Guro Rio Berge

Cryptographic Protocols from Lattice Assumptions

Bachelor's thesis in Mathematical Science
Supervisor: Kristian Gjøsteen
Co-supervisor: Tjerand Silde
June 2022

Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Guro Rio Berge

Cryptographic Protocols from Lattice Assumptions

Bachelor's thesis in Mathematical Science
Supervisor: Kristian Gjøsteen
Co-supervisor: Tjerand Silde
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Cryptographic Protocols from Lattice Assumptions

Guro Rio Berge

June 1, 2022

ABSTRACT

This thesis will introduce two of the lattice-problems, ***learning with errors*** (LWE) and ***shortest integer solution*** (SIS), which the security of lattice cryptography rely on. These problems were introduced by Oded Regev in 2005 [1] and Ajtai Miklos in 1996 [2], respectively. Further, the thesis will provide insight into how one can use them in encryption and commitment schemes, zero-knowledge protocols (ZKP), and digital signatures.

SAMMENDRAG

Bachelor oppgåva vil introdusera to gitter-problem, ***learning with errors*** (LWE) and ***shortest integer solution*** (SIS), som sikkerheten til gitter-kryptografi baserer seg på. LWE og SIS var introdusert av henholdsvis Oded Regev i 2005 [1] og Ajtai Miklos i 1996 [2]. Vidare vil bacheloren gi eit innblikk i korleis ein kan bruka gitter-problema i kryptering, forpliktelse-protokoll, kunnskapsløyse bevis, og digitale signaturer.

Acknowledgements

First, I want to give a big thanks to my supervisor Tjerand Silde. For our weekly meetings motivating me throughout my bachelor to do my best work and inspiring me by showing me all the cool sides of crypto. Secondly, I want to thank my fellow student Ole Martin Edstrøm for helping me when I was stuck on understanding difficult concepts and for always helping me with a smile.

I also want to thank all the students on Matteland who motivated me towards the end of my bachelor and beared through my complaining when I was tired. And lastly, I want to thank Karl Kristian Ladegård Lockert for reading and spell-checking my thesis.

I would not have been able to write this thesis without the help of all the people mentioned above, and I am grateful to them all.

Contents

List of Figures

Chapter 1

Introduction

Cryptography, or shortened crypto, can be seen as techniques to secure communication from malicious attackers. Today we focus the use of cryptography for communication over the internet, but crypto has in some form existed for many thousands of years. Before the age of computers, they used more basic codes and cryptic keys to hide secrets. An example is when they shifted the alphabet to encrypt messages. The term modern cryptography has emerged to differ between this earlier crypto and the crypto used for computers today.

In modern cryptography, we divide encryption into symmetric and asymmetric encryption. We use asymmetric crypto to exchange secret keys to the party communicating, among more, and symmetric crypto to encrypt the messages. In this thesis, we will focus on asymmetric crypto. Refer to [3] for more information about symmetric and asymmetric cryptography.

Asymmetric encryption is applied in many applications you most likely use daily. Examples are Facebook messenger and TLS (a cryptosystem that secures communication over a computer network such as email and Web browser). The security of asymmetric encryption today depends on the Diffie-Hellman assumption [4, section 1] and the RSA assumption [5, section 1], which rely on the hardness of the discrete logarithm and prime number factorization, respectively. This thesis will introduce two new mathematical assumptions asymmetric encryption can depend on.

1.1 Why Lattice Cryptography?

Since around 1980, scientists have worked on building a quantum computer. It is suggested this computer holds the potential to compute problems significantly faster than a classical computer can, and among these problems are discrete logarithms and factoring. Through Shor's algorithm, a quantum computer will be able to compute this problem in polynomial time [6]. Therefore we need new mathe-

matical problems whose computational complexity in a quantum computer is such that the asymmetric encryption is secure enough to rely on.

This is where lattice cryptography becomes quite helpful. The assumptions lattice cryptography is based on are not only hard for a classical computer but also for a quantum computer. For this reason, lattice cryptography has become a trendy and useful field studied today.

Chapter 2

Theory

We will assume that the reader has a basic knowledge of cryptography and ring theory.

2.1 Notation

The operations we will be performing through this paper will be in the ring $(\mathbb{Z}_q, +, \times)$, where q represent some positive integer. We let $\mathbb{U}$ be a set, and $\beta \in \mathbb{Z}$. Some useful notation to keep in mind:

- $[\beta]$: the set $\{-\beta, ..., -1, 0, 1, ..., \beta\}$.
- $\mathbf{t} \in \mathbb{U}^n$: $\mathbf{t}$ is a vector with n entries, and with element from $\mathbb{U}$ in each entry.
- $\mathbb{U}^{n \times m}$: an $n \times m$ matrix with element from $\mathbb{U}$ in each entry.
- $\mathbf{t} \xleftarrow{\$} \mathbb{U}$: $\mathbf{t}$ chosen at random from $\mathbb{U}$.
- $||\mathbf{s}||_\infty \leq \beta$: we have $s_i \leq \beta$ for $i = 1, 2, ...$

You will be introduced to the *commitment, challenge, and ciphertext space* in this thesis, and they will all be denoted $\mathcal{C}$, but from the context, it will be clear which space we are working with.

2.2 CPA Security

The security of the schemes will be CPA-secure (*semantically secure against chosen plaintext attack*). We will assume the reader has some knowledge of this and will only state the definitions. You can find a more detailed description of the definitions in [7, section 5.3]. Semantic security is defined to be the security against the adversary's advantage to differ between the encryption of two given messages. We will omit the formal definition of semantic encryption, but it can also be found in [7, section 5.2].

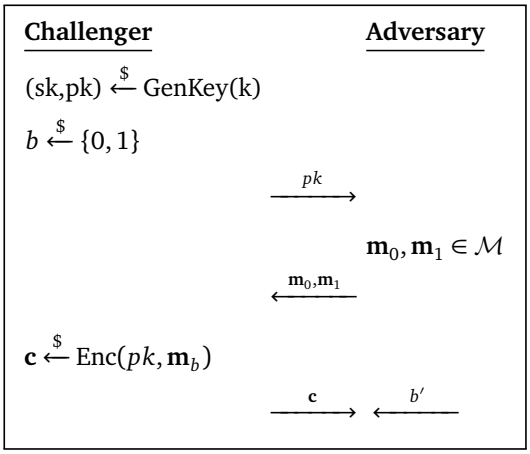

Figure 2.1: Visualisation of CPA-security

Attack game *(CPA-security)*

For a given scheme $\mathcal{E}$ = (E,D), defined over the key ($\mathcal{K}$), message ($\mathcal{M}$) and chipertext space ($\mathcal{C}$), and for a given adversary $\mathcal{A}$, we define an experiment

***Experiment*:**

- *The challenger selects* $(pk, sk) \xleftarrow{\$} \mathrm{GenKey}(k)$ *and* $b \xleftarrow{\$} \{0,1\}$.
- *The adversary submits two messages* $\mathbf{m}_0, \mathbf{m}_1 \in \mathcal{M}$ *of same length to the challenger.*
- *The challenger computes* $\mathbf{c} \xleftarrow{\$} Enc(pk, \mathbf{m}_b)$*, and sends* $\mathbf{c}$ *to the adversary.*
- *The adversary outputs a bit* $b' \in \{0,1\}$.

$\mathcal{A}$ wins if $b = b$'.

See figure 2.1 for a visualization of the game.

For $b \in \{0,1\}$, let W_1 be the event that $\mathcal{A}$ outputs 1 in the experiment, and W_0 when $\mathcal{A}$ outputs 0 in the experiment. We define $\mathcal{A}$'s advantage with respect to $\mathcal{E}$ as

$$CPAadv[\mathcal{A}, \mathcal{E}] := |Pr[W_0] - Pr[W_1]|.$$

Definition 1 *(CPA security)*

A scheme $\mathcal{E}$ is called ***semantically secure against chosen plaintext attack****, or simply* ***CPA secure****, if for all efficient adversaries $\mathcal{A}$, the value $CPAadv[\mathcal{A}, \mathcal{E}]$ is negligible.*

2.3 Learning With Errors

Now, the first mathematical problem which the lattice cryptography relies its security on, learning with errors (LWE). The definition below can be found in [8,

section 2.3]

Definition 2 *(Learning With Errors problem)*

For positive integers m, n, q og $\beta \ll q$, the $\mathsf{LWE}_{m,n,q,\beta}$ *problem asks to distinguish between the following two distributions:*

1. $(\mathbf{A}, \mathbf{As}+\mathbf{e})$*, where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{s} \xleftarrow{\$} [\beta]^{\mathbf{m}}, \mathbf{e} \xleftarrow{\$} [\beta]^{\mathbf{n}}$
2. $(\mathbf{A}, \mathbf{u})$*, where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{nxm}$ *and* $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}}$

i.e. given (A, t) *you are supposed to be able to distinguish* $\boldsymbol{t} = \boldsymbol{As} + \boldsymbol{e}$ *and* $\boldsymbol{u}$ *where* $\boldsymbol{u}$ *is picked at random from an uniform distribution over* $\mathbb{Z}_q^n$.

For the LWE problem to be hard, we have to use the LWE assumption. The assumption says that if we add a short vector to another vector, it will appear randomly picked from a uniform distribution.

What makes the problem hard is, therefore, the presence of $\mathbf{e}$. How hard the problem is relies on the parameters m, n, q, and β. How their values are determined in connection with each other. The more we increase m and β/q, the harder the problem will get. When increasing m the size of the sets $[\beta]^m$, $\mathbb{Z}_q^{n\times m}$ will increase. The bigger β/q are, the more values we can choose between for our parameters $\mathbf{s}$ and $\mathbf{e}$, and the harder they are to find for an adversary trying to determined if $\mathbf{t} = \mathbf{As} + \mathbf{e}$ or is picked at random. For a more detailed explanation of how to choose the parameters and how they are related, we refer to [8, section 2].

2.4 Shortest Integer Solution

The shortest integer solution, or SIS , problem focus on the hardness of finding a short vector satisfying a given equation.

Definition 3 *(Shortest Integer Solution Problem)*

For positive integers n, m, q and $\beta \ll q$*, and given* $(\mathbf{A}, \mathbf{t})$ *where*

$$\mathbf{A} \leftarrow \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}\times\mathbf{m}} \text{ and } \mathbf{t} = \mathbf{As},$$

The SIS *problem asks you to find a* $\mathbf{s}$ *such that* $\mathbf{t} = \mathbf{As}$ *and* $||\mathbf{s}||_\infty \leq \beta$.

The hardness of this problem rely on the size of β. The bigger β is, the more solutions there exist for the equation $\mathbf{t} = \mathbf{As}$. If $\beta > q/2$ the problem becomes trivial [8, section 3.2].

The lattice cryptography in focus in this thesis relies on both the SIS problem and the LWE problem. Therefore, when choosing values for the parameters, one has to look at them related to both the problems individually and together.

2.5 Lattices

We will briefly introduce how lattices are defined, and introduce the LWE and SIS problems in this notation. However, in this thesis, we will not be using this notation. This is because we will not have use for the formal definitions in our explanations. Therefore when working with these lattices, we will refer to $\mathbb{Z}_q$ over some vector space and consider matrices and vectors there.

Definition 4 *(Integer lattice)*

Let $(\mathbb{Z}^m, +)$ be a group, and let $\mathcal{B}$ be a basis for the vectorspace $\mathbb{Z}^{m\times m}$. Then we define the lattice Λ with respect to $\mathcal{B}$ to be a subgroup of $(\mathbb{Z}^m, +)$ such that

$$\Lambda = \mathcal{L}(\mathcal{B}) = \{\mathbf{v} \in \mathbb{Z}^m | \exists \mathbf{z} \in \mathbb{Z}^m; \mathcal{B}\mathbf{z} = \mathbf{v}\}$$

We will look at the set of lattices defined above, satisfying the definition below.

Definition 5 *(Q-ary integer lattice)*

Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, then the q-ary integer lattice Λ with respect to $\mathbf{A}$ is defined to be

$$\Lambda = \mathcal{L}_q^{\perp}(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}^m | \mathbf{A}\mathbf{v} \equiv 0(\mod q)\}$$

To connect lattice notation to the LWE and SIS problems, we have to introduce more theory. First, we say that two element $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}_q^n$ is in the same coset if and only if $\mathbf{A}\mathbf{s}_1 \equiv \mathbf{A}\mathbf{s}_2(\mod q)$. There are q^m such cosets since we work modulo q, and the elements are vectors of length m.

We also need some notation to measure the distance in the lattice. The definition can be found in [8, section 3.1.2].

For an m-dimensional lattice Λ and any vector $\boldsymbol{r} \in \mathbb{Z}_q^m$(not necessarily in Λ), the l_p-norm distance from $\boldsymbol{r}$ to the lattice is defined as

$$\Delta_p(\mathbf{r}, \Lambda) = \min_{\mathbf{v}\in\Lambda} \|\mathbf{v} - \mathbf{r}\|_p.$$

Let $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{Z}_q^m/\Lambda$, where $\mathbb{Z}_q^m/\Lambda$ is some coset, then we have $\Delta_p(\mathbf{r}_1, \Lambda) = \Delta_p(\mathbf{r}_2, \Lambda)$. Distance is therefore a well defined notation for coset.

If $\Lambda = \mathcal{L}_q^{\perp}(\mathbf{A})$ and $\mathbf{t} \equiv \mathbf{A}\mathbf{s}$ (mod q) defines a coset $\mathbf{s}+\Lambda$, we write

$$\Delta_p^C(\mathbf{t}, \Lambda) = \Delta_p(\mathbf{s}, \Lambda),$$

where Δ^C denote that $\mathbf{t}$ is the image of the coset under A, instead of some coset representative as $\mathbf{s}$.

We are now ready to introduce the LWE and SIS problems in lattice notation.

Definition 6 *(The* LWE *problem in lattice notation)*

For positive integers m, n, q og $\beta \ll q$*, and* $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$*, define* $\Lambda = \mathcal{L}_q^{\perp}(\boldsymbol{A} \mid \boldsymbol{I}_n)$*. The* $\mathsf{LWE}_{m,n,q,\beta}$ *problem asks to distinguish between the following two distributions:*

1. $\boldsymbol{t}$ *where* $\boldsymbol{t}$ *is some coset of* $\mathbb{Z}_q^m/\Lambda$*, and* $\Delta_\infty^C(\boldsymbol{t},\Lambda) \leq [\beta]$*.*
2. $\boldsymbol{u}$ *where* $\boldsymbol{u}$ *is some random coset of* $\mathbb{Z}_q^m/\Lambda$*.*

i.e. the problem becomes to distinguish between the two coset $\boldsymbol{t}$ *and* $\boldsymbol{u}$*.*

Definition 7 *(The* SIS *problem in Lattice notation)*

For positive integers n, m, q and $\beta \ll q$*, and given* $(\mathbf{A}, \mathbf{t})$ *where*

$$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}\times\mathbf{m}}\text{, and define } \mathcal{L}_q^{\perp}(\boldsymbol{A} \mid \boldsymbol{I}_n),$$

The SIS *problem asks you to find a* $\mathbf{s} \in \mathcal{L}_{\mathbf{q}}^{\perp}(\boldsymbol{A} \mid \boldsymbol{I}_n)$ *such that* $\|\mathbf{s}\|_\infty \leq \beta$*.*

More about this can be found in [8, section 3].

2.6 Polynomial Rings

As you might know, the polynomial ring holds a lot of structure. This can be advantageous in crypto schemes and protocols for optimization and making them more efficient. But the structure can also be used for attack, and therefore there must be taken precaution when choosing what ring to work with and parameters for the schemes and protocols.

In this section, you will get an introduction to polynomial rings and how to relate them to lattices. Read more about this in [8, section 4].

An Introduction to Polynomial Rings

One classic example of a ring is the integers $\mathbb{Z}$. And you might see the ring $M_2(\mathbb{F})$, the 2×2-matrix over $\mathbb{F}$, where $\mathbb{F}$ is a field. The polynomial ring is an extension of some ring $\mathcal{R}$, where one uses the elements of the ring as coefficients in polynomials. The polynomials will then be the element of the ring, and we denote the ring as $\mathcal{R}[\mathbb{X}]$. This ring fulfills the axioms of a ring structure and is left to be checked by the reader.

When formally defining a polynomial ring we write $(\mathcal{R}[\mathbb{X}], +, \times)$, where $\mathcal{R}$ is some ring, in our case $\mathbb{Z}$. The element of the ring $\mathcal{R}[\mathbb{X}]$ is then defined to be of the form $r(x) = \sum_{i=0}^{\infty} r_i x^i$ where $r_i \in \mathcal{R}$.

Addition will be performed the same way as in $\mathbb{R}[\mathbb{X}]$, polynomials over the real numbers, by adding the coefficient of the same degree term together. Multiplication will be the usual polynomial multiplication in $\mathbb{R}[\mathbb{X}]$ as well.

- $r(x)+a(x)=\sum_{i=0}^{d}(r_i+a_i)x^i$ where d = max{deg$(r(x),a(x))$}
- $r(x)\cdot a(x) = \left(\sum_{i=0}^{t} r_i x^i\right)\cdot\left(\sum_{i=0}^{s} a_i x^i\right) = \sum_{k=0}^{t+s}\sum_{l=0}^{k} r_l a_{k-l} x^k$, t and s equals deg$(r(x))$ and deg$(a(x))$ respectively.

In this thesis, we will focus on subrings of the polynomial ring $(\mathbb{Z}[\mathbb{X}],+,\times)$. The subring will be all polynomials of the form above, but only up to a given degree d. We note this subring $\mathbb{Z}[\mathbb{X}]/f(x)$, or simply $\mathbb{Z}_f[\mathbb{X}]$, where $f(x)$ is a monic irreducible polynomial of degree d. If you are familiar with some Galois theory, you might recognize this as the Galois group of the polynomial f.

The element of the ring will work modulo $f(x)$. Meaning, if a polynomial $g(x)$ has degree higher than $f(x)$, deg$(f(x))<$ deg$(g(x))$, there will exist another polynomial $r(x)\in\mathbb{Z}_f[\mathbb{X}]$ representing $g(x)$, where deg$(r(x))<$ deg$(f(x))$. We write $g(x)\equiv r(x) \mod (f(x))$. This is based on the euclidean algorithm, for every polynomial $g(x)$ and $f(x)$, deg$(f(x))<$ deg$(g(x))$ we can write $g(x)=f(x)h(x)+r(x)$, where deg$(r(x))<$ deg$(f(x))$, and $h(x)\in\mathbb{Z}[\mathbb{X}]$.

Some Linear Algebra

The question then becomes how to work with these polynomials in terms of matrices. This section is mainly based on [8, Section 4.1.1 and 4.2] with some more detailed calculations.

Polynomial Modulo Arithmetic

First a helpful trick to make note of is when multiplying two polynomials, $g(x)$ and $h(x)$, modulo some other polynomial $f(x)$. The multiplication can be written as the multiplication of a matrix $\mathbb{Z}^{d\times d}$ and a vector $\mathbb{Z}^{d}$, where d is the degree of the polynomial we are working module with. Observe:

$$g(x)\cdot h(x)=g(x)\cdot\left(\sum_{i=0}^{d-1}h_i x^i\right) \mod f(x)=\sum_{i=0}^{d-1}(g_i x^i \mod f(x))h_i \qquad (2.1)$$

To better understand this, we will go through an example part by part, but first, the example will be presented without any calculation.

Example, with out any calculation, of a reduction of the product of the two polynomials $g(x)=x^3-x^2-1$ and $h(x)=x^2-2$ modulo $f(x)=x^4+1$.

$$(x^3-x^2-1)(x^2-2) \mod (x^4+1) = -2x^3+x^2-x+3 \tag{2.2}$$

Detailed explanation:

$$(x^3-x^2-1)(x^2-2) = x^5-x^4-2x^3+x^2+2$$

Now we need to get rid of the 5^{th} and 4^{th} degree term. When working modulo a polynomial we can express terms with higher degree than $\deg(f(x))$ in lower degree terms by manipulating the polynomial $f(x)$.

$f(x) = x^4+1 = 0$ because we work modulo f(x).
$x^4 = -1$
$x^5 = -x$ By multiplying each side by x.

Replace the 5^{th} and $4th$ degree term with the new terms found above.

$$\begin{aligned}(x^3-x^2-1)(x^2-2) &= x^5-x^4-2x^3+x^2+2\\ &= -x-(-1)-2x^3+x^2+2\\ &= -2x^3+x^2-x+3\end{aligned}$$

Matrix Representation

Further, we will rewrite the polynomial in matrix representation. First, we will present the matrix representation. Then there will be a more detailed explanation of how to find the matrix representations.

I will represent $g(x)$ in the vector space of polynomials modulo $f(x)$. Hence the first term of equation 2.1, $\sum_{i=0}^{d-1}(g_i x^i \mod f(x))$. Then multiply it with there vector representation of $h(x)$.

$$\begin{pmatrix}-1 & -1 & 1 & 0\\ 0 & -1 & -1 & 1\\ -1 & 0 & -1 & -1\\ 1 & -1 & 0 & -1\end{pmatrix}\cdot\begin{pmatrix}-2\\ 0\\ 1\\ 0\end{pmatrix}=\begin{pmatrix}3\\ -1\\ 1\\ -2\end{pmatrix}$$

The vector representation of $h(x)$ is pretty straightforward, the first entry represent the lowest degree term, and for each entry the degree increase by 1. The matrix representation on the other hand is not that straightforward. Each of the column represent $g(x)$ time x^i for $i \in \{0,..,\deg(f(x))-1\}$. The i'th Column is the vector representation og $g(x)\cdot x^{i-1} \mod (f(x))$. We will only show the calculation for column 3.

$$\begin{aligned}&(x^3-x^2-1)\cdot x^2 \mod (x^4+1)\\ &= x^5-x^4-x^2 \mod (x^4+1)\\ &\equiv -x-(-1)-x^2\\ &= -x^2-x+1\end{aligned}$$

which has the vector representation $\begin{pmatrix}1 & -1 & -1 & 0\end{pmatrix}^T$

Some new notation will be necessary to talk about this theory.

For a polynomial $g(x)=\sum_{i=0}^{d-1} g_i x^i \in \mathbb{Z}_f[\mathbb{X}]$ we define:

- $\mathcal{V}_{g(x)} = \begin{pmatrix} g_0 \\ g_1 \\ \dots \\ g_{d-1}\end{pmatrix} \in \mathbb{Z}^d$.
- $\mathcal{M}_{g(x)} = \begin{pmatrix}\mathcal{V}_{g\cdot 1 \bmod f(x)} & \mathcal{V}_{g\cdot x \bmod f(x)} & \dots & \mathcal{V}_{g\cdot x^{d-1} \bmod f(x)}\end{pmatrix} \in \mathbb{Z}^{d\times d}$

Rewriting equation 2.2 in the notation above:

$$\mathcal{M}_{x^3-x^2-1}\cdot \mathcal{V}_{x^2-2} = \mathcal{V}_{-2x^3+x^2-x+3}$$

The polynomial $f(x)$ should be apparent from the context. So we can omit writing modulo $f(x)$ in the notation.

Extending the notation above, the matrix and vector can store even more data. This is done by letting each entry of an $n\times m$ matrix and vector of length n consist of an $\mathcal{M}_g$ for some $g(x)\in\mathbb{Z}_f[\mathbb{X}]$. We will use this when redefining an encryption scheme in section 3.2.

The definition is found in [8, section 4.1.1]:

Let $\mathbf{g} = \begin{pmatrix} g(x)_1 \\ g(x)_2 \\ \dots \\ g(x)_n\end{pmatrix}$ where $g(x)_i \in \mathbb{Z}_f[\mathbb{X}]$, and $G = \begin{pmatrix} g(x)_{1,1} & \dots & g(x)_{1,m} \\ \dots & \dots & \dots \\ g(x)_{m,1} & \dots & g(x)_{m,n}\end{pmatrix}$

where $g(x)_{i,j} \in \mathbb{Z}_f[\mathbb{X}]$

Further we define $\mathcal{V}_{\mathbf{g}}$ and $\mathcal{M}_G$:

$$\mathcal{V}_{\mathbf{g}} = \begin{pmatrix}\mathcal{V}_{g(x)_1} \\ \mathcal{V}_{g(x)_2} \\ \dots \\ \mathcal{V}_{g(x)_n}\end{pmatrix} \in \mathbb{Z}^{dn}, \text{ and } \mathcal{M}_G = \begin{pmatrix}\mathcal{M}_{g(x)_{1,1}} & \dots & \mathcal{M}_{g(x)_{1,m}} \\ \dots & \dots & \dots \\ \mathcal{M}_{g(x)_{m,1}} & \dots & \mathcal{M}_{g(x)_{m,n}}\end{pmatrix} \in \mathbb{Z}^{dn\times dm}.$$

This gives the equation

$$\mathcal{M}_G \cdot \mathcal{V}_{\mathbf{a}} = \mathcal{V}_{G\mathbf{a}} \in \mathbb{Z}^{dn} \text{ for } G \in \mathbb{Z}_f[\mathbb{X}]^{n\times m} \text{ and } \mathbf{a} \in \mathbb{Z}_f[\mathbb{X}]^m,$$

which can be checked holds using the earlier definitions.

The Generalized-LWE and -SIS Problem

We now have the tools to generalize the LWE and SIS problems from the integers $\mathbb{Z}$ to the polynomial ring $\mathbb{Z}_f[\mathbb{X}]$. The integer case can then be seen as a special case over $\mathbb{Z}_f[\mathbb{X}]$, namely when $\deg(f(x)) \leq 1$.

We will introduce some more notation before defining the generalized version of the LWE problem. $\mathbb{Z}_f[\mathbb{X}]$ has been used repeatedly throughout this section. But it is not to be confused with the subring $\mathbb{Z}_{q,f}[\mathbb{X}]$, which will appear in between from now. The difference is the space which the coefficients are picked from. In $\mathbb{Z}_{q,f}[\mathbb{X}]$, we have coefficients in $\mathbb{Z}_q$ instead of the entire $\mathbb{Z}$. In addition, throughout this section the notation $\mathbf{a} \xleftarrow{\$} [\beta]$ will mean the coefficients of all the polynomials in the vector $\mathbf{a}$ is picked uniformly from $[\beta]$. The following definition can be found in [8, section 4.2]

Definition 8 *(Learning With Errors Problem over $\mathbb{Z}_{q,f}[\mathbb{X}]$)*

For positive integers m, n, q and $\beta \ll q$, and ring $\mathbb{Z}_{q,f}[\mathbb{X}]$, the $\mathbb{Z}_{q,f}[\mathbb{X}] - \mathsf{LWE}_{n,m,\beta}$ problem asks to distinguish between the following two distributions:

1. $(\mathbf{A}, \mathbf{As} + \mathbf{e})$*, where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_{q,f}[\mathbb{X}]^{n\times m}, \mathbf{s} \xleftarrow{\$} [\beta]^m, \mathbf{e} \xleftarrow{\$} [\beta]^n$
2. $(\mathbf{A}, \mathbf{u})$*, where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_{q,f}[\mathbb{X}]^{n\times m}$ *and* $\mathbf{u} \xleftarrow{\$} [\beta]^n$

Definition 9 *(Shortest Integer Solution Problem over $\mathbb{Z}_{q,f}[\mathbb{X}]$)*

For positive integers n, m, q and $\beta \ll q$, and given $(\mathbf{A}, \mathbf{t})$ where

$$\mathbf{A} \leftarrow \mathbb{Z}_{q,f}[\mathbb{X}]^{\mathbf{n}\times\mathbf{m}} \text{ and } \mathbf{t} = \mathbf{As},$$

The SIS *problem asks you to find a* $\mathbf{s}$ *such that* $\mathbf{t} = \mathbf{As}$ *and* $||\mathbf{s}||_\infty \leq \beta$.

The hardness of the $\mathbb{Z}_{q,f}[\mathbb{X}]$-$\mathsf{LWE}_{n,m,\beta}$ and $\mathbb{Z}_{q,f}[\mathbb{X}]$-$\mathsf{SIS}_{n,m,\beta}$ problems can be reduced to the LWE defined in 2.3 and the SIS problem defined in 2.4.

Chapter 3

Encryption

The scheme presented in this thesis is the same as can be found in [8, section 2]. This section will start with an introduction to a scheme, the proof of its security, and then two ways to generalize it. First, by choosing the element used in the scheme from matrices instead of vectors, and then using polynomial rings instead of the integers $\mathbb{Z}$.

3.1 Encryption Scheme

3.1.1 Public-Key Encryption Schemes

When defining a public-key encryption scheme, we define three algorithms. The key-generation algorithm, which generates the public key (pk) and the secret key (sk), and the encryption and decryption algorithms which encrypt the message $\mathbf{m}$ and decrypt the ciphertext $\mathbf{c}$, respectively. We often shorten these terms to Gen, Enc, and Dec, respectively. Below is an overview of the inputs and outputs of the algorithms.

- $(pk, sk) \xleftarrow{\$} \text{GenKey}(k)$
- $\mathbf{c} \xleftarrow{\$} \text{Enc}(pk, \mathbf{m})$
- $\mathbf{m} \leftarrow \text{Dec}(sk, \mathbf{c})$

The Generator protocol takes as input the security parameter k, and as mentioned outputs a public key (pk) and secret key (sk) for encryption and decryption. The encryption scheme takes pk as input and uses it to encrypt the message $\mathbf{m}$. Then the decryption uses sk to decrypt $\mathbf{c}$ and finds the message $\mathbf{m}$. How the different protocols work on the inside varies from scheme to scheme. We will focus on a scheme presented in [8, section 2.3] represented in figure 3.1.

Gen(k)	**Enc**($\mathbf{m}, pk$)
$\mathbf{s}, \mathbf{e_1} \xleftarrow{\$} [\beta]^m$	$\mathbf{r}, \mathbf{e_2} \xleftarrow{\$} [\beta]^m$
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{mxm}$	$e_3 \xleftarrow{\$} [\beta]$
$\mathbf{t} := \mathbf{As} + \mathbf{e_1}$	$\mathbf{u^T} := \mathbf{r^T A} + \mathbf{e_2^T}$
$sk := \mathbf{s}, pk := (\mathbf{A}, \mathbf{t})$	$v := \mathbf{r^T t} + e_3 + \lceil q/2 \rfloor \mathbf{m}$
return (sk, pk)	$\mathbf{c} = (\mathbf{u^T}, v)$
Dec($sk, \mathbf{c}$)	return $\mathbf{c}$
$\mathbf{m} = v - \mathbf{u^T s}$	
return $\mathbf{m}$	

Figure 3.1: A CPA secure encryption scheme $\mathcal{E}$ based on the LWE -problem.

3.1.2 An LWE-Based Encryption Scheme

The security of the scheme $\mathcal{E}$ in figure 3.1 and the later generalized version of the scheme is based on the LWE problem in section 2.3. The scheme in figure 3.1 will be the encryption scheme we will be following and building upon in this thesis. You will find a more detailed description of the scheme in [8, section 2.3.1].

Note, in the scheme in figure 3.1 to differ between the message $\mathbf{m}$, and the dimension of the vectors and matrices m, we write the message $\mathbf{m}$ in bold even though it is not a vector.

For a game-based proof of the CPA security of the scheme, look at figure 3.2. If you're not familiar with game-based proof, one can find another explanation in [8, section 2.3.1]. From the proof in figure 3.2 we have

$$\text{CPAadv}[\mathbf{A}, \mathcal{E}] \leq \text{LWEadv}[\mathbf{R}, \mathcal{E}]$$

Thus if we can break the $\mathcal{E}$ scheme with probability α, we can break LWE with the same probability.

To see the correctness of the scheme we will write out the decryption,

$$\begin{aligned}\mathbf{v} - \mathbf{u^T s} =& \mathbf{r^T}(\mathbf{As} + \mathbf{e_1}) + e_3 + \lceil 2/q \rfloor \mathbf{m} - (\mathbf{r^T A} + \mathbf{e_2^T})\mathbf{s} \\ =& \mathbf{r^T e_1} + e_3 + \lceil 2/q \rfloor \mathbf{m} - \mathbf{e_2^T s}.\end{aligned}$$

This is not the same $\mathbf{m}$ decrypted, the decryption is left with multiple error terms and the message is multiplied with $\lceil 2/q \rfloor$. But looking closely at the decryption, you will realise it is pretty close. If you remember, the error terms are all bounded by some β. We rewrite the error term, $\mathbf{r^T e_1} + e_3 - \mathbf{e_2^T s} = e$. Now by looking at the upper bound of each of the terms in the error we get $\mathbf{r^T e_1} \leq m\beta^2$, $\mathbf{e_2^T s} \leq m\beta^2$ and $e_3 \leq \beta$. This gives us $e \leq 2m\beta^2 + \beta$. Then, by depending the parameters on the

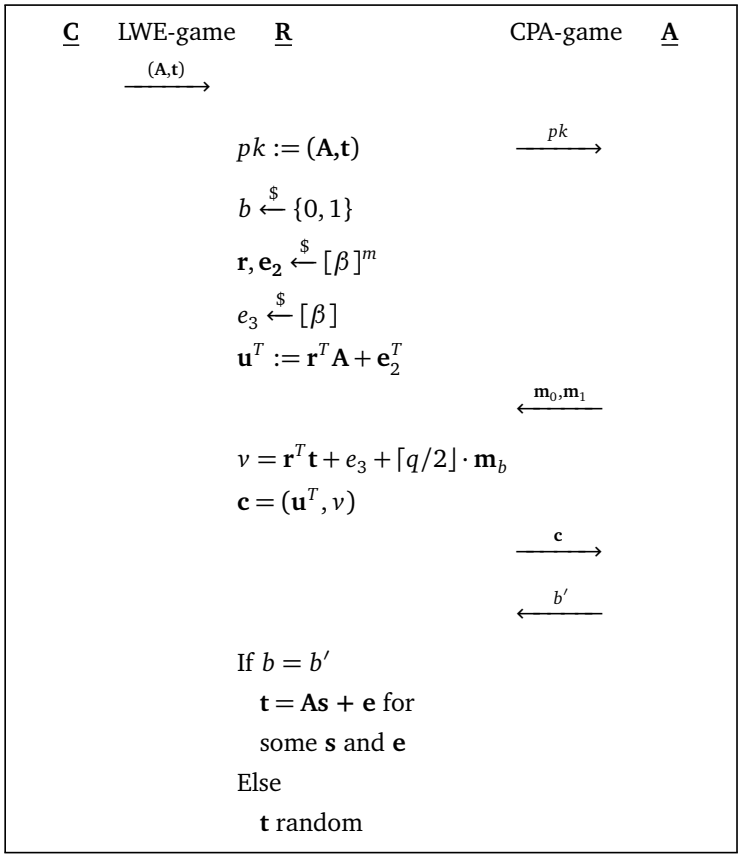

Figure 3.2: A CPA security proof of the encryption scheme in figure 3.1

Gen(k)	**Enc**(sk,pk)
$\mathbf{S}, \mathbf{E}_1 \xleftarrow{\$} [\beta]^{m\times l}$	$\mathbf{R}, \mathbf{E}_2 \xleftarrow{\$} [\beta]^{k\times m}$
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times m}$	$\mathbf{E}_3 \xleftarrow{\$} [\beta]^{k\times l}$
$\mathbf{T} := \mathbf{AS} + \mathbf{E}_1$	$\mathbf{U} := \mathbf{RA} + \mathbf{E}_2$
$sk := \mathbf{S}, pk := (\mathbf{A}, \mathbf{T})$	$\mathbf{V} := \mathbf{RT} + \mathbf{E}_3 + \lceil q/2 \rfloor \mathbf{M}$
return (sk, pk)	$\mathbf{C} = (\mathbf{U},\mathbf{V})$
Dec(sk, c)	return $\mathbf{C}$
$\mathbf{M} = \mathbf{V} - \mathbf{US}$	
return $\mathbf{M}$	

Figure 3.3: A generalised version of the LWE -scheme in figure 3.1

inequality $2m\beta^2 + \beta < \frac{q}{4}$, we can find the value of $\mathbf{m}$ by examining the value of the decryption. If the decryption has value closer to $q/2$ than 0, we will conclude with $\mathbf{m} = 1$. On the other hand, if the decryption has value closer to 0, we will conclude $\mathbf{m} = 0$.

So far, the only downside of the scheme seems to be the restriction of the length of the message. Using the scheme in figure 3.1 we will only be able to decrypt one bit at a time. Therefore we have constructed a more generalized scheme, figure 3.3, where we trade a longer public key for a longer message.

3.1.3 A Generalised Version of the Scheme

The same security argument from the scheme in figure 3.1 applies in the generalised scheme. Decomposing the parameter $\mathbf{T}$ from the second part of the public key, we see $\mathbf{T} = \mathbf{AS} + \mathbf{E}_1 = (\mathbf{As}_1 + \mathbf{e}_1, ..., \mathbf{As}_l + \mathbf{e}_l)$. Thus we can look at the public key as $(\mathbf{A}, \mathbf{T}) = (\mathbf{A}, \mathbf{As}_1 + \mathbf{e}_1, ..., \mathbf{As}_l + \mathbf{e}_l) = (\mathbf{A}, \mathbf{As}_1 + \mathbf{e}_1), ..., (\mathbf{A}, \mathbf{As}_l + \mathbf{e}_l)$. Which is secure by the game based security proof in figure 3.2. But there will be some loss of the security from the expansion [8, section 2.4].

When Looking at the $(i, j)^{th}$ coefficient of $\mathbf{V} - \mathbf{US} = \mathbf{RE}_1 + \mathbf{E}_3 + \frac{q}{2}\mathbf{M} - \mathbf{E}_2\mathbf{S}$ we see $\mathbf{r}^T\mathbf{e}_1 + e_3 + \frac{q}{2}\mathbf{m} - \mathbf{e}_2^T\mathbf{s}$ which should look recognizable. It is the same decryption we got when decryption the scheme earlier. The same arguments for correctness follows.

3.1.4 The Difference of the two Schemes

Here are a couple of things to make a note of. First, the message size has increased from 1 bit to $k \times l$ bits. On the other hand the public key size has increased from $|\rho| + m\log(q)$ to $|\rho| + lm\log(q)$ bits, and the ciphertext has increased from $(m+1)\log(q)$ to $k(m+l)\log(q)$ bits. By varying the parameters k and l, we can

minimize the combined size of the public key and the ciphertext. But it has increased by a significant amount.

On the upside, we can now encrypt bigger messages, as was our goal. But there is still potential for improvement. You can read more about some of the ways of optimizing the scheme in [8, section 2.5]. Another way is by instead of using the integers $\mathbb{Z}$ for building the matrices and vectors. One can use the polynomial ring $\mathbb{Z}_f[\mathbb{X}]$, which we will discuss in the next section.

3.2 Using the Structure of Polynomial Rings

In section 2.6, you were introduced to the polynomial ring, more specifically, its structure and how to convert it to linear algebra. In this section, we will use this to generalize the scheme from figure 3.1. The only difference is a switch from computing over $\mathbb{Z}$ to $\mathbb{Z}_{q,f}[\mathbb{X}]$. One of the advantages of the new scheme will be the ability to encrypt a message of d bits instead of only one, as in the scheme in figure 3.1. There will also be some advantages we can use from the structure of the polynomial ring for efficiency, but the structure can be used for attacks as well. To read more about this, see in [8, section 4.4].

The new scheme can be found in figure 3.4. Remember that the notation $\mathbf{a} \xleftarrow{\$} [\beta]$ when considering polynomial rings will mean that the coefficients of all the polynomials in $\mathbf{a}$ is picked uniformly from $[\beta]$.

The correctness of the scheme in figure 3.4 can be computed in a similar manner as the scheme in figure 3.1. The error term

$$\mathcal{M}_{\mathbf{r}^T}\mathcal{V}_{\mathbf{e}_1} + \mathcal{V}_{e_3} - \mathcal{M}_{\mathbf{e}_2^T}\mathcal{V}_{\mathbf{s}}$$

can be bounded as before by $q/4$ by making β and m depend on an equation derived from the upper bound on the error term.

The security will as well follow from the security proof in figure 3.2. Read more about the security argument in [8, section 4.3.2].

GenKey(k)	**Enc**(sk, pk)
$\mathbf{s}, \mathbf{e}_1 \xleftarrow{\$} [\beta]^m$	$\mathbf{r}, \mathbf{e}_2 \xleftarrow{\$} [\beta]^m$
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_{q,f}[\mathbb{X}]^{m \times m}$	$e_3 \xleftarrow{\$} [\beta]$
$\mathbf{t} := \mathbf{A}\mathbf{s} + \mathbf{e}_1$	$\mathbf{u} := \mathbf{r}^T\mathbf{A} + \mathbf{e}_2^T$
$sk := \mathbf{s}, pk := (\mathbf{A}, \mathbf{t})$	$v := \mathbf{r}^T\mathbf{t} + e_3 + \lceil q/2 \rfloor \mathbf{m}$
return (sk, pk)	$\mathbf{c} = (\mathbf{u}, v)$
Dec$(sk, \mathbf{c})$	return $\mathbf{c}$
$\mathbf{m} = v - \mathbf{u}\mathbf{s}$	
return $\mathbf{m}$	

Figure 3.4: A CPA secure encryption scheme based on the $\mathbb{Z}_{q,f}[\mathbb{X}]$ -LWE problem, generalized from the scheme in figure 3.1

Chapter 4

Commitments and Zero-Knowledge Proofs

Commitment schemes and zero-knowledge proofs (ZKP) are central in modern cryptography. Commitment is used for committing to a value without sharing it. Thus you can not change the value after committing to it without the other party knowing. It is used in numerous protocols, for instance, in coin flipping and zero-knowledge.

Zero-knowledge also has many applications. A few examples are electronic voting, authentication, and blockchains. Zero-knowledge is used to prove knowledge about a statement without sharing any more information than the public part of the statement. Let's look at the zero-knowledge part of a digital election. You want to vote for the party you're supporting, but one has to prove to the voting system that one's vote is valid. At the same time, you don't want to share what the vote said.

This section will start with a more formal definition of commitment and an example of a commitment scheme. Then we will continue with a more formal definition of zero-knowledge, and end with a ZKP -protocol using the previous commitment scheme.

4.1 Commitment

A commitment scheme consists of three algorithms the public-key generator, *KeyGen*, which takes the security parameter k as input and outputs a public commitment key, PP, the algorithm *Commit* which commits a given message using the public commitment key, and the algorithm *Open* which verifies or reject the commitment. The public-key generator also outputs a definition of the message space, probability space, and the commitment space.

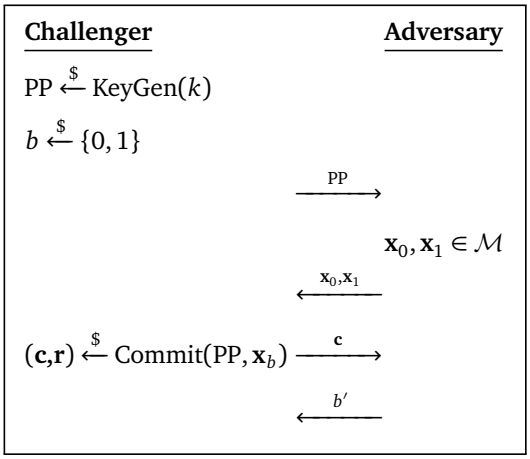

Figure 4.1: A visualisation of the hiding property.

- $\mathsf{PP} \xleftarrow{\$} \text{KeyGen}(k)$, KeyGen also output a definition of the message space $\mathcal{M}$, randomness space $\mathcal{R}$, and commitment space $\mathcal{C}$.
- $(\mathbf{c}, \mathbf{r}) \xleftarrow{\$} \text{Commit}(\mathsf{PP}, \mathbf{x})$, where $\mathbf{x} \in \mathcal{M}$, and $\mathbf{c} \in \mathcal{C}$ and $\mathbf{r} \in \mathcal{R}$.
- $b \leftarrow \text{Open}(\mathsf{PP}, \mathbf{x}, \mathbf{c}, \mathbf{r})$, where $b \in \{0,1\}$.

KeyGen and Commit are PPT (*Probabilistic polynomial time*) algorithms, while the last one, Open, is a deterministic algorithm. The algorithm Open outputs 0 or 1 dependent on if it verifies or rejects the commitment to $\mathbf{x}$.

For the scheme to be a commitment scheme, it has to fulfill two properties called hiding and binding.

Definition 10 ***(Hiding)***

The hiding property is dependent on the scheme's ability to hide the message. For two messages $\mathbf{x}_0$ and $\mathbf{x}_1$ the adversary $\mathcal{A}$ should not be able to distinguish between the commitments for the messages. For a visualization, see figure 4.1.

We differ between the two terms computational and statistical hiding. The scheme is computational hiding if $\mathcal{A}$ is restricted to polynomial-time algorithms and is statistical hiding if we allow $\mathcal{A}$ to be any all-powerful algorithm.

Definition 11 ***(Binding)***

The scheme is said to be binding if the probability for an adversary $\mathcal{A}$ finding two openings for the same commitment is less than ϵ.

$$Pr\Big[\mathcal{A}(\mathsf{PP}) = (\mathbf{x}, \mathbf{x}', \mathbf{r}, \mathbf{r}', \mathbf{c})\ s.t.\ \mathbf{x} \neq \mathbf{x}' \wedge Open(\mathsf{PP}, \mathbf{x}, \mathbf{c}, \mathbf{r}) = Open(\mathsf{PP}, \mathbf{x}', \mathbf{c}, \mathbf{r}') = 1 | \mathcal{PP} \xleftarrow{\$} KeyGen(k)\Big] < \epsilon$$

KeyGen(k)

$\mathbf{A_1} = [\mathbf{I_n}\ \mathbf{A_1'}] \xleftarrow{\$} \mathbb{Z}_{q,f}[\mathbb{X}]^{n\times(k-n)}$

$\mathbf{A_2} = [\mathbf{0}^{l\times n}\ \mathbf{I_l}\ \mathbf{A_2'}] \xleftarrow{\$} \mathbb{Z}_{q,f}[\mathbb{X}]^{l\times(k-n-l)}$

Return $\mathbf{A_1}, \mathbf{A_2}$

Com(x,$\mathbf{A_1}, \mathbf{A_2}$)

$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_{\beta,f}[\mathbb{X}]^k, f = 1$

$\mathbf{c} = [\mathbf{c}_1\ \mathbf{c}_2]^T = [\mathbf{A}_1\ \mathbf{A}_2]^T \cdot \mathbf{r} + [\mathbf{0}^n\ \mathbf{x}]^T$

Return $\mathbf{c}, \mathbf{r}, f$

Open(c, r, x,$f, \mathbf{A_1}, \mathbf{A_2}$)

Check:

$f \cdot [\mathbf{c}_1\ \mathbf{c}_2]^T = [\mathbf{A_1}\ \mathbf{A_2}]^T \cdot \mathbf{r} + f \cdot [\mathbf{0}^n\ \mathbf{x}]^T$

and

$\|r_i\|_2 \leq 4\sigma\sqrt{N}\ \forall\ i$ and $f \in \mathcal{C}'$

return 0 or 1

Figure 4.2: A Lattice based Commitment Scheme

We also differ between computational and statistical binding. Similar to above, if the adversary $\mathcal{A}$ is restricted to polynomial-time algorithms, we say it is computational, and if $\mathcal{A}$ is allowed to be an all-powerful algorithm, it is statistical like above.

4.1.1 Example Commitment Scheme

In this section, there will be introduced a commitment scheme, figure 4.2, depending on the security of the LWE and SIS problems. Further, we will use the scheme to define a ZKP -protocol, figure 4.4. For a more detailed overview of the schemes, you can look in [9, Section 4.1]. A heads-up about the notation used in the thesis referred to, they rely the security on the DKS^{∞} *(Decisional Knapsack problem in l_∞ norm)* and SKS^2 *(Search Knapsack problem in l_2 norm)* problem. These problems can be rewritten to the LWE and SIS problems, respectively, and will be used instead in the security argument.

The hiding and binding properties of the scheme in figure 4.2 rely on the LWE and SIS problems, respectively. Now there will be stated a couple of lemmas from [9, section 4] which argue for hiding and binding of the scheme.

Lemma 1 *For any $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_f[\mathbb{X}]^l$, if there exists an algorithm $\mathcal{A}$ that has advantage ϵ in breaking the hiding property of the commitment scheme, figure 4.2, then there exists another algorithm $\mathcal{A}'$ that runs in the same time and has advantage ϵ in solving the $\mathsf{LWE}_{n+l,k,\beta}$ problem.*

Proof can be found in [9, section 4.2].

Lemma 2 *If there is an algorithm $\mathcal{A}$ that can break the binding of the commitment scheme, figure 4.2, with probability ϵ, then there is an algorithm $\mathcal{A}'$ who can solve the $\mathsf{SIS}_{n,k,16\sigma\sqrt{\kappa N}}$ problem with advantage ϵ.*

Proof can be found in [9, section 4.2].

The binding and hiding property follows from lemma 1 and 2.

By varying the parameters, one can make the scheme either statistical hiding or statistical binding, but not both simultaneously. The parameters can also be set such that the scheme is hiding and binding but then computational for both. Read more about the different instantiations here [9, section 4.3], and how to set the parameters in [9, section 3.1].

4.2 Zero-Knowledge

For a protocol to be Zero-Knowledge (ZK) it has to satisfy three properties completeness, soundness, and honest-verifier zero-knowledge. We will introduce these properties below. The ZK protocols are divided into interactive and non-interactive protocols. In this thesis, we will look at an interactive one. But it can be transformed into a non-interactive one by the Fiat-Shamir transform [10].

The protocol consists of a prover ($\mathcal{P}$) and a verifier ($\mathcal{V}$), and the prover will try to prove his knowledge to the verifier but at the same time don't reveal any more information than necessary. The ZK protocol in figure 4.3 can be read in more detail in [11, Section 2].

Both $\mathcal{P}$ and $\mathcal{V}$ will input x, also called the statement. The statement consists of one or more public parameters used in the ZK -protocol. $\mathcal{P}$ will have another input as well, the witness w unknown to $\mathcal{V}$. The witness is used to prove the knowledge of $\mathcal{P}$ to $\mathcal{V}$.

For the protocol to be secure, it has, as mentioned, to fulfill three properties.

Definition 12 *Completeness*

If $\mathcal{P}$ and $\mathcal{V}$ on input (x,w) and x, respectively, follow the protocol honestly. Then $\mathcal{V}$ will always accept except with a negligible probability.

Definition 13 *(Soundness)*

Soundness looks at the protocol's protection from a dishonest prover. A dishonest prover is a $\mathcal{P}'$ who doesn't know the witness w but can convince $\mathcal{V}$ with more than probability $\mathcal{E}$ it knows w.

Another property that implies soundness is special soundness.

If $\mathcal{P}$ sends $\mathbf{a}$, $\mathcal{V}$ answer $\mathbf{e}$ and $\mathbf{e}'$ where $\mathbf{e} \neq \mathbf{e}'$ and $\mathcal{P}$ is able to compute $\mathbf{z}$ and $\mathbf{z}'$, $\mathbf{z} \neq \mathbf{z}'$, which both is accepted by $\mathcal{V}$. Then $\mathcal{V}$ can efficiently compute the secret $\mathbf{w}$.

Definition 14 *Honest-Verifier* ZK

The Honest-verifier ZK *property gives the protocol the security against $\mathcal{V}$ learning*

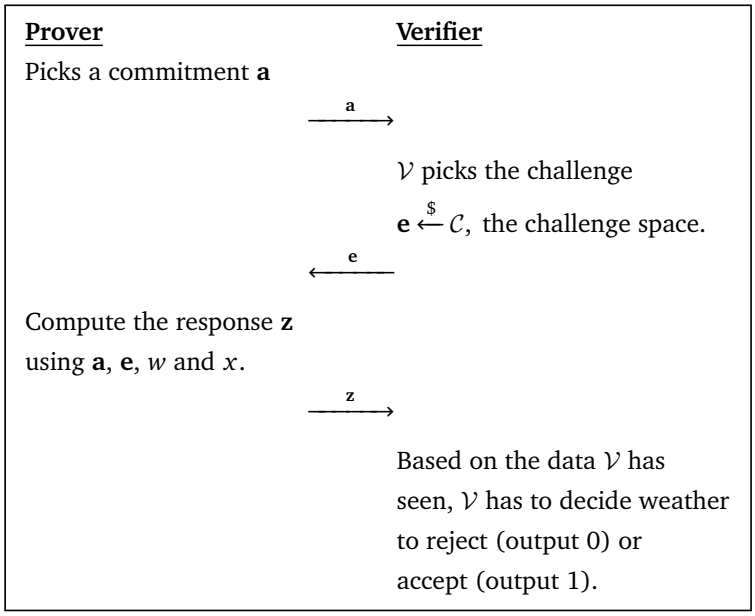

Figure 4.3: An general description of a interactive Zero-Knowledge proof

anything about the secret **w** *only* $\mathcal{P}$ *knows. Given the following property to the protocol, it should satisfy honest-verifier* ZK.

There exist a polynomial time algorithm able to compute a accepted conversation $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ *on input x, which will have the same probability distribution as an actually conversation between* $\mathcal{P}$ *and* $\mathcal{V}$.

4.2.1 Example Zero-Knowledge Protocol

This protocol will use the commitment scheme introduced in the previous section. The protocol is described in figure 4.4.

Below you will find an argument for completeness, soundness, and Honest-verifier Zero-Knowledge for figure 4.4. The argument is found in [9, section 4.4].

Completeness:

For completeness there are two properties that has to be satisfying. for all i $\|z_i\|_2 \leq 2 \cdot \sigma\sqrt{N}$ and $\mathbf{A}_1 \cdot \mathbf{z} = \mathbf{t} + \mathbf{d} \cdot \mathbf{c}_1$. We only have to worry about the first one, since by definitions of the parameters the second one holds. From in [9, section 2.3, Remark 1] the first property holds except with negligible probability.

Special Soundness:

As mentioned, if the protocol fulfills special soundness, it fulfills soundness.

Pick two different challenges $\mathbf{d}$ and $\mathbf{d}'$. Define $\mathbf{f} = (\mathbf{d} - \mathbf{d}') \in \mathcal{C}'$ and $\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 \\ ... \\ \mathbf{r}_k \end{bmatrix} = \mathbf{z} - \mathbf{z}'$ such that $\mathbf{A}_1 \cdot \mathbf{r} = \mathbf{f} \cdot \mathbf{c}_1$. The message $\mathbf{x}$ contained in $\mathbf{c}$ is then defined to be $\mathbf{x} = \mathbf{c}_2 - \mathbf{f}^{-1} \cdot \mathbf{A}_2 \cdot \mathbf{r}$. We then have $\|r_i\|_2 \leq \|z_i\|_2 + \|z_i\|_2 \leq 4\sigma\sqrt{N}$ and $\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \cdot \mathbf{r} + \mathbf{f} \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{x} \end{bmatrix} = \mathbf{f} \cdot \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}$, the opening ($\mathbf{x}$, $\mathbf{r}$,$\mathbf{f}$) is valid.

Honest-Verifier Zero-Knowledge:

To simulate a conversation, one start by picking a $\mathbf{z} \xleftarrow{\$} \mathcal{N}_\sigma^k$, a vector of length k where the element of the entries are picked from a normal distribution centred around zero. Then one compute $\mathbf{t} = \mathbf{A}_1\mathbf{z} - \mathbf{dc}_1$ using the already decided $\mathbf{z}$. This conversation will be statistical indistinguishable from the real non-aborting transcript by [9, section 2.3, lemma 2].

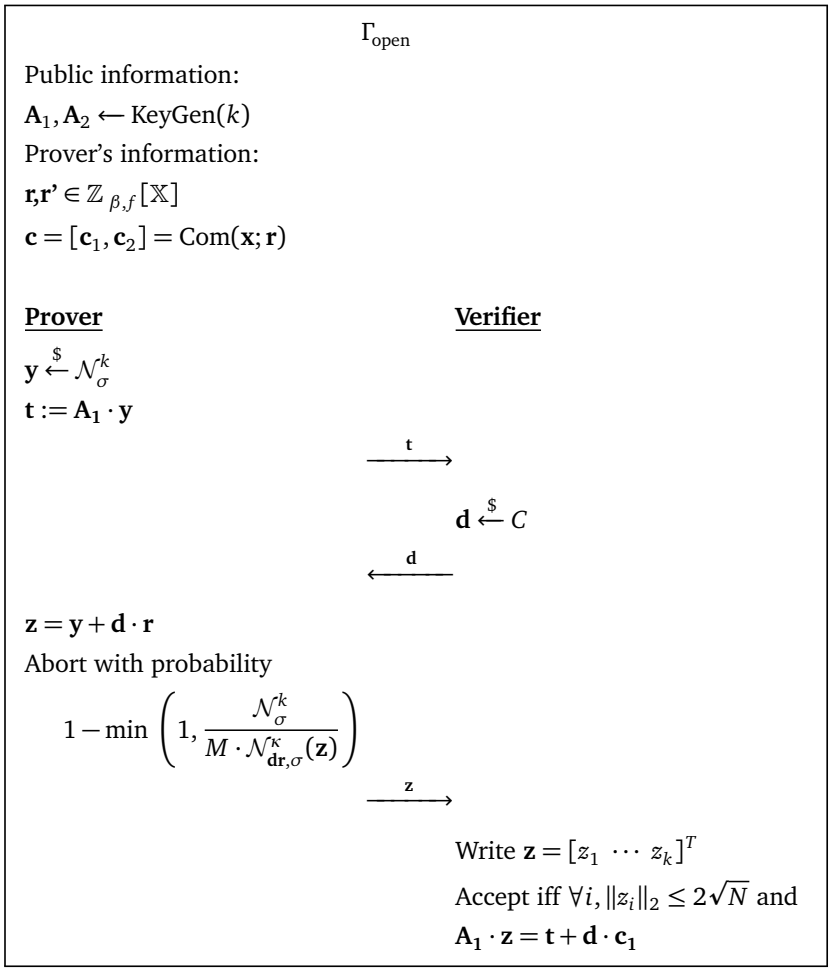

Figure 4.4: ZK Protocol using commitment from figure 4.2

Chapter 5

Digital Signature

Digital signatures are schemes used for verifying the integrity and authenticity of messages and other received data. One example of use is to verify the sender when sending a message.

In this section, we will first present an interactive ZK -protocol found in [8, section 5.2], which can be used to build a digital signature scheme using the Fiat-Shamir transform [10] making the ZK -protocol non-interactive.

5.1 Digital Signature from the Σ-Protocol

From the public information given in the Σ-protocol, figure 5.1, (**A**,**t**), one can see that the prover's information is kept secret by using the hardness of the $\mathbb{Z}_{q,f}[\mathbb{X}]$ -LWE and $\mathbb{Z}_{q,f}[\mathbb{X}]$ -SIS problem. The scheme relies on the conversation and the hardness of the same problems.

The Σ-protocol also allow to rather be checking for some $\mathbf{s'}_1 \in [\beta']^m$, $\mathbf{s'}_2 \in [\beta']^n$ where $\beta' > \beta$, instead of checking for $[\beta]$. The reason for this is that it is significant less efficient proving knowledge for small $\mathbf{s} \in [\beta]$ than for a relaxed versions $\mathbf{s'} \in [\beta']$. Therefor one end up proving knowledge about $(\mathbf{A}, \mathbf{t} = \mathbf{As_1} + \mathbf{s_2})$ by actively proving knowledge about the relaxed solution $\mathbf{As'}_1 + \mathbf{s'}_2 = \mathbf{t'}$ where $\mathbf{t'} = \mathbf{e't}$. A more detailed explanation of why we can do this can be read in [8, section 5.1]

The size of the coefficients of $\mathbf{s'_1}, \mathbf{s'_2}$ and $\mathbf{e'}$ will be determined by the challenge space. One can read more about this in [8, section 5.1.1].

The Σ-protocol is visualised in figure 5.1. The protocol consists of the prover generating to masking variables $\mathbf{y_1}$ and $\mathbf{y_2}$ used to mask the commitment ω. Then the verifier responds with a challenge e from the challenge space $\mathcal{C}$. The last part consists of the prover adding the masking variable to the challenge multiplied by the witness $\mathbf{s_1}$ and $\mathbf{s_2}$.

Completeness:

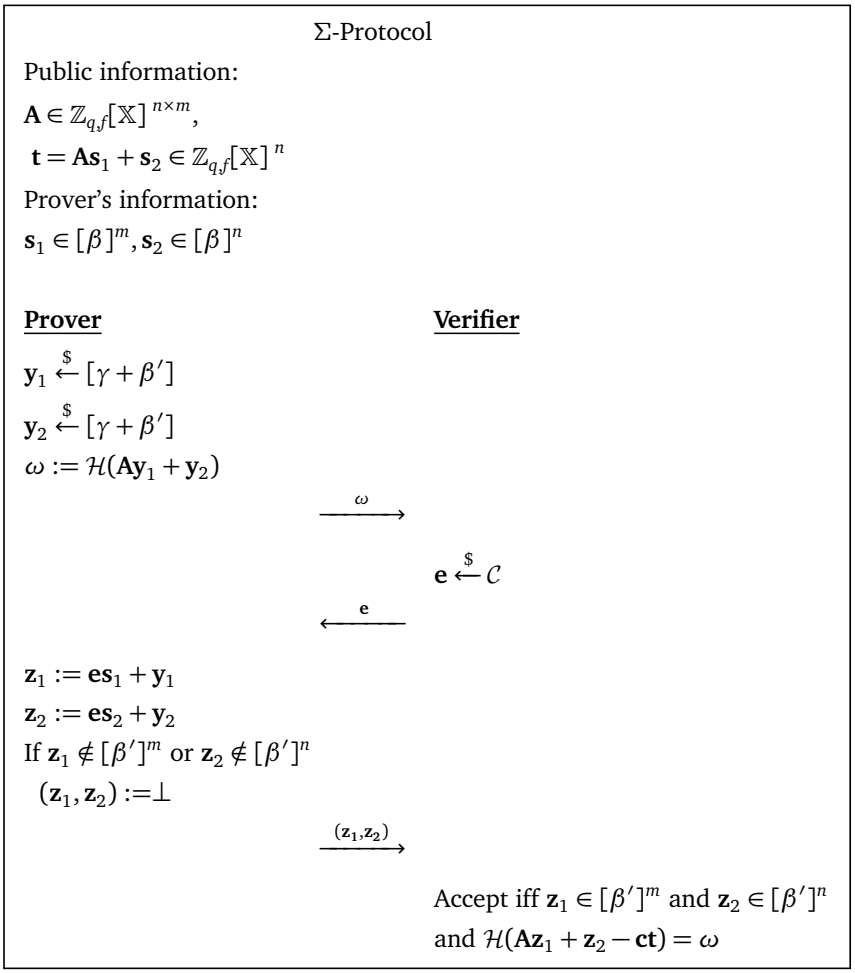

Figure 5.1: ZK -Protocol used for building a digital signature.

The protocol won't have perfect completeness, and will therefore have a rejection sampling step were checking the size of $\mathbf{z}_1$ and $\mathbf{z}_2$. But when $(\mathbf{z_1}, \mathbf{z_2} \neq \perp)$, then the first property is fulfilled. For the second property one have to check if $\mathbf{Ay}_1 + \mathbf{y}_2 = \mathbf{Az}_1 + \mathbf{z}_2 - \mathbf{ct}$, which written out are the same.

Soundness:

We use an extractor to obtain two transcripts (ω, $\mathbf{e}$, $\mathbf{z}_1$ $\mathbf{z}_2$) and (ω, $\mathbf{e}'$, $\mathbf{z}'_1$, $\mathbf{z}'_2$) where both pairs ($\mathbf{z}_1$,$\mathbf{z}_2$) and ($\mathbf{z}'_1$, $\mathbf{z}'_2$) is accepted by the verifier. Then, if there are no collisions, we have $\mathbf{Az_1} + \mathbf{z_2} - \mathbf{et} = \mathbf{Az'_1} + \mathbf{z'_2} - \mathbf{e't}$ which then gives $\mathbf{A}(\mathbf{z_1} - \mathbf{z'_1})$ $+(\mathbf{z_2} - \mathbf{z'_2}) - (\mathbf{e} - \mathbf{e}')\mathbf{t}$. The last solution will give the exact statement to the relaxed solution.

Honest Verifier Zero-Knowledge:

Recall that HVZK means you should be able to compute a conversation (ω, $\mathbf{e}$, $(\mathbf{z}_1, \mathbf{z}_2)$) with the same distribution as a real conversation in the protocol without knowing $\mathbf{s}_1$ and $\mathbf{s}_2$. This can be shown and is in detailed explained in [8, section 5.2.1].

As mentioned, when using the Fiat-Shamir tranform [10], one can make the Σ-protocol non-interactive. The protocol can also be optimized. More about this can be read in [8, section 5].

Bibliography

[1] O. Regev, *On lattices, learning with errors, random linear codes, and cryptography*, https://dl.acm.org/doi/pdf/10.1145/1060590.1060603, 2005.

[2] A. Miklós, *Generating hard instances of lattice problems*, https://dl.acm.org/doi/pdf/10.1145/237814.237838, 1996.

[3] G. J. Simmons, *Symmetric and asymmetric encryption*, https://dl.acm.org/doi/pdf/10.1145/356789.356793, 1979.

[4] D. Boneh, *The decision diflie-hellman problem*, https://link.springer.com/content/pdf/10.1007/BFb0054851.pdf, 2006.

[5] B. K. Ronald L. Rivest, *Rsa problem*, http://people.csail.mit.edu/rivest/RivestKaliski-RSAProblem.pdf, 2003.

[6] P. W. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring*, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=365700, 1994.

[7] V. S. Dan Boneh, *A graduate course in applied cryptography*, https://toc.cryptobook.us/book.pdf, 2020.

[8] V. Lyubashevsky, *Basic lattice cryptography: Encryption and fiat-shamir signatures*, https://drive.google.com/file/d/1JTdW5ryznp-dUBBjN12QbvWz9R41NDGU/view, 2020.

[9] V. L. Carsten Baum Ivan Damgård, *More efficient commitments from structured lattice assumptions*, https://eprint.iacr.org/2016/997.pdf, 2016.

[10] A. S. Amos Fiat, *How to prove yourself: Practical solutions to identification and signature problems*, https://link.springer.com/content/pdf/10.1007/3-540-47721-7_12.pdf, 1987.

[11] I. Damgård, *On Γ-protocols*, https://cs.au.dk/~ivan/Sigma.pdf, 2010.

NTNU

Norwegian University of
Science and Technology