates we can prove a more general result about the depth that scheme 4 is able to evaluate.

Theorem 22. *Scheme 4 correctly evaluates circuits of depth up to*

$$\log\log r_{Dec} - \log\log(\gamma_{Mult}(R) \cdot r_{Enc}).$$

Proof. Direct consequence of the previous lemma 9. □

From the above we get that the expansion factor of multiplication is an important issue towards working on getting a *deep* circuit. This factor though is more flexible from the corresponding factor of addition, since for addition the upper bound of γ_{Mult} is in any ring bounded by the triangle inequality, the same does not hold for γ_{Mult} which depends on the ring. To make it clear, consider the ring $\mathbb{Z}[x]/\langle f(x)\rangle$ with $f(x) = x^n - 2\cdot x^{n-1}$. Then take the polynomials $g_1(x) = x^{n-1} + \langle f(x)\rangle$, $g_2(x) = x^{n-1} + \langle f(x)\rangle$ and $g_3(x) = x^2 + \langle f(x)\rangle$, their product is x^{2n} which, after the reduction modulo $f(x)$ equals to $2^{n+1}\cdot x^{n-1} + \langle f(x)\rangle$. In this case after only two multiplications we have an exponential growth of the coefficient (initially it was 1 in every polynomial but finally becomes 2^{n+1}). So, such a choice for the ring to base an ideal lattice on is not efficient. Nevertheless, not all rings have this bad behavior; one good example is the ring $\mathbb{Z}[x]/\langle f(x)\rangle$ for $f(x) = x^n - 1$.

Lemma 10. *Consider the ring $R[x] = \mathbb{Z}[x]/\langle f(x)\rangle$ with $f(x) = x^n - 1$, then for every $g_1(x), g_2(x) \in R[x]$ it holds that*

$$\|g_1(x)\cdot g_2(x)\| \le \sqrt{n}\cdot\|g_1(x)\|\cdot\|g_2(x)\|$$

Proof. Let $g_1(x), g_2(x) \in R[x]$ and $h(x) = g_1(x)\cdot g_2(x)$, denote by h_i the coefficient of x^i for polynomial $h(x)$ and by a_i, b_i the coefficients of x^i for polynomials $g_1(x)$, $g_2(x)$ respectively. We have that $h_i = \sum_{j=0}^{n-1}(a_i \cdot a_{i-j \mod n})$. Consider now the absolute value for each one of the h_i's. We have

$$|h_i| = |\sum_{j=0}^{n-1}(a_i\cdot b_{i-j \mod n})|, \tag{VIII.2}$$

denote by $g_{3,i}(x)$ the polynomials $\sum_{j=0}^{n-1} b_{i-j \mod n}\cdot x^j$ in particular $g_{3,i}(x)$ are all possible circulant rotated versions of $g_2(x)$. But for every $g_{3,i}(x)$ holds that $\|g_{3,i}(x)\| = \|g_2(x)\|$. Now we can rewrite the equation VIII.2 as:

$$|h_i| = |\sum_{j=0}^{n-1}(a_i\cdot b'_i)| = \sqrt{\left(\sum_{j=0}^{n-1}(a_i\cdot b'_i)\right)^2}, \tag{VIII.3}$$

with b'_i we denote the coefficient of x^i for polynomial $g_{3,i}(x)$ now the last equation after the *Cauchy-Schwarz inequality* gives:

$$|h_i| \leq \sqrt{\sum_{i=0}^{n-1}(a_i)^2} \cdot \sqrt{\sum_{i=0}^{n-1}(b'_i)^2} = \|g_1(x)\| \cdot \|g_2(x)\|. \tag{VIII.4}$$

To end the proof consider now the Euclidean norm of $h(x)$,

$$\|h(x)\| = \sqrt{\sum_{i=0}^{n-1}(h_i)^2} \tag{VIII.5}$$

from the inequality VIII.4 follows that: $\|h(x)\| \leq \sqrt{\sum_{i=0}^{n-1}(\|g_1(x)\| \cdot \|g_2(x)\|)^2}$ which implies $\|h(x)\| \leq \sqrt{n \cdot (\|g_1(x)\| \cdot \|g_2(x)\|)^2} = \sqrt{n} \cdot \|g_1(x)\| \cdot \|g_2(x)\|$ as required. $\square$

Remarkably, theorem 10 gives us a "low" upper bound on the expansion factor of multiplication in the ring $\mathbb{Z}[x]/\langle x^n - 1\rangle$. This bound perfectly fits to our needs and this will be the ring that the ideal lattices of the scheme will be based on.

Previous results provide some tips for achieving the more efficiency from addition and multiplication procedures after showing how we can cram more and more of those gates in order to get the maximum homomorphic evaluations that we can in scheme 4.

VIII.II Instantiating r_{Enc}, r_{Dec}

Recall r_{Enc} and r_{Dec} now, and keep an eye on the figure VIII.I, since we try to increase the set of permitted circuits $\mathcal{C}'_{\mathcal{E}}$ a good idea would be to shrink the ball r_{Enc} and simultaneously increase the ball r_{Dec}, this also can be deduced by equation VIII.1. After this modification, geometrically, there is more space between the r_{Enc} and r_{Enc} so, the result of an addition or multiplication from elements of r_{Enc} is more likely to fall within the ball r_{Dec}. Formally, we want to increase the ratio r_{Dec}/r_{Enc}. Yet, this is not a trivial task because the more we shrink the ball r_{Enc} the more identified it becomes among the cosets of ideal I, thus we may hurt security. Under last restriction we have to take care on how small the ball r_{Enc} could be

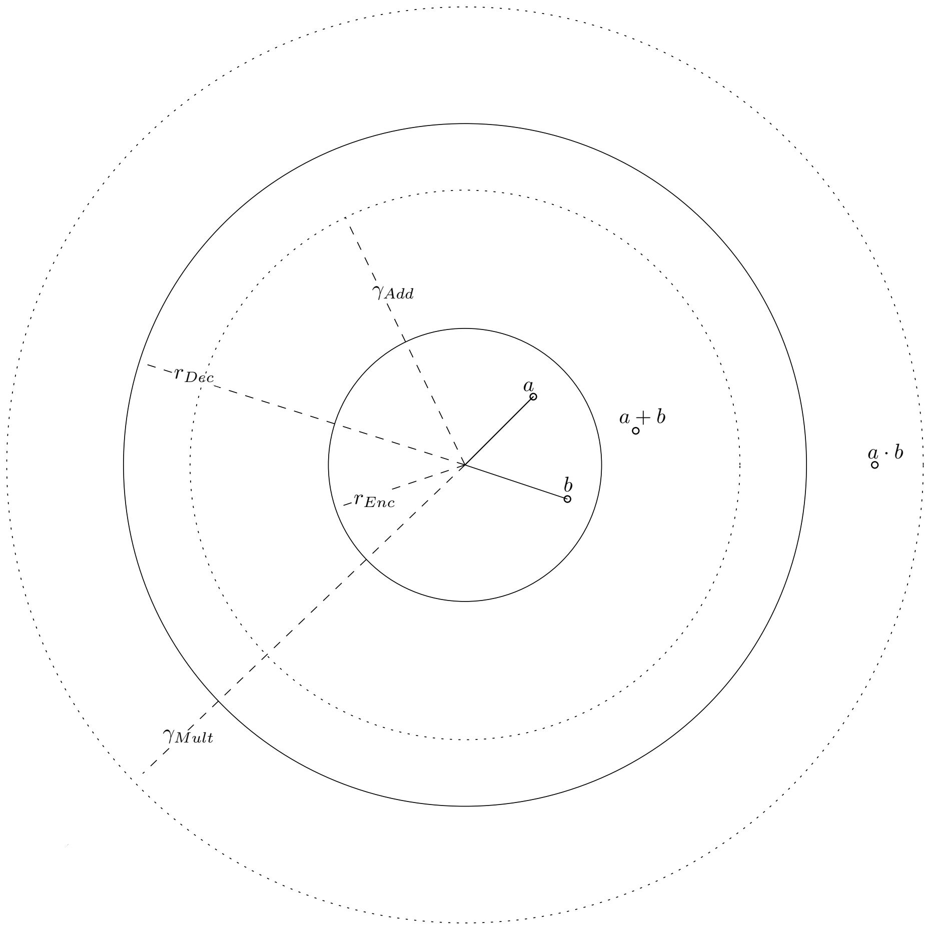

Figure VIII.I: A geometrically view of r_{Enc} and r_{Dec}.

and then we can set it to be smallest possible. Towards trying to specify a perfect size for r_{Enc} we dig up the early definition of $Samp$ made especially for the security proof of scheme 4.

$$Samp(x, B_I, R, B_J) = x + r \times s, \text{ with } r \overset{R}{\leftarrow} Samp_1(R)$$

Now, by definition we have that $r_{Enc} = max\{\|x + r \times s\|\}$ we now get an upper bound for this max.

Lemma 11. *When we use Samp and $Samp_1$ as they instantiated in security proof of proposition 8, and considering also that B_I is the rotation basis corresponding to vector s in the ideal $\mathbb{Z}[x]/\langle x^n - 1\rangle$ then we get that:*

$$r_{Enc} \leq n \cdot \|B_I\| + \ell_{Samp_1} \cdot \|B_I\|$$

where ℓ_{Samp_1} is an upper bound on the length of r.

Proof. By the definition of r_{Enc} and the triangle inequality we immediately have the following inequality: $r_{Enc} \leq max\{\|x\| + \|r \times s\|\}$ but because the vector x is a distinguishable representative for cosets of ideal I we have that $x = B_I \cdot y$, $y = (y_1, y_2, ..., y_n)$ with $|y_i| \leq 1, \forall i \in [1, n]$. Let $B_I = (b_1, b_2, ..., b_n)$, with b_i we denote the column vectors of B_I. Then the i-th coordinate of vector x is at most $|b_{1,i} + b_{2,i} + ... + b_{n,i}|$ thus $|x_i| \leq n \cdot |b_{k,i}|$, where $|b_{k,i}| \geq |b_{t,i}|, \forall t \in [1, n]$. Finally:

$$\|x\| = \sqrt{\sum_{i=1}^{n} |x_i|^2} \leq \sqrt{\sum_{i=1}^{n} (n \cdot |b_{k,i}|)^2} \leq n \cdot \|B_I\| \qquad \text{(VIII.6)}$$

For the other term $r \times s$, we first obtain by that $\|r \times s\| = |r| \cdot \|s\|$, we also have that ℓ_{Samp_1} is an upper bound for the length of r, this implies that

$$\|r \times s\| \leq \ell_{Samp_1} \cdot \|s\|. \qquad \text{(VIII.7)}$$

Also, according to lemma 8 the columns of basis B_I consisted of circular permutations of s we take that

$$\|s\| = \|B_I\| \qquad \text{(VIII.8)}$$

We have the result after combining the three labeled equations. □

Last lemma provides us a nice upper bound on r_{Enc} which is essentially based only on ℓ_{Samp_1}, but as already mentioned we can't let r_{Enc} be arbitrary small because then the ICP problem becomes vulnerable to attacks. But, actually a choice of $\ell_{Samp_1} = n$ makes the entropy of r_{Enc} is sufficiently large to provide security based on ICP. On the other hand according to Babai's Nearest Plane Algorithm we can't let the fraction $\lambda_1(J)/r_{Enc}$ be as large as 2^n because then the scheme becomes vulnerable to attacks that hurt tis semantic security. Nevertheless, as far as we know up to this date if we set up the parameters so that the problem of finding closest lattice vectors needs approximation factor 2^{n^c} for $c < 1$ its practically infeasible for one to solve this instance. Under this observation we can set $r_{Dec} = 2^{n^{c_1}}$ and $\gamma_{Mult} \cdot r_{Enc} = 2^{n^{c_2}}$, for $c_1 - c_2 < 1$. Now, if we set up the parameters this way according to theorem 22 the homomorphic evaluative capacity of the scheme is $(c_1 - c_2) \log n$

We already have a stepping stone for setting up r_{Enc}, now we have to find a way to instantiate the outer ball which is the limit for the homomorphic operations. About r_{Dec} there are no security issues, so we can set it as long as we can. The only restriction for r_{Dec} is to be circumscribed by B_J^{sk}. If we choose to scale up the whole basis B_J^{sk} is equivalent to have a small r_{Enc} and thus it is not a good idea in order to increase r_{Dec}. For fixed I, J the only think one could do to make r_{Dec} as large as possible is to find a basis that, geometrically, is the fattest achievable among all bases of J. Formally, last property described by the following lemma.

Lemma 12. *Let B be a lattice basis and r the radius of the largest sphere centered at* 0 *and circumscribed by $\mathcal{P}(B)$, then* $r = \dfrac{1}{2 \cdot \|(B^{-1})^T\|}$

Proof. Consider a vector x, then $x \in \mathcal{P}(B) \Leftrightarrow x \equiv x \mod B$ but it holds only when $x - B \cdot \lceil B^{-1} \cdot x \rfloor = x$ which means $\lceil B^{-1} \cdot x \rfloor = 0$. Now, to prove the lemma we have to show first that:

$$\forall y : \|y\| < r \Rightarrow \lceil B^{-1} \cdot y \rfloor = 0 \tag{VIII.9}$$

and also that:

$$\forall \varepsilon > 0 \exists y : \|y\| > r + \varepsilon \text{ and } \lceil B^{-1} \cdot y \rfloor \neq 0 \tag{VIII.10}$$

For the first part we consider a vector $y : \|y\| < r$, then $\|y\| < \dfrac{1}{2 \cdot \|(B^{-1})^T\|} \Rightarrow$ $\|y\| \cdot \|(B^{-1})^T\| < 1/2$. Note now that every single coefficient v_i of the vector $v = B^{-1} \cdot y$ is the inner product of the corresponding column of matrix $(B^{-1})^T$ and y. Formally, $v_i = \langle b_i^*, y \rangle$, where b_i^* is the i-th column of matrix $(B^{-1})^T$. Next, we can mull over $|v_i|$, $|v_1| \leq |\langle b_i^*, y \rangle|$, by Cauchy-Schwartz

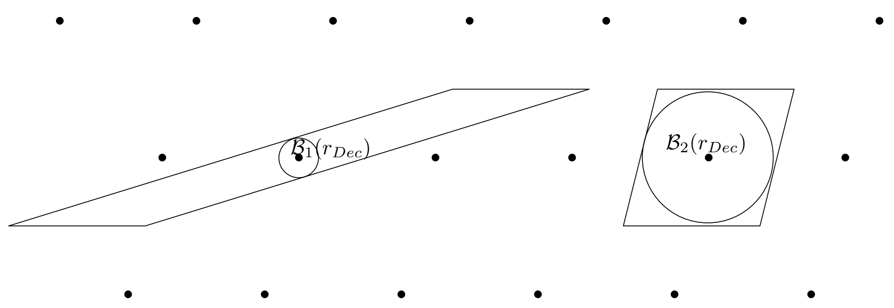

Figure VIII.II: Different bases, different r_{Dec}.

inequality $|v_1| \leq \|b_i^*\| \cdot \|y\|$ but, since $\|b_i^*\| \leq \|(B^{-1})^T\|$ we conclude that $|v_1| \leq \|(B^{-1})^T\| \cdot \|y\|$ which combined with $\|y\| \cdot \|(B^{-1})^T\| < 1/2$ implies that $|v_i| < 1/2 \Rightarrow \lceil v \rfloor = 0 \Rightarrow \lceil B^{-1} \cdot y \rfloor = 0$ as required.

Let $\varepsilon > 0$, consider the vector $y : \|y\| = r + \varepsilon$ which is also parallel to the longest column vector, say b_k of matrix $(B^{-1})^T$. $\|y\| = r + \varepsilon$ implies $\|y\| > r$ which means $\|y\| \cdot \|(B^{-1})^T\| > 1/2$. Furthermore, because y is parallel to b_k we have that $|\langle y, b_k \rangle| = \|y\| \cdot \|b_k\| = \|y\| \cdot \|(B^{-1})^T\| > 1/2$, thus the k-th coordinate of the vector $B^{-1} \cdot y$ has magnitude bigger than 1/2, consequently $\lceil B^{-1} \cdot y \rfloor \neq 0$ □

Direct consequence of the last lemma is the following lemma.

Lemma 13. *For the scheme 4 the following equality holds:*

$$r_{Dec} = \frac{1}{2 \cdot \|((B_J^{sk})^{-1})^T\|}$$

VIII.III Achieving Bootstrappability

Starting from scheme 4 we have managed to efficiently instantiate each of its algorithms without hurting its security. This modified scheme is almost homomorphic, until this point it offers correct homomorphic evaluation for deep enough circuits, but this does not hold for arbitrary circuits. In previous subsection we show that the depth of clear evaluation that one can get is about $\log n$. The final step in order to obtain a fully homomorphic scheme is then to implement the theory of chapter IV, that is, we have to manage to shorten the decryption circuit's depth. A fully homomorphic encryption scheme after swallowing the circuit is feasible by implementing the generic construction of chapter IV in our scheme.

In this section our main goal is to modify the decryption circuit in order to implement the above. We first refitting the decryption algorithm as in a way that affects the key generation algorithm also. The main concept behind the alteration is to give the decrypter a ciphertext that is as easy to be decrypted as possible, we bring off such a task by forcing the encrypter to start operating on the data on behalf of the decrypter.

Algorithm 5 (SplitKey(sk, pk))**.**

- *Extracts* B_J^{sk} *from* sk
- τ *is a set of* $\delta(n)$ *uniformly random matrices* $B_1, B_2, ..., B_{\delta(n)}$. *There is an ordered subset* S *of* $[1, \delta(n)]$ *consisting of* $\delta_{sub}(n)$ *elements, such that*
$$\sum_{i \in S} B_i = {B_J^{sk}}^{-1}$$
- sk' *is a square matrix* M *with dimensions* $\delta_{sub}(n) \times \delta_{sub}(n)$:
$$M_{i,j} = \begin{cases} 1 & \text{,if } j \text{ is the } i\text{-th member of } S \\ 0 & \text{,otherwise} \end{cases}$$
- *Outputs* (sk', τ)

Algorithm 6 (ExpandCT(pk, ψ))**.**

- *Outputs* $c_i \leftarrow B_i \cdot \psi \mod B_I$, *for* $i \in [1, \delta(n)]$

Algorithm 7 (Decrypt(sk, ψ))**.**

- *Set the vectors* $w_{i,j} = M_{i,j} \cdot c_j$.
- *Set* $x_i = \displaystyle\sum_{j=1}^{\delta(n)} w_{i,j}$.
- *Generate* $\delta_{sub}(n)+1$ *integer vectors* $y_1, y_2, ..., y_{\delta_{sub}(n)+1}$ *with sum* $\left\lceil \displaystyle\sum_{i=1}^{\delta_{sub}(n)} x_i \right\rfloor$.
- *Compute* $\pi = \psi - B_I \cdot \left(\displaystyle\sum_{i=1}^{\delta_{sub}(n)+1} y_i \right) \mod B_I$

After describing the two new algorithms its obvious that the modified decryption algorithm is useless for the scheme 4 since the form of of the keys and ciphertexts differentiate from those the last algorithm can decrypt. We will present the final scheme in the next chapter, where we also show that this above approach is secure so that it can be adapted to previous scheme smoothly.

In order to make the scheme bootstrappable our requirement is a low depth decryption circuit. Recall now the decryption algorithm and the operations needed to be done. We first focus on how to efficiently generate $n+1$ integer vectors with the same sum to the rounded of previous n vectors. Surprisingly this is a demanding task and lead us to reconsider some parts of the system in order to achieve low depth decryption circuit.

There is the classic Karp's algorithm presented in [15] for obtaining $n+1$ integer vectors from n vectors having rounded the same sum. This algorithm is also known as the "3 for 2 trick" on account of the way it approaches the solution. In breath, the algorithm takes three numbers and then replaces them by two integers having the same sum, this is an operation that takes place once in every single level of the circuit and consequently we need $\log_{3/2} n$ depth for its operation. The problem that causes such a depth is that for every number we cannot eliminate none of its bits because it is possible for the least significant bit of a number to affect the most significant bit of the sum. But even if we have the promise that every x_i is sufficiently close to an integer it is not possible to create a circuit with depth much smaller that $\log n$ which means that our scheme will be inefficient after all those decryption calls.

Next theorem implies a strategy for computing the problematic rounded sum in a low complexity circuit:

Lemma 14. *Consider t real numbers $a_1, a_2, ..., a_t$, the binary representation for each of these numbers, $a_i = ... + a_{i,1} \cdot 2^1 + a_{i,0} \cdot 2^0 + a_{i,-1} \cdot 2^{-1} + ...$, and suppose that if A_i is the closest integer number to a_i then holds that $|A_i - a_i| \leq 1/4$. We can create a $\mod B_I$ circuit that generates $t+1$, represented in binary, integer numbers whose sum is $\left\lceil \sum_{i=1}^{n} a_i \right\rfloor$, such that if its inputs are in $\mathcal{B}(r_{in})$ the its outputs are in $\mathcal{B}((\gamma_{Mult}(R) \cdot n \cdot \|B_I\| \cdot (1 + \gamma_{Mult(R)} \cdot r_{in})^t \cdot t)^{polylog(t)})$.*

Proof. The advantage that causes the efficiency of this method is that we can eliminate all but $\lfloor \log t \rfloor + 2$ from every a_i without break down their rounded sum. Indeed, let $T = \lfloor \log t \rfloor + 2$ and $a_i' = ... + a_{i,1} \cdot 2^1 + a_{i,0} \cdot 2^0 +$

$a_{i,-1} \cdot 2^{-1} + ... + a_{i,-T} \cdot 2^{-T}$, then $\left| \sum_{i=1}^{t} a'_i - \sum_{i=1}^{t} a_i \right| = \left| \sum_{i=1}^{t} \sum_{j=T}^{+\infty} 2^{-j} \cdot a_{i,-j} \right| < \frac{1}{4}$ because all a_i's are at most 1/4 far from an integer.

The first n numbers that the algorithm returns are every integer part of the a_i's and then the final integer to be returned is the rounded sum, say S, of their fractional parts witch now has only T elements for each number. Consider now the T vectors $b_i = (a_{1,-i}, a_{2,-i}, ..., a_{t,-i})$, for $i = 1$ to T. Let c_i be the Hamming weight of vector b_i we have that $S = \sum_{j=1}^{T} 2^{-j} \cdot c_i$. Now if we put one more constraint to our hypotheses then this sum can be computed efficiently. Our extra requirement is to restrict the plaintext space to $\mathcal{P} = \{0, 1\} \mod I$ in order to be able to use the property* of symmetric elementary polynomials that gives us the binary representation of Hamming weights. This is because the known algorithms for computing such sums work only in $\mathbb{Z}_2$ which also implies that, for convenience, it would be helpful to instantiate the ideal I to represent $\mathbb{Z}_2$ inside n dimensions, that is we set up I as $(2 \cdot e_i)$, where with e_i we denote the vector witch is all zero but its i-th position has 1. After these assumptions we have that $c_i = (e_{2^{\lceil \log t \rceil}}(b_{i,1}, b_{i,2}, ..., b_{i,t}) \mod 2, ..., e_{2^0}(b_{i,1}, b_{i,2}, ..., b_{i,t}) \mod 2)$ which essentially gives us the final information to compute S. The time complexity for the computation each of these polynomials is $poly(n)$ and because every polynomial among $e_{2^0}, ..., e_{2^{\lceil \log n \rceil}}$ has a degree upper bounded by t it causes the exponential factor $polylog(t)$. □

The above algorithm provides us with a strong and efficient algorithm that is a solution to the problem of computing the rounded sum of n real numbers but as we have already seen it causes an inconvenience in the whole schema, since it limits the plaintext space to $\{0, 1\}$ which means that we lose an advantage of lattices which is a huge plaintext space within low dimensions and small bases. Up to this date the problem of finding a scheme that utilizes the whole basis as plaintext space remains open. Another constraint that being introduced after the above proof is the instantiation of B_I to be the full of zeros except one position where it has 2 instead of 0.

Lemma 14 implies for our choice of parameters that after the computation of the rounded sum of t real numbers the results lie in a sphere of radius $(\sqrt{n} \cdot r_{Enc})^{t+polylog(t)}$.

*A detailed proof is on appendix A

Lemma 15. *Consider t real numbers $a_1, a_2, ..., a_t$, the binary representation for each of these numbers, $a_i = ... + a_{i,1} \cdot 2^1 + a_{i,0} \cdot 2^0 + a_{i,-1} \cdot 2^{-1} + ...$, and suppose that if A_i is the closest integer number to a_i then holds that $|A_i - a_i| \leq 1/4$. Suppose also that we have set up the ring of our scheme to be $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$, B_I a matrix full of zeros except one position where it has 2 instead of 0 and the plaintext of the scheme to be $\{0, 1\}$ Then, we can create a $\mod B_I$ circuit that generates $t+1$, represented in binary, integer numbers whose sum is $\left\lceil \sum_{i=1}^{n} a_i \right\rfloor$, such that if its inputs are in $\mathcal{B}(r_{Dec})$ the its outputs are in $\mathcal{B}((\sqrt{n} \cdot r_{Enc})^{t+polylog(t)})$.*

Last theorem also indicates that the modification of the decryption circuit using the SplitKey algorithm is necessary since in the initial scheme there were n numbers to be added for the computation of the problematic rounded sum. This implies that $r_{Dec}/r_{Enc} \geq r_{out}/r_{in} \geq 2^n$ which, as we have already seen, is a bad instantiation of r_{Dec}, r_{Enc} because of the existence of Nearest Plane Algorithm.

After all the above combine the last lemma with the the proof of lemma 9, then we get the next theorem.

Theorem 23. *The scheme is bootstrappable when we set*

$$\delta_{sub}(n) \cdot \log^{c_1} \delta_{sub}(n) \leq \frac{\log(r_{Dec}/2)}{2^{c_2} \cdot log(\gamma_{Mult}(R) \cdot r_{Enc}}$$

where $\log^{c_1} \delta_{sub}(n)$ is a polylog term and c_2 is a constant representing the depth needed in a circuit having Add_{B_I} gates with $\gamma_{Mult}(R) = n^{\Omega(1)}$ fan-in and $Mult_{B_I}$ gates with constant fan-in.

We close this chapter by obtaining one more instantiation, this time for $\delta_{sub}(n)$. Suppose that $\gamma_{Mult}(R) \cdot r_{Enc}$ is polynomial in n, and $r_{Dec} = 2^{n^C}$ for $C < 1$. In this case, $\delta_{sub}(n)$ can be sub-linear polynomial in n.

Chapter **IX**

Fully Homomorphic Scheme

AFTER chapter IV where we first show that we can develop a fully homomorphic scheme without the need of an ad-hoc fully homomorphic encryption function we start building a homomorphic scheme step by step. Our first approach was the initial scheme that we present in chapter VII which followed by various modifications conducive to botstrappability of the scheme. In this chapter we demonstrate the resulting fully homomorphic scheme with every instantiation that has to be taken into account. In the second part of this chapter we discuss about the security of the scheme after the modifications that we make.

IX.I The Final Scheme

Before start presenting each algorithm of the final scheme we have to declare that we use algorithms that defined either in abstract scheme or during the modifications in previous chapter. To avoid confusion we denote by $\mathcal{E}^*$ the abstract homomorphic scheme and by $\mathcal{E}$ the final scheme.

Setting up the Parameters

1. Set the degree, n, of the polynomial $f(x) = x^n - 1$ according to λ and $\delta_{sub}(n)$ sub-linear polynomial in n.

2. Set B_I to be a matrix with small norm.

3. Set $\ell_{Samp} \approx n$.

4. Set r_{Enc} to be the maximum possible such that:
$$r_{Enc} \leq n \cdot \|B_I\| + \ell_{Samp_1} \cdot \|B_I\|.$$

5. Set r_{Dec} so that:
$$\delta_{sub}(n) \leq \frac{\log(r_{Dec}/m)}{\alpha \cdot 2^c \cdot \log(\gamma_{Mult}(R) \cdot r_{Enc})}.$$

6. Compute an ideal J relational prime to $I = \langle f(x) \rangle$ with the rotation basis B_J^{sk} of a polynomial $h_2(x) \in J$ so that:
$$r_{Dec} \approx \frac{1}{2 \cdot \|((B_J^{sk})^{-1})^T\|}.$$

7. Set as B_J^{pk} the HNF of B_J^{sk}

8. Plaintext space: $\mathcal{C} = \{0, 1\}$ for every polynomial $g(x) \in \mathbb{Z}[x]/\langle f(x) \rangle$ let s be the sum of its coefficients, then $g(x)$ encodes 0 or 1 if $s \equiv 0 \mod 2$ or $s \equiv 1 \mod 2$ respectively.

A Fully Homomorphic Encryption Scheme

```
$KeyGen_{\mathcal{E}}(\lambda)$
    1. $(pk^*, sk^*) \leftarrow KeyGen_{\mathcal{E}^*}(\lambda)$
    2. $(sk, \tau) \leftarrow SplitKey_{\mathcal{E}}(sk^*, pk^*)$
    3. return $\{sk, (pk^*, \tau)\}$

$Encrypt_{\mathcal{E}}(pk, \pi)$
    1. $\psi^* \leftarrow Encrypt_{\mathcal{E}^*}(pk^*, \pi)$
    2. $\psi' \leftarrow ExpandCT_{\mathcal{E}}(pk, \psi^*)$
    3. return $\psi = \{\psi^*, \psi'\}$

$Decrypt_{\mathcal{E}}(sk, \psi)$
    1. Set the vectors $w_{i,j} = M_{i,j} \cdot \psi'_j$ and $x_i = \sum_{j=1}^{\delta(n)} w_{i,j}$.
    2. Generate $\delta_{sub}(n) + 1$ integer vectors $y_1, y_2, ..., y_{\delta_{sub}(n)+1}$ with sum $\left\lceil \sum_{i=1}^{\delta_{sub}(n)} x_i \right\rfloor$.
    3. Compute $\pi = \psi^* - B_I \cdot \left( \sum_{i=1}^{\delta_{sub}(n)+1} y_i \right) \mod B_I$

$Add_{\mathcal{E}}(pk, \psi_1, \psi_2)$
    1. Extracts $(\psi_1^*, \psi_2^*)$ from $(\psi_1, \psi_2)$
    2. $\psi^* \leftarrow Add_{\mathcal{E}^*}(pk^*, \psi_1^*, \psi_2^*)$
    3. $\psi' \leftarrow ExpandCT_{\mathcal{E}}(pk, \psi^*)$
    4. return $\psi = \{\psi^*, \psi'\}$

$Mult_{\mathcal{E}}(pk, \psi_1, \psi_2)$
    1. Extracts $(\psi_1^*, \psi_2^*)$ from $(\psi_1, \psi_2)$
    2. $\psi^* \leftarrow Mult_{\mathcal{E}^*}(pk^*, \psi_1^*, \psi_2^*)$
    3. $\psi' \leftarrow ExpandCT_{\mathcal{E}}(pk, \psi^*)$
    4. return $\psi = \{\psi^*, \psi'\}$
```

IX.II Security

This section is entirely devoted to security of the whole scheme as it comes from its different parts. But before all recall the ICP on which we based the abstract scheme's security. The whole construction is a modification of that scheme, consequently its security must be based on a similar problem. The main difference is the use of lattices instead of rings, but ideal lattices essentially is just another interpretation of ideal rings. For completeness we define a, hard, lattice problem analogous to the ICP.

Definition 65 (Bounded Distance Decoding Problem for Ideal Lattices (BDDP))**.** *Consider the ring $R = \mathbb{Z}/\langle f(x)\rangle$, the algorithm $Samp_1$ that efficiently samples from $\mathbb{Z}^n$ and also let $IdealGen$ is an algorithm that produces a basis of an ideal in R. We define the bounded distance decoding problem problem to be the following game: The first player chooses a $b \overset{R}{\leftarrow} Samp_1(R)$ and two bases $(B_J^{sk}, B_J^{pk}) \overset{R}{\leftarrow} IdealGen(R, B_I)$. If $b = 0$ he chooses $r \overset{R}{\leftarrow} Samp_1(R)$ and $t \leftarrow r \mod B_J^{pk}$. If $b = 1$ he chooses randomly a t from $R \mod B_J^{pk}$. The second player has to find b given (t, B_J^{pk}).*

The security proof, based on BDDP, for the final scheme is akin to the proof for the ring based scheme. The hardness of BDDP depends on $Samp_1$ algorithm as in the case of rings because we need the entropy of its output to be sufficient big inside the lattice to prevent it from being easily identified among the points of the lattice. Another subject that affects the hardness of BDDP is the length of the shortest vector of the lattice J.

But after all the alterations that we have made to the initial scheme the above security proof does not suffice. We also have to show that the new decryption method does not hurt security. The problem that we based on is the SplitKey Distinguishing Problem which refers to a game similar to the key generation algorithm.

Definition 66 (SplitKey Distinguishing Problem (SDP))**.** *The challenger sets $(sk, pk) \leftarrow KeyGen_{\mathcal{E}}$ and $b \leftarrow \{0, 1\}$. If $b = 0$, it sets $(sk, \tau) \leftarrow SplitKey(sk, pk)$. If $b = 1$, it sets $(sk, \tau) \leftarrow SplitKey(\perp, pk)$, where $\perp$ is a special symbol. The problem: guess b given (τ, sk, pk).*

Theorem 24. *Let $\mathcal{A}$ be an algorithm that breaks the semantic security of $\mathcal{E}$ with advantage ϵ. Then there exist algorithms $\mathcal{B}_0$, $\mathcal{B}_1$ such that either $\mathcal{B}_0$'s advantage against the SplitKey Distinguishing Problem or $\mathcal{B}_1$'s advantage against the semantic security of $\mathcal{E}^*$ is at least $\epsilon/3$.*

Proof. Next we consider two games between the challenger and $\mathcal{A}$. Let Game 0 be the classic semantic security game, and Game 1 is like Game 0, except the challenger generates pk differently. Instead of inputting sk into SplitKey, it inputs $\perp$ to obtain τ , and adds τ to the pk it sends to $\mathcal{A}$. By assumption, is $\mathcal{A}$s advantage in Game 0. Let be $\mathcal{A}$'s advantage in Game 1. $\mathcal{B}_0$ runs as follows. The challenger sets bit $b \leftarrow \{0,1\}$ and sends a SplitKey Distinguishing Problem instance (τ, sk, pk) to $\mathcal{B}_0$. $\mathcal{B}_0$ sends $pk \leftarrow (pk, \tau)$ to $\mathcal{A}$. When $\mathcal{A}$ asks for a challenge ciphertext on one of (π_0, π_1), $\mathcal{B}_0$ sets $\beta \leftarrow \{0,1\}$ and sends $\psi \leftarrow Encrypt_{\mathcal{E}}(pk, \pi_\beta)$. After all, $\mathcal{A}$ sends back a bit β. $\mathcal{B}_0$ sends $b \leftarrow \beta \oplus \beta$ to the challenger. Note that the public key pk is distributed exactly as in Game b. We compute that $\mathcal{B}_0$'s advantage is at least $|\epsilon - \epsilon'|/2$. $\mathcal{B}_1$ runs as follows. It receives an $\mathcal{E}^*$ public key pk from the challenger. Then it runs $(sk, \tau) \leftarrow SplitKey(\perp, pk)$ and sends $pk \leftarrow (pk, \tau)$ to $\mathcal{A}$. When $\mathcal{A}$ asks for a challenge ciphertext on one of (π_0, π_1), $\mathcal{B}_1$ asks the challenger for a challenge ciphertext on one of (π_0, π_1). The challenger sends back ψ . $\mathcal{B}_1$ sets ψ to include ψ^* and the output of $ExpandCT_{\mathcal{E}}(pk, \psi^*)$ and sends ψ to $\mathcal{A}$. $\mathcal{A}$ sends a bit b , which $\mathcal{B}_1$ forwards to the challenger. We see that the distribution is the same as in Game 1. Also, $\mathcal{B}_1$'s bit is correct if $\mathcal{A}$'s bit is correct; so $\mathcal{B}_1$ has advantage ϵ'.

□

Part III

Appendices

Appendix ***A***

Algebra

I.I Symmetric Elementary Polynomials

Definition 67. *A multivariate polynomial P on variables $x_1, x_2, ..., x_n$ called symmetric if there is no permutation of its variables that changes the polynomial.*

An alternative, but more practical, definition would be:

Definition 68. *We call* elementary symmetric polynomial *on n variables the polynomials e_i defined as:*

- $e_0(x_1, x_2, ..., x_n) = 1$
- $e_1(x_1, x_2, ..., x_n) = x_1 + x_2 + ... + x_n$
- $\vdots$
- $e_k(x_1, x_2, ..., x_n) = \sum_{1 \le a_1 < ... < a_k \le n} x_{a_1} \cdot x_{a_2} \cdots x_{a_k}$
- $\vdots$
- $e_n(x_1, x_2, ..., x_n) = x_1 \cdot x_2 \cdots x_n$

Elementary symmetric polynomials will soon be our key to construct the preferred shallow circuit. We also fall over the notion of *Hamming weight* which is momentary presented.

Definition 69. *Let x be a vector of n elements in $\mathbb{Z}_2$, we define the* Hamming weight of x *to be the number of non-zero coordinates of x.*

Surprisingly, there is a tight relationship between Hamming weight and symmetric elementary polynomials since for a vector $(x_1, x_2, ..., x_n)$, $x_i \in \mathbb{Z}_2$ the binary presentation of its hamming weight is:

$$(e_{2^{\lceil \log n \rceil}}(x_1, x_2, ..., x_n) \mod 2, ..., e_{2^0}(x_1, x_2, ..., x_n) \mod 2)$$

Towards the direction of proving the above we give the following fundamental theorem established by Adrien-Marie Legendre in 1808.

Theorem 25. *Let p prime and $a \in \mathbb{N}$, consider the p-adic representation of a, $a = a_k \cdot p^k + a_{k-1} \cdot p^{k-1} + ... + a_1 \cdot p + a_0$, $p^k \leq a < p^{k+1}, 0 \leq a_i \leq p-1$. Then if m is the biggest value of p that divides $a!$ (i.e. $p^m \mid a!$ but $p^{m+1} \nmid a!$) it holds that*

$$m = \sum_{i=0}^{\infty} \left[\frac{a}{p^i}\right] = \frac{a - (a_0 + a_1 + ... + a_k)}{p-1}$$

Proof. Let a, p, m as stated in the theorem, then $a! = p^m \cdot b$, where b is an integer such that $p \nmid b$. Now, by the Euclidean algorithm, there exists integers p_1, r_1 satisfying

$$a = q_1 \cdot p + r_1, \text{ with } 0 \leq q_1 \text{ and } 0 \leq r_1 < p, \tag{A.1}$$

namely, q_1 is the quotient of the division of a by p, that is, $q_1 = \left[\frac{a}{p}\right]$.

By the relation A.1 we obtain that all possible positive multiples of p smaller than a are the numbers: $p, 2 \cdot p, ..., q_1 \cdot p$, which implies that $p \cdot 2 \cdot p \cdots q_1 \cdot p = p^m \cdot b'$ for some integer $b' : p \nmid b'$, equivalently $p^{q_1} \cdot (q_1!) = p^m \cdot b'$. Let m_1 be the biggest integer such that $p^{m_1} \mid q_1!$ then the last gives that $q_1 + m_1 = m$.

If we repeat the above process for $q_1 = q_2 \cdot p + r_2$ we take that $m_1 = m_2 + q_2$, where m_2 be the biggest integer such that $p^{m_2} \mid q_2!$. Inductively, we have that $m = q_1 + q_2 + ... + q_i + ...$, where $q_i = \left[\frac{a}{p^i}\right]$ and m_i that by definition is the biggest integer such that $p^{m_i} \mid q_i!$.

After the above we take that

$$m = \sum_{i=0}^{\infty} q_i = \sum_{i=0}^{\infty} \left[\frac{a}{p^i}\right]. \tag{A.2}$$

But we also have that

$$\begin{aligned}
\left[\frac{a}{p}\right] &= a_k \cdot p^{k-1} + \cdots + a_1, \\
\left[\frac{a}{p^2}\right] &= a_k \cdot p^{k-2} + \cdots + a_2, \\
&\vdots \\
\left[\frac{a}{p^k}\right] &= a_k.
\end{aligned}$$

Combining the last with the equality A.2 the following comes out

$$\begin{aligned}
m &= (a_k \cdot p^{k-1} + \cdots + a_1) + (a_k \cdot p^{k-2} + \cdots + a_2) + \cdots + (a_k) \\
&= a_k \cdot (p^{k-1} + \cdots + 1) + a_{k-1} \cdot (p^{k-2} + \cdots + 1) + \cdots + a_2(p+1) + a_1 \\
&= \frac{1}{p-1}\Big(a_k \cdot (p^k - 1) + a_{k-1} \cdot (p^{k-1} - 1) + \cdots + a_2 \cdot (p^2 - 1) + a_1 \cdot (p-1)\Big) \\
&= \frac{1}{p-1}\Big(a_k \cdot p^k + a_{k-1} \cdot p^{k-1} + ... + a_1 \cdot p + a_0 - (a_k + a_{k-1} + \cdots + a_1 + a_0)\Big) \\
&= \frac{1}{p-1}\left(a - (a_k + a_{k-1} + \cdots + a_1 + a_0)\right)
\end{aligned}$$

as required. □

Direct consequence of last theorem is the following corollary that instantiates the theorem for $p = 2$.

Corollary 1. *Let $a \in \mathbb{N}$, and $a_k \cdot 2^k + a_{k-1} \cdot 2^{k-1} + ... + a_1 \cdot 2 + a_0$ its binary representation. Then if m is the highest power of 2 that divides $a!$ it holds that*

$$m = a - (a_0 + a_1 + ... + a_k)$$

We continue by establishing a useful lemma for

Theorem 26. *Let $a, b \in \mathbb{Z}_2$, with $a > b$, consider also the mapping $h : \mathbb{Z}_2 \leftarrow \mathbb{N}$ defined as*

$$h(x) = \text{ the number of 1's in the binary representation of } x$$

then for arbitrary numbers a, b, represented both in binary, the following equality holds:

$$h(a-b) = h(a) - h(b) + borrows(a-b)$$

by $borrows(a-b)$ we denote the number of borrows when subtracting b from a.

Proof. We will prove the above by induction in the bit size of a, b. First we assume without loss of generality that the numbers a and b represented by the same number of bits (otherwise fill with 0's the shortest of them).

First, easily verified that the proposition holds when a and b are at most 2 bits long.

Assume now that it also holds for integers of size up to k bits, we will prove that the same holds for $(k+1)$-bit integers. □

Theorem 27. *Consider two natural numbers n, k, $n \geq k$, then $\binom{n}{k} = 1 \mod 2$ iff there are no borrows when subtracting k from n in binary.*

Proof. For convenience denote by $borrows(a-b)$ the number of borrows appearing during subtracting the binary number b from the binary number a and by $ones(n)$ the number of 1's in the binary representation of n, also $t(n)$ represents the highest power of 2 that divides n.

First from corollary 1 we get that $t(n) = n - ones(n)$, consequently:

$$t(\binom{n}{k}) = t(e!) - t(k!) - t((n-k)!) = (n - t(n)) - (k - t(k)) - (n - k - t(n-k))$$

the last implies that:

$$t(\binom{n}{k}) = ones(k) + ones(n-k) - ones(n) \tag{A.3}$$

But, we also know that $ones(a-b) = ones(a) - ones(b) - borrows(a-b)$, this gives that

$$ones(n-k) - ones(n) = borrows(n-k) - ones(k) \tag{A.4}$$

From A.3 and A.4 we get that:

$$t(\binom{n}{k}) = borrows(n-k)$$

which is a generalization of the required relation. □

Now we can present the main theorem of this section used in chapter VIII to achieve an efficient algorithm for addition of many numbers.

Theorem 28. *The i-th bit of the binary representation of Hamming weight of a binary vector b equals to $e_{2^{i-1}}(b_1, b_2, ..., b_n) \mod 2$.*

Proof. Observe that if the Hamming weight of b equals to w then by the definition of symmetric elementary polynomials we have that

$$e_{2^{i-1}}(b_1, b_2, ..., b_n) = \sum_{1 \le a_1 < ... < a_{2^{i-1}} \le n} (b_{a_1} \cdot b_{a_2} \cdots b_{a_{2^{i-1}}}) = \binom{w}{2^{i-1}}.$$

According to previous theorem though, $e_{2^{i-1}}(b_1, b_2, ..., b_n) = 1 \mod 2$ iff the i=th bit of the binary representation of w is 1. □

I.II Quadratic Residues

Definition 70. *Let $m > 1$ be a natural number and a an element in $\mathbb{Z}_m$, we say that a is a* quadratic residue modulo m *if there exists $x \in \mathbb{Z}_p$ with $x^2 \equiv a \mod p$.*

In this case, we call x a square root of a. An element that is not a quadratic residue is called a quadratic non-residue.

Lemma 16. *For every prime $p > 2$ it holds that in $\mathbb{Z}_p$ every quadratic residue has exactly two square roots.*

Proof. Take an element $y \in \mathbb{Z}_p$ and suppose that y is a quadratic residue. Then, by definition, $\exists x \in \mathbb{Z}_p$ such that $x^2 \equiv y \mod p$. But we also have that, $(-x)^2 \equiv x^2 \mod p$, so $-x$ is a square root of y. Because p is odd, $-x \not\equiv x \mod p$. So, y has at least two square roots.

Let $x' \in \mathbb{Z}_p$ be a square root of y different than $x, -x$. Then $x^2 \equiv y \equiv (x')^2 \mod p$ implying that $x^2 - (x')^2 = 0 \mod p$. Factoring the left-hand we obtain $(x - x')(x + x') \equiv 0 mod p$, so that either $p \mid (x - x')$ or $p \mid (x + x')$.

In the first case, $x' \equiv x \mod p$ and in the second case $x' \equiv -x \mod p$, showing that y indeed has only $x, -x$ as square roots □

Direct consequence of last theorem is the following corollary.

Corollary 2. *The number of quadratic residues modulo a prime $p > 2$ is exactly $\frac{p-1}{2}$.*

Definition 71 (Jacobi Symbol)**.** *Let $p > 2$ be a prime, and $x \in \mathbb{Z}_p$, we define $J_p(x)$, the Jacobi symbol of x modulo p, as follows.*

$$J_p(x) = \begin{cases} 1 & \text{if } x \text{ is a quadratic residue modulo } p \\ -1 & \text{if } x \text{ is not a quadratic residue modulo } p \end{cases}$$

Lemma 17. *Let $p > 2$ be a prime. Then $J_p(x) = x^{\frac{p-1}{2}} \mod p$.*

Proof. Consider an arbitrary generator g of $\mathbb{Z}_p \setminus \{0\}$. If $x = g^i$ for some even integer i is a quadratic residue. Writing $i = 2j$ with j an integer we have:

$$x^{\frac{p-1}{2}} = (g^{2\cdot j})^{\frac{p-1}{2}} = g^{(p-1)\cdot j} = (g^{p-1})^j = 1^j = 1 \mod p$$

and so $Jp(x) = 1 = x^{\frac{p-1}{2}} \mod p$ as claimed.

On the other hand if x is not a quadratic residue then $x = g^i$ for some odd integer i. Writing $i = 2j + 1$ with j an integer we have

$$x^{\frac{p-1}{2}} = (g^{2\cdot j+1})^{\frac{p-1}{2}} = (g^{2\cdot j})^{\frac{p-1}{2}} \cdot g^{\frac{p-1}{2}} = g^{\frac{p-1}{2}} \mod p.$$

Now, $(g^{\frac{p-1}{2}})^2 = g^{p-1} = 1 mod p$, and so $g^{\frac{p-1}{2}} = 1 mod p$ or $g^{\frac{p-1}{2}} = -1 mod p$ since both $1, -1$ are the two square roots of 1. Since g is a generator it has order $p - 1$ and so $g^{\frac{p-1}{2}} \neq 1 \mod p$. It follows that $J_p(x) = -1 = x^{\frac{p-1}{2}} \mod p$. □

Lemma 18. *Let $p > 2$ be a prime, and $x, y \in \mathbb{Z}_p$, then $J_p(x \cdot y) = J_p(x) \cdot J_p(y)$.*

Proof. Using the previous lemma, $J_p(x \cdot y) = (x \cdot y)^{\frac{p-1}{2}} = x^{\frac{p-1}{2}} \cdot y^{\frac{p-1}{2}} = J_p(x) \cdot J_p(y)$ Since $J_p(x \cdot y), J_p(x), J_p(y)$ are all 1 or -1, equality holds over the integers as well □

Lemma 19. *Let $N = p \cdot q$ with p, q distinct primes and $y \in \mathbb{Z}_N \setminus \{0\}$ with $y \equiv y_p \mod p$ and $y \equiv y_q \mod q$. Then y is a quadratic residue modulo N if and only if y_p is a quadratic residue modulo p and y_q is a quadratic residue modulo q.*

Proof. If y is a quadratic residue modulo N then, by definition, $\exists x \in \mathbb{Z}_N$ such that $x^2 = y \mod N$. Let $x \equiv x_p \mod p$ and $x \equiv x_q \mod q$, $y \equiv y_p \mod p$ and $y \equiv y_q \mod q$. Then $x_p^2 \equiv y_p \mod p$ and $x_q^2 \equiv y_q \mod q$. We have thus shown that $y_p = x_p^2 \mod p$ and $y_q = x_q^2 \mod q$ and y_p, y_q are quadratic residues (with respect to the appropriate moduli). Conversely, if $y \equiv y_p \mod p$ and $y \equiv y_q \mod q$ with y_p, y_q be quadratic residues, then there exists $x_p \in \mathbb{Z}_p$ and $x_q \in \mathbb{Z}_q$ such that the previous equation holds. Let $x \in \mathbb{Z}_N$ be such that $x \equiv x_p \mod p$ and $x \equiv x_q \mod q$. Reversing the above steps shows that x is a square root of y modulo N. □

Now, we can extend the definition of the Jacobi symbol to the case of $N = p \cdot q$ a product of distinct, odd primes as follows.

Definition 72. *For any x relatively prime to $N = p{\cdot}q$, $J_N(x) = J_p(x){\cdot}J_q(x)$. We define J_N^{+1} as the set of elements in $\mathbb{Z}_N$ having Jacobi symbol $+1$, and J_N^{-1} analogously.*

Note now that $J_N(x) = +1$ can also occur when $J_p(x) = J_q(x) = -1$, that is, when x is not quadratic residue neither modulo p nor modulo q which implies that x is not a quadratic residue in $\mathbb{Z}_N$.

All the above are useful properties that used in Goldwasser-Micali cryptosystem which also takes advantage of the following property:

Lemma 20. *Let $N = p \cdot q$ be a product of distinct, odd primes, and $x, y \in \mathbb{Z}_N$. Then $J_N(x \cdot y) = J_N(x) \cdot J_N(y)$.*

Proof. Using the definition of Jacobi symbol and lemma 18, we have $J_N(x \cdot y) = J_p(x{\cdot}y){\cdot}J_q(x{\cdot}y) = J_p(x){\cdot}J_p(y){\cdot}J_q(x){\cdot}J_q(y) = J_p(x){\cdot}J_q(x){\cdot}J_p(y){\cdot}J_q(y)$ which implies that $J_N(x \cdot y) = J_N(x) \cdot J_N(y)$ as required. $\square$

Bibliography

[1] Craig Gentry (2009). Fully Hommomorphic Encryption Using Ideal Lattices. *STOC'09*, pp. 160–178.

[2] van Dijk, Marten and Gentry, Craig and Halevi, Shai and Vaikuntanathan, Vinod (2009). Fully Homomorphic Encryption over the Integers. *Cryptology ePrint Archive, Report 2009/616.*

[3] Craig Gentry (2010). Computing arbitrary functions of encrypted data. *Commun. ACM* pp. 97–105.

[4] Vadim Lyubashevsky, Daniele Micciancio (2005). Generalized Compact Knapsaks are Collision Resistant. *ICALP'06*, pp. 144–155.

[5] Daniele Micciancio, Oded Regev (2008). Lattice-Based Cryptography. *Post-Quantum Cryptography*, pp. 147–191.

[6] L. Babai (1986). On Lovász lattice reduction and the nearest lattice point problem Proc. *STACS '85*, pp. 13–20

[7] H.W. Lenstra, A.K. Lenstra, L. Lovász (1982). Factoring polynomials with rational coefficients. *Mathematische Annalen*, pp. 515–534

[8] Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman (1998). NTRU: A ring-based public key cryptosystem. *ANTS-III*, pp. 267–288

[9] Ronald L. Rivest, Adi Shamir, Leonard M. Adleman (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *CACM*, pp. 120–126

[10] Ronald Cramer, Victor Shoup (1998). A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. *CRYPTO '98*, pp. 1–3.

[11] Boneh, Dan and Goh, Eu-Jin and Nissim, Kobbi (2005). Evaluating 2-DNF Formulas on Ciphertexts. *Theory of Cryptography*, pp. 325–341.

[12] Elgamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE '85*, pp. 469–472.

[13] Okamoto, Tatsuaki, Uchiyama, Shigenori (1998). A new public-key cryptosystem as secure as factoring. *EUROCRYPT'98*, pp. 308–318.

[14] Whitfield Diffie, Martin E. Hellman (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, pp. 644–654.

[15] Richard M. Karp, Vijaya Ramachandran (1990). A Survey of Parallel Algorithms for Shared-Memory Machines. *Handbook of theoretical computer science (vol. A)*, pp. 869–941.

[16] J. Boyar, R. Peralta and D. Pochuev (1998). On The Multiplicative Complexity of Boolean Functions over the Basis $(\wedge, \oplus, 1)$. *University of Southern Denmark.*

[17] Daniele Micciancio, Bogdan Warinschi (2001). A Linear Space Algorithm for Computing the Hermite Normal Form. *Proceedings ISSAC 2001, Lecture Notes in Computer Sci., 2146.* pp. 126–145.

[18] P. Ribenboim (1991). The little book of big primes. *Springer-Verlag.*

[19] N. L. Carothers (1999). Real Analysis. *Cambridge University Press*

[20] Marlow Anderson and Todd Feil (2005). A First Course in Abstract Algebra: Rings, Groups and Fields, Second Edition. *Chapman & Hall/CRC.*

[21] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester (2005). A Modern Introduction to Probability and Statistics: Understanding Why and How (Springer Texts in Statistics) *Springer*

Index