Homomorphism learning problems and its applications to public-key cryptography

Christopher Leonardi[1, 2] and Luis Ruiz-Lopez[1, 2]

[1]University of Waterloo
[2]Isara Corporation

May 23, 2019

Abstract

We present a framework for the study of a learning problem over abstract groups, and introduce a new technique which allows for public-key encryption using generic groups. We proved, however, that in order to obtain a quantum resistant encryption scheme, commutative groups cannot be used to instantiate this protocol.

Keywords: Learning With Errors, isogenies, non-commutative cryptography

1 Introduction

Lattice based cryptography is nowadays the most prominent among the candidate areas for quantum resistant cryptography. The great popularity of lattice based cryptography is, in great part, due to its versatility—several different primitives have been constructed based on lattice problems—and security guarantees such as average-case to worst-case reductions to problems that are presumably hard even for quantum algorithms. Particularly, the *short integers solutions* problem (SIS), used by Ajtai in his seminal paper [1] to construct a one-way function, and the *learning with errors* problem (LWE), introduced by Regev in [18], have served as the backbone for several cryptographic constructions.

The importance of these two problems goes beyond their applications in cryptography, since their formulation was motivated by purely mathematical problems of a mixed geometric and algebraic character. For example, SIS can be thought as the problem of finding short elements in the kernel of a linear function. For its part, LWE is the problem of finding solutions to a system of noisy linear equations. With these statements of the problems it is possible to imagine several generalizations of them, since some elements in the statements may seem rather arbitrary.

In this paper we place the learning with errors problem in a generic framework that allows us to explore the possible versions of it that might be useful for cryptographic applications, and that, in turn, also encompass other hard problems that have appeared in previous constructions. Specifically we interpret LWE as a learning problem in the context of noisy group homomorphisms. By abstracting the notion of noise, we dispense with the need of having a metric defined that is also efficiently computable. We describe a new way to sample a noise that is efficiently erasable, which allows for the generic construction of a public-key cryptosystem.

Motivation The study of a generic version of LWE is motivated by a variety of reasons. The best attacks for LWE are instance-specific—they make use of the fact that the relevant homomorphisms are between the groups $(\mathbb{Z}/q\mathbb{Z})^n$ and $\mathbb{Z}/q\mathbb{Z}$, which are linear functionals described

as an inner product. Therefore, an instance using generic groups may avoid these attacks, and be benefited from having smaller components (keys and ciphertexts) and better performance. Nevertheless, studying the possible abstract meaning of the concepts the concern LWE, such as "learning", "noise" and "rounding", as well as exploring the mathematical aspects involved in this abstractions is an interesting endeavor on its own.

1.1 Generalizations and other variants of LWE

The learning with errors problem has received special attention, and several efforts have been made to improve its efficiency, as a consequence, cryptosystems based on LWE have particularly enjoyed of a large number of improvements and generalizations. In 2009, *polynomial learning with errors* (PLWE) was introduced by Stehlé et al. [20] as a way to optimize computation and key-sizes for the constructions based on LWE, apparently without compromising the practical security of these constructions. Short after, Lyubashevsky et al. [13] independently proposed *ring learning with errors* (RLWE), which further generalizes PLWE by allowing the objects to belong to the ring of integers of a number field. The recently popular *module*-LWE, first introduced in [5] as *general*-LWE, is the generalization of RLWE to a multidimensional module over the same ring—generalizing both, LWE and RLWE. Lastly, *learning with rounding* (LWR)[2] is a variant of the original LWE problem on which the error is sampled deterministically.

There have been several works outlining generalized versions of LWE in different contexts. Short after Regev's introduction of LWE in 2005 [18], Peikert published a work on the hardness of *error-correction in the exponent* [16], on which he proves that, for suitable parameters on the error, *bounded-distance decoding* (BDD) for a black-box cyclic group is, at least, as hard as the discrete logarithm problem on the same group. This work lead to posterior analysis of the learning with errors problem in the exponent by Demarest et al. [9], which generalizes the original formulation of LWE to the problem of decoding over the group C_p^n, and use a new technique to provide a generic lower bound on the number of queries necessary to solve the decoding problem in this group. Independently, Dagdelen et al. studied the same problem in [8], where they describe a relation of this to a generalization of the computational Diffie-Hellman problem.

In [3], Baumslag et al. propose a generalization of LWE to abstract groups by considering the distance in a Cayley graph associated to the group. As the authors mention in the paper, this distance is not always easy to compute, moreover, the problem is known to be NP-complete for certain instances [19]. However, they propose the use of Burnside groups of exponent 3 (denoted as B_3) to instantiate their construction. In a follow-up paper, Fazio et at. [10] make a deeper study of the hardness of this problem on B_3, and provide a worst-case to average-case reduction of this problem, by proving that it is random self-reducible.

Another approach generalizing the learning problem was proposed by Gama et al. in [11]. They generalize the LWE and SIS problems to finite Abelian groups. The authors show that the more general versions of the problems still enjoy the worst-case to average-case reductions that the original formulations have, provided that the instance group is large enough.

Another attempt to use non-commutative groups is described in [6]. In this pre-print Cheng et al. study the learning problem over the group ring $R[G]$, an algebraic structure which consists of formal sums of elements of G with coefficients in R. As a concrete instance they choose $R = \mathbb{Z}$ and $G = D_{2n}$, the dihedral group of order $2n$. Their main motivation is to recreate ring-LWE using a non-commutative group (using integer coefficients) instead of the cyclic group (using coefficients in the integers or in a cyclotomic ring), to avoid attacks on principal ideal lattices.

Lastly, a recent manuscript by Bootland et al. [4] describes a framework in linear algebra that encompasses different problems that have appeared in lattice based cryptography, such as LWE, MLWE and RLWE, as well as in code-based cryptography and the recent constructions modulo Mersenne primes. This framework allows to obtain a generalization of problems such as LWE and SIS by choosing the environment: a parent ring, a ciphertext, modulus and a rank.

1.2 Our work

This paper regards LWE as a learning problem, specifically as a problem of learning homomorphisms between two algebraic objects from noisy samples. The final objective of this generalization is the possible application of the new version of the problem to construct cryptographic primitives. With an ideal generalization one would be able to emulate any construction that uses LWE. However, when trying to recreate the public-key construction described in [18] we face three main challenges which are how to erase the error, how to combine the elements of the public key to encrypt a message, and finding instances on which this problem is hard.

Decryption. The decryption algorithm consists of two main steps. The first step is to subtract the mask using the secret key—in the case of LWE is to subtract $\langle \mathbf{s}, \mathbf{a} \rangle$—resulting in a noisy version of the plaintext—in the case of LWE, $b - \langle \mathbf{s}, \mathbf{a} \rangle = \lfloor \frac{q}{2} \rfloor \mu + e$. The second step is to erase the noise, for this LWE scales down the error to be erasable by rounding the result to the nearest integer. This, however, depends on having a notion of size—a metric—for the elements of the group. For a generic group given by a finite set of generators, there is always a well defined metric—for example, the word metric (referred to as the Cayley distance in [3]). However, in general this distance may not be efficiently computable.

We propose a purely algebraic approach to define the noise. More specifically, the noise is sampled from a secret normal subgroup $N \leq H$ that is subsequently eliminated by projecting onto the quotient H/N. This approach allows for an unbounded number of operations with public-key elements, since the noise does not "accumulate", causing overflows and decryption errors as in LWE based construction.

Encryption. Public-key encryption is achieved by randomly mixing elements from the public key, generating a new uniformly looking sample. Mixing noisy elements by summing a random subset of the public key generates an element with the same structure when the noise elements commute with the image of φ, however, this cannot be guaranteed in general.

Our encryption algorithm works in a similar way, by taking a random word with elements of the public key. When the group is non-Abelian, this does not always result in a sample from the same distribution, however, it is possible to recover the message erasing the error prior to removing the mask. This makes possible the construction of an LWE-like public key cryptosystem using non-Abelian groups.

Failures and what we learned The main roadblock we encountered was finding an appropriate example to instantiate the protocol described in Section 5. Since this construction is an attempt to bring the learning with errors problem to a generic setting, one of the desired properties that we had in mind was quantum resistance, which is one of the properties that have made LWE such an important cryptographic problem in the last decade.

The protocol requires three groups G, H and N, with N a normal subgroup of H, as well as a homomorphism $\varphi\colon G \to H$. The private key consists of φ and N, therefore, to simply avoid exhaustive search attacks, it is necessary that

- the group H has a large number of normal subgroups and
- the set $\mathrm{Hom}(G, H)$ of homomorphisms from G to H is large.

Moreover, the feasibility of the construction depends as well on the ability of sample from a probability distribution whose support is included on the kernel of φ. It is evident that not every group satisfies these restrictions, moreover, finding groups with these desired properties has been a roadblock for this project.

Since every subgroup of an Abelian group is normal, we started our search for examples in Abelian groups. Two different instances for this construction are described in 6.1 and 6.2.

Nevertheless, as it is later explained in Section 8, these examples only serve to illustrate the functioning of the generic encryption scheme, as the fact that the groups used are Abelian make these instances vulnerable to quantum attacks (and a very trivial classical attack in the case of 6.1).

1.3 Organization

The paper is divided into three main parts. Sections 2 and 3 constitute Part 1, Part 2 is formed by sections 4 and 5, and sections 7 and 8 make up part 3.

In the first part we provide outline the required material and fundamental theory. In Section 2 we cover most of the theoretical background that is be necessary to provide the context for the following sections. In Section 3 we discuss the subject of learning a function, and we provide a definition that is appropriate for this case, as well as the definitions of the computational problems related to learning noisy homomorphisms between semigroup.

In the second part of the paper we talk about its potential applications to cryptography. In Section 4 we discuss the difficulties that we find when trying to construct a public-key encryption protocol following Regev's blueprint, and we address an alternative solution. Section 5 is dedicated to explicitly describe the construction of a generic pubic-key cryptosystem and the potential properties that this may have. In Section 6 we make an expository explanation of two instances of the previous construction.

The third and final part of the paper is dedicated to the cryptanalysis of the construction and instances described in the previous sections. Section 7 contains basic generic procedures that may lead to the extraction of information about the secret key, and Section 8 describes two attacks for the specific case where the group that is used is Abelian.

2 Preliminaries

2.1 Groups and Semigroups

A *semigroup* is a set S together with an associative binary operation $\cdot\colon S \times S \to S$, sometimes called the *semigroup law*. An element $e \in S$ is called *identity* if, for all $s \in S$, $s \cdot e = e \cdot s = s$. The identity element is unique in S. Given $s \in S$, an *inverse* of s is an element $s' \in S$ such that $s \cdot s' = s' \cdot s = e$. It follows that for all $s \in S$, the inverse is unique and it is denoted by s^{-1}. A semigroup with an identity element and closed under inverses is called a *group*. A semigroup is *commutative* (or *Abelian*) if, for all $s, s' \in S$, $s \cdot s' = s' \cdot s$. If S is a group, a *subgroup* of S is a subset $H \subseteq S$ closed under the group operation and inverses, and such that $e \in H$. The subgroup relation is denoted as $H \leq S$. The *center* $Z(S)$ of a semigroup S is the set of elements $z \in S$ such that for all s, $zs = sz$. The subgroup *generated* by a collection $\{s_1, \ldots, s_\ell\} \subseteq S$ is the minimum subgroup $\langle s_1, \ldots s_\ell \rangle$ of S containing them. The *order* $O(s)$ of an element $s \in S$ is the cardinality of the group generated by s. A subgroup H of S is *normal* if, for all $s \in S$, $s^{-1}Hs = H$. We denote this relation as $H \trianglelefteq S$. The center is a normal subgroup. The (left) cosets sH of a normal subgroup H of S form a group under the operation $sHs'H := ss'H$. This is called the *quotient group*, and it is denoted as S/H.

Given two semigroups S and S', a mapping $\varphi\colon S \to S'$ is a *semigroup homomorphism* if for all $s, s' \in S$, $\varphi(ss') = \varphi(s)\varphi(s')$. If S and S' are groups (i.e. if possess an identity element and every element has an inverse), then it follows that $\varphi(e_S) = e_{S'}$ and for all $s \in S$, $\varphi(s^{-1}) = \varphi(s)^{-1}$. In this case φ is called a *group homomorphism*. A bijective homomorphism is called an *isomorphism*. If e' is the identity element of S', the *kernel* of a homomorphism is the set $\operatorname{Ker}(\varphi) := \varphi^{-1}(e') \subseteq S$. The kernel of a homomorphism is a normal subgroup of S, moreover, if $\varphi\colon S \to S'$ is a homomorphism, the image of φ is a subgroup of S' isomorphic to $S/\operatorname{Ker}(\varphi)$.

For a subset $\Sigma = \{s_1, \ldots, s_m\}$ of a group S, a *word* on Σ of *length* ℓ is an expression of the form $s_{w_1}^{\sigma_1} \cdots s_{w_\ell}^{\sigma_\ell}$, where ℓ is a non-negative integer and for all $i \in \{1, \ldots, \ell\}$, $w_i \in \{1, \ldots, m\}$ and $\sigma_i = \pm 1$. The *empty word* is defined as the unique word of length 0. In this paper we denote the sequence of indices $w_1, \ldots, w_\ell$ as $\mathbf{w}$, and the word $s_{w_1}^{\sigma_1} \cdots s_{w_\ell}^{\sigma_\ell}$ as $\prod_{\mathbf{w}} s_{w_i}^{\sigma_i}$.

2.2 Rings, fields and polynomials

A *ring* is a set R endowed with two operations—a sum denoted with "+", and a multiplication denoted with "$\cdot$"—such that R is an Abelian group with respect to the sum, and R a semigroup with respect to the product, and such that for any $a, b, c \in R$, $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(b + c) \cdot a = b \cdot a + c \cdot a$. We write R^+ when referring to R as the group under addition, and $R^\times$ when referring to it as a semigroup under multiplication. We say that R is a *commutative ring* if it is an Abelian semigroup with respect to the product. Any ring considered in this paper is commutative and has a product identity. An *ideal* is a subgroup $I \leq R^+$ such that for any $a \in R$, $aI = I$. For an ideal I, the quotient group R/I (with respect to the addition) is a ring, with the multiplication $aI \cdot bI = abI$. In this paper $\mathbb{Z}_n$ denotes the ring $\mathbb{Z}/n\mathbb{Z}$.

A *field* is a ring with a multiplicative identity 1 and such that any element different from 0 has a multiplicative inverse. The *characteristic* of a field is the smallest positive integer c such that $c \cdot 1 = 0$. If such an integer does not exists we say that the characteristic of the field is 0. Moreover, we have that the characteristic of a field either 0 or a prime, hence any field contains $\mathbb{F}_p$ as a subfield, for some prime p. For any prime p and positive integer k there exists a field with exactly p^k elements. This field is unique up to isomorphism, and it is denoted as $\mathbb{F}_{p^k}$.

Given a ring R, let $R[x]$ denote the set of polynomials on x with coefficients in R. The set $R[x]$ forms a ring with the usual addition and multiplication of polynomials. For polynomials $f_1(x), \ldots, f_k(x) \in R[x]$ let $\langle f_1(x), \ldots, f_k(x) \rangle$ denote the additive subgroup generated by $f_1(x), \ldots, f_k(x)$. This subgroup is an ideal of the ring $R[x]$. If $f(x) \in R[x]$ we denote $R[x]/f(x)$ as the quotient $R[x]/\langle f(x) \rangle$.

2.3 Elliptic curves

Let $\mathbb{F}$ be field of characteristic different from 2 or 3. An *elliptic curve* is the set $E(\mathbb{F})$ of solutions to an equation of the form $E : y^2 = x^3 + ax + b$ over $\mathbb{F}$ and an additional identity element. An elliptic curve has a natural associative operation such that it becomes an Abelian group. The identity element is usually referred to as the point at infinity. An *isogeny* over $\mathbb{F}$ is a non-constant map $\varphi \colon E_1(\mathbb{F}) \to E_2(\mathbb{F})$ of the form

$$(x, y) \mapsto \left(\frac{f_1(x)}{g_1(x)}, \frac{f_2(x)}{g_2(x)} y \right)$$

that fixes the point at infinity, where f_1, f_2, g_1, g_2 are polynomials in $\mathbb{F}[x]$. In this case, E_1 and E_2 are called *isogenous*. An isogeny induces a group homomorphism from $E_1(\mathbb{F})$ to $E_2(\mathbb{F})$. The *degree* of an isogeny is $\max\{f_1(x, y), g_1(x, y)\}$. An isogeny is called *separable* if the derivative of $\frac{f_1(x)}{g_1(x)}$ is nonzero.

Not every two curves are isogenous, however, for any prime power q, two elliptic curves E_1, E_2 are isogenous over $\mathbb{F}_q$ if and only if $|E_1(\mathbb{F}_q)| = |E_2(\mathbb{F}_q)|$. Furthermore, given a fixed curve $E_1(\mathbb{F}_q)$ and a subgroup $G \leq E_1(\mathbb{F}_q)$ there exist a curve $E_2(\mathbb{F}_q)$ and a separable isogeny $\varphi \colon E_1 \to E_2$ over $\mathbb{F}_q$ with kernel G, moreover, E_2 and φ are unique up to $\mathbb{F}_q$-isomorphism. This isogeny can be computed from a set of generators of its kernel by using Velu's formulas [21].

The *isogeny problem* is the problem of finding an isogeny between two elliptic curves E_1, E_2 such that $|E_1(\mathbb{F}_q)| = |E_2(\mathbb{F}_q)|$.

2.4 Learning With Errors

Given n, q positive integers and χ a probability distribution over $\mathbb{Z}_q$, for $\mathbf{s} \in \mathbb{Z}_q^n$ let $A_{\mathbf{s},\chi}$ denote the probability distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by sampling $\mathbf{a}$ from $\mathbb{Z}_q^n$ uniformly at random, sampling e form $\mathbb{Z}_q$ according to χ, and outputting the pair $\big(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e\big)$. Distinguishing $A_{\mathbf{s},\chi}$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is called the *decision learning with errors problem*. If q is a prime in $poly(n)$, then this problem is equivalent to LWE.

The main result in [18] is a reduction from the learning with errors problem to variants of classical problems in geometry of numbers, namely the approximation versions of GapSVP and SIVP, where the approximation factor γ is polynomial on the dimension n. GapSVP_γ on a lattice $\mathcal{L}$ refers to the problem of deciding if the magnitude λ_1 of a shortest vector of $\mathcal{L}$ is less than an input value d, or larger than γd; for its part, SIVP_γ refers to the problem of finding n linearly independent vectors in $\mathcal{L}$—where n is the dimension of $\mathcal{L}$—all of which are bounded, in magnitude, by γ times λ_n, where λ_n is the magnitude of the largest vector in the "smallest" basis of $\mathcal{L}$. See [14] and [17] for more precise definitions. It is worth remarking that the proof requires the modulus q to be bounded by a polynomial on n and, if q is a prime number, the learning with errors problem can be reduced to its decision variant.

Both reductions are achieved by constructing a quantum algorithm to sample from a discrete Gaussian distribution of small width over the lattice $\mathcal{L}$. This algorithm repeatedly iterates a classical and a quantum step. The algorithm starts with vectors sampled from a wide Gaussian (which are easy to sample); during the classical step, the algorithm uses the available samples to construct (an approximation to) the Fourier transform of the discrete Gaussian over $\mathcal{L}$, which is then used to solve a CVP instance on the dual lattice $\mathcal{L}^*$ with the help of the LWE oracle. The quantum step uses the classical part in superposition to construct a sampler from a smaller discrete Gaussian distribution on $\mathcal{L}$.

In the same paper, Regev proposed a public-key encryption scheme whose security guarantee is based on the hardness of solving LWE. More precisely, let $\alpha > 0$ and let $\hat{\Psi}$ be the probability distribution over $\mathbb{Z}_q$ resulting from sampling a real value from the Gaussian distribution over $\mathbb{R}$ defined as

$$\rho_s(x) = \exp\big(-\pi(x/s)^2\big),$$

for $s = \alpha q$, rounding the result to the nearest integer and reducing it modulo q. The cryptosystem can be better understood as an adaptation of the following protocol: suppose $\mathbf{s} \in \mathbb{Z}_q^n$ is a shared secret key. To encrypt $\beta \in \{0, 1\}$, sample $a \leftarrow \mathbb{Z}_q^n$ uniformly and $e \leftarrow \mathbb{Z}_q$ according to χ, and output $\big(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s}\rangle + e + \beta\lfloor \frac{q}{2} \rfloor\big)$. To decrypt the ciphertext $(\mathbf{a}, b)$, compute $b - \langle \mathbf{a}, \mathbf{s}\rangle$ and observe if the result is closer to $\frac{q}{2}$ or 0. The decryption scheme returns the correct result if and only if the noise is in the interval $(-\frac{q}{4}, \frac{q}{4})$.

The public key cryptosystem is obtained by publishing "encryptions of zero" and use these to create new LWE samples by adding a random subset of the public key. Strictly speaking, the resulting sample is under a new LWE distribution, since the noise of some of the elements in the public key was added. However, since the result is an LWE sample, we use the procedure described above to decrypt. Below we outline the explicit public key encryption protocol.

KeyGen : Fix a constant $\epsilon > 0$. Let $q \in \{n^2, \ldots, 2n^2\}$ be a prime and let $m = (1+\varepsilon)(n+1)\log q$. Choose $\mathbf{s} \in \mathbb{Z}_q^n$ uniformly at random. For $i \in \{1, \ldots, m\}$, sample $\big(\mathbf{a}_i, b_i\big)$ from $A_{\mathbf{s},\hat{\Psi}}$. The private key is $\mathbf{s}$, and the public key is $\Big\{\big(\mathbf{a}_i, b_i\big) : i \in \{1, \ldots, m\}\Big\}$

Enc : Given a message $\mu \in \{0, 1\}$, choose a random $(r_1, \ldots, r_n) \in \{0, 1\}^n$ and output

$$\left(\sum_{i=1}^{n} r_i\big(\mathbf{a}_i, b_i\big)\right) + \Big(\mathbf{0}, \mu\left\lfloor \frac{q}{2} \right\rfloor\Big) = \left(\sum_{i=1}^{n} r_i\mathbf{a}_i, \mu\left\lfloor \frac{q}{2} \right\rfloor + \sum_{i=1}^{n} r_i b_i\right).$$

Dec : Given a pair $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, compute

$$\mu' = \left\lfloor \frac{2}{q}(b - \langle \mathbf{s}, \mathbf{a} \rangle) \right\rceil .$$

If the error accumulated during the encryption step is less than $\frac{q}{4}$, then the decryption step recovers the original message, hence the width of the Gaussian should be small enough so that decryption errors are unlikely. However, the error should also be large enough to provide security. The security reduction discussed above requires $\alpha > \frac{2\sqrt{n}}{q}$. In current LWE-based cryptosystems, the requirement of q being a prime is dropped for simplicity and efficiency reasons.

3 Learning Homomorphisms

It is well known that a polynomial function $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ of degree n over any field can be uniquely determined provided of $n+1$ input/output pairs $\big(a, p(a)\big)$. We "determine" this function by computing the coefficients $a_0, a_1, \ldots, a_n$ of $p(x)$. With this information we can efficiently compute the polynomial function at any point in the field, hence we can say we "learned" the function. The concept of "learning" a function can thus be thought as the process of acquiring enough information to efficiently simulate the behavior of the function at any point in the domain.

As an example, let V be a linear space over a field $\mathbb{F}$ of dimension n and consider $\mathcal{F} = Hom_{\mathbb{F}}(V, \mathbb{F})$, the set of all linear functions from V to its field of scalars—also called *functionals*. Notice then that, by using the algebraic structure of V, it is possible to learn f given n samples $\big(\mathbf{v}_1, f(\mathbf{v}_1)\big), \ldots, \big(\mathbf{v}_n, f(\mathbf{v}_n)\big)$, provided that $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are linearly independent. This can be done by writing $\mathbf{v}$ in terms of $\mathbf{v}_1, \ldots, \mathbf{v}_n$—as a linear combination $\mathbf{v} = \sum_{i=1}^n b_i \mathbf{v}_i$—computing the inverse of the matrix whose columns are the vectors $\mathbf{v}_i$. Using the linearity of f we obtain $f(\mathbf{v}) = \sum_i b_i f(\mathbf{v}_i)$. Moreover, let $\mathbf{e_1}, \ldots, \mathbf{e}_n$ be the canonical basis and let $s_i := f(\mathbf{e}_i)$—this can be computed using Gaussian elimination. Thus, for $\mathbf{v} = (a_1, \ldots a_n)$ we can write $f(\mathbf{v}) = \sum_{i=1}^n a_i s_i = \langle \mathbf{s}, \mathbf{v} \rangle$, where $\mathbf{s} = (s_1, \ldots, s_n)$. This means that every $f \in \mathcal{F}$ can be expressed as an inner product by a constant vector $\mathbf{s}$, where $\mathbf{s}$ depends only on f and can be found efficiently.

In the case of morphisms between algebraic objects, the precise notion of "learning a morphism" is intrinsically dependent on the model used for the algebraic structures. For instance, assume that we know (the encodings of) a generating set $g_1, \ldots, g_m$ for a group G, as well as (the encodings of) their corresponding images $\varphi(g_1), \ldots, \varphi(g_m)$ under a morphism $\varphi \colon G \to H$. This information uniquely determines the morphism φ, as the value of $\varphi(g)$ for an element $g \in G$ is given by $\prod_{\mathbf{w}} \varphi(g_{w_i})$, where $g = \prod_{\mathbf{w}} g_{w_i}$. However, computing the word $\mathbf{w}$ may be a hard problem in the group G.

A *black-box semigroup* is a finitely generated semigroup S together with an injective encoding function $\mathrm{enc} \colon S \to \{0,1\}^*$ and an oracle $\mathcal{O}$ that returns the encoding result of operations in a predetermined operation set Π, where Π contains, at least, the group law. We say that an algorithm A has *black-box access* to a finitely generated semigroup $S = \langle s_1, \ldots, s_m \rangle$ if it has access to the list of encodings $\{\mathrm{enc}(s_1), \ldots, \mathrm{enc}(s_m)\}$ and input/output access to the oracle $\mathcal{O}$.

Definition 1. Let G and H be finitely generated semigroups and let $\varphi \colon G \to H$ be a homomorphism. Let ξ be a probability distribution over G. Suppose that an algorithm A has black-box access to G and H. We say that an algorithm A *learns* the function φ with respect to ξ from m samples $\big(g_i, \varphi(g_i)h_i\big)$, with $h_i \leftarrow \chi$, if given $g \leftarrow \xi\big(\langle g_1, \ldots, g_m \rangle\big)$, the algorithm A outputs $\varphi(g)$ with non-negligible probability, where $\xi(S)$ denotes the probability distribution ξ restricted to $S \leq G$.

Notice that in the case of noiseless samples, the learning problem reduces to the problem of finding an expression for g in terms of $g_1, \ldots, g_m$. This problem, in the case of semigroups, is called the *constructive semigroup membership* problem. In [7], Childs and Ivanyos proved that the constructive semigroup membership problem has an exponential quantum query lower bound.

Let G and H be groups. Notice that the set $\mathrm{Hom}(G, H)$ of homomorphisms $\varphi\colon G \to H$ is not empty, since the function that maps every element in G to the identity element in H is itself a homomorphism. In general, however, this set may contain several other elements. Let $\varphi \in \mathrm{Hom}(G, H)$ and let $g_1, \ldots, g_m \in G$.

In order to frame this as a computational problem, we shall assume that it is possible to efficiently sample from a probability distribution χ over G. For $\varphi \in \mathrm{Hom}(G, H)$ let $\Gamma^{\xi}_{\varphi,\chi}$ be the probability distribution over $G \times H$ obtained by sampling $g \in G$ according to ξ, $h \in H$ according to χ and outputing $\big(g, \varphi(g)h\big)$. If G is a finite group and ξ is the uniform distribution over G, we will omit ξ and denote $\Gamma^{\xi}_{\varphi,\chi}$ as $\Gamma_{\varphi,\chi}$. The problem of learning φ given samples from $\Gamma^{\chi}_{\varphi,\chi}$ is formally described in the following definition.

Definition 2. Let G and H be finitely generated groups. Let ξ and χ be probability distributions over G and H, respectively. We say that an algorithm $\mathcal{A}$ solves the *learning homomorphism with noise problem* (LHN) for G, H, ξ and χ if for any $\varphi\colon G \to H$, $\mathcal{A}$ is able to learn φ given a set of samples from the distribution $\Gamma^{\xi}_{\varphi,\chi}$ with overwhelming probability.

In the previous definition it is not required for the groups G and H to be finite, as this restriction would leave out several basic examples, such as the integers. Ideally, we would like to have the possibility to consider infinite groups for the distinguishing version of LHN. Nonetheless, the distinguishing versions of hard problems are usually about differentiating a particular distribution from uniform. To consider an infinite group, therefore, we need to replace the uniform distribution with a fixed distribution defined on the group, as the uniform distribution is not defined on infinite sets.

Definition 3. Let G and H be finitely generated groups and fix a probability distribution Ξ over $G \times H$. Let ξ and χ be probability distributions over G and H, respectively. We say that an algorithm $\mathcal{A}$ solves the *distinguishing homomorphism with noise problem* (DHN) for G, H and ξ and χ with respect to Ξ if for any $\varphi\colon G \to H$, $\mathcal{A}$ is able to distinguish the distribution $\Gamma^{\chi}_{\varphi,\chi}$ from the distribution Ξ over $G \times H$.

Remark 4. The homomorphism learning problem is the noiseless case of the previous problem—when the support of the noise distribution is $1 \in H$. As conjugation in a group is an automorphism, the homomorphism learning problem is, in turn, trivially a generalization of the conjugacy problem.

4 Public-key cryptography from LHN

In 2011, Baumslag et al. proposed a generic framework for the study of the problem of learning noisy homomorphisms over abstract groups, using the word norm as their tool to measure noise. From a hard instance of this problem it is easy to derive a symmetric key encryption scheme. The idea is to share a homomorphism $\varphi\colon G \to H$ as the secret key, this allows to recover $e\tau^{\mu}$ from the pair $\big(g, \varphi(g)e\tau^{\mu}\big)$. If τ is large and the noise is small, it is possible to distinguish whether μ is 0 or 1.

Deriving a public-key cryptosystem, however, is significantly more challenging. Using this problem in a way that is similar to the one described in 2.4, requires the group to have certain properties. In a generic language, the idea of the cryptosystem described in 2.4 is to randomly mix samples $\big(g_i, \varphi(g_i)e_i\big) \in G \times H$ from the public key to obtain a new sample (g, h) whose

distribution provides no information about the secret key φ. This allows us to encode a message μ in an element $\tau_\mu \in H$ by "hiding" it in the second coordinate as $(g, h\tau_\mu)$. To recover τ_μ it is enough to compute h from g and the secret key φ. However, h is formed by alternating multiplication of $\varphi(g_{w_i})$ and elements from the error distribution

$$h = \prod_{\mathbf{w}} \varphi(g_{w_i}) e_{w_i} = \varphi(g_{w_1}) e_{w_1} \varphi(g_{w_2}) e_{w_2} \cdots \varphi(g_{w_\ell}) e_{w_\ell}, \tag{1}$$

while g is only related to $g_{w_1} \cdots g_{w_\ell}$, in other words, the error elements are "on the way" in h.

One way to solve this problem is to use private information to erase the errors first. As a concrete example, let K be a group and let $\psi\colon H \to K$ be a second secret homomorphism, and assume that the error distribution over H efficiently samples elements $e \in \mathrm{Ker}(\psi)$. Hence we can erase the error elements by first applying ψ to h to obtain

$$\begin{aligned} \psi(h) &= \psi\big(\varphi(g_{w_1})\big)\psi(e_{w_1})\psi\big(\varphi(g_{w_2})\big)\psi(e_{w_2}) \cdots \psi\big(\varphi(g_{w_\ell})\big)\psi(e_{w_\ell}) \\ &= \psi\big(\varphi(g_{w_1})\big)\psi\big(\varphi(g_{w_2})\big) \cdots \psi\big(\varphi(g_{w_\ell})\big). \end{aligned}$$

Since φ and ψ are group homomorphisms, we may now recover the relation of the second coordinate with g by computing $\psi \circ \psi(g)$. This motivates the following definition.

Definition 5. Let G, H and K be groups and let $\varphi\colon G \to H$, $\psi\colon H \to K$ be group homomorphisms. Let χ be a probability distribution over H whose support is a subset of $\mathrm{Ker}(\psi)$. We say that an algorithm $\mathcal{A}$ solves the *normal-learning homomorphism with noise* problem normal-LHN) if $\mathcal{A}$ is able to learn φ from a set of samples from the distribution $\Gamma_{\varphi,\chi}$.

Notice that if the group H is Abelian—or, more generally, if the errors are sampled from the center of H—Equation 1 can be rewritten as

$$h = \prod_{\mathbf{w}} \varphi(g_{w_i}) \prod_{\mathbf{w}} e_{w_i}.$$

Nevertheless, this may lead to weaknesses in the construction. If the center of H, $Z(H)$, is a proper subgroup, and the projection $H \mapsto H/Z(H)$ is efficiently computable, we are in the case described in Subsection 7.2. This procedure does not provide additional information to an attacker when H is an Abelian group, since the projection onto the quotient yields a trivial distribution $(g, 1)$. However, in Section 8 we describe a more effective way to solve normal-LHN in this case.

5 A Public Key Cryptosystem based on Normal-LHN

In the previous section we argued the possible difficulties when using LHN to obtain cryptographic primitives, and we motivated the definition of normal-LHN based on this discussion, with the possibility of arriving to a general procedure to construct a public-key cryptosystem form a generic group. In this section we describe this procedure. As with constructions based on LWE, we start by describing a symmetric encryption scheme that is later transformed into a public-key encryption scheme using the algebraic properties inherit to LHN. In Section 6 we describe two constructions using different algebraic objects: polynomial ring and elliptic curves. However, in Section 8, we argue why this constructions are insecure in the quantum setting.

Start by recalling that a subgroup $N \leq H$ is normal if and only if it is the kernel of a homomorphism from H. Consider three finitely generated groups G, H and K, and let ξ and χ be probability distributions over G and H respectively such that both distributions can be sampled efficiently.

5.1 A symmetric-key construction

KeyGen(1^λ)**:** Given the security parameter λ, choose $\varphi\colon G \to H$ and $\psi\colon H \to K$ efficiently computable homomorphisms, and $\tau \in H \setminus \mathrm{Ker}(\psi)$. The shared key is a description of φ and ψ, together with the group element τ.

Enc(β)**:** Given a message $\beta \in \{0,1\}$, sample an element g from G according to ξ and h from $\mathrm{Ker}(\psi) \leq H$ according to χ. The encryption of β is $\big(g, \varphi(g)h\tau^\beta\big)$

Dec(g, h')**:** Given a pair $(g, h') \in G \times H$, compute $\nu = \psi\big(\varphi(g)\big)^{-1} \cdot \psi(h')$ and output

$$\beta' = \begin{cases} 0 & \text{if } \nu = 1_K, \\ 1 & \text{if } \nu \neq 1_K. \end{cases}$$

Correctness. Suppose that (g, h') is a correctly formed encryption of $\beta \in \{0,1\}$. Then the intermediate step of the decryption algorithm computes

$$\begin{aligned} \nu &= \psi\big(\varphi(g)\big)^{-1} \cdot \psi(h') \\ &= \psi\big(\varphi(g)\big)^{-1} \cdot \psi\big(\varphi(g)h\tau^\beta\big) \\ &= \psi\big(\varphi(g)\big)^{-1} \cdot \psi\big(\varphi(g)\big) \cdot \psi(h) \cdot \psi(\tau)^\beta \\ &= \psi(\tau)^\beta. \end{aligned}$$

The correctness then follows since τ is not in the kernel of ψ. □

5.2 A public-key construction

KeyGen(1^λ)**:** Given the security parameter λ, choose $\varphi\colon G \to H$ and $\psi\colon H \to K$ efficiently computable homomorphisms, and for $i \in \{1, \ldots, m\}$ compute

$$\big(g_i, \varphi(g_i)h_i\big) \in G \times H,$$

where g_i is sampled from ξ and h_i is sampled from $\mathrm{Ker}(\psi) \leq H$ according to χ. The private key is a description of φ and ψ. The public key is the set

$$\Big\{\big(g_i, \varphi(g_i)h_i\big)\colon i = 1, \ldots, m\Big\} \subseteq G \times H,$$

together with a public element $\tau \in H \setminus \mathrm{Ker}(\psi)$.

Enc(β)**:** Given a message $\beta \in \{0,1\}$, sample a word $\omega = w_1 \cdots w_\ell$ over the indices $\{1, \ldots, m\}$ of length ℓ and compute

$$(g, h') = \left(\prod_{i=1}^{\ell} g_{w_i}, \prod_{i=1}^{\ell} \varphi(g_{w_i})h_{w_i}\right).$$

Then output $(g, h'\tau^\beta)$.

Dec(g, h)**:** Run the decryption procedure described in Subsection 5.1.

Correctness. Suppose that (g, h) is a correctly formed encryption of $\beta \in \{0,1\}$. Then the intermediate step of the decryption algorithm computes

$$\begin{aligned} \nu &= \psi\big(\varphi(g)\big)^{-1} \cdot \psi(h) \\ &= \psi\left(\varphi\left(\textstyle\prod_{i=1}^{\ell} g_{w_i}\right)\right)^{-1} \cdot \psi\left(\left(\textstyle\prod_{i=1}^{\ell} \varphi(g_{w_i})h_{w_i}\right) \cdot \tau^\beta\right) \\ &= \psi\left(\varphi\left(\textstyle\prod_{i=1}^{\ell} g_{w_i}\right)\right)^{-1} \cdot \left(\textstyle\prod_{i=1}^{\ell} \psi\big(\varphi(g_{w_i})\big)\psi(h_{w_i})\right) \cdot \psi(\tau)^\beta \\ &= \psi\left(\varphi\left(\textstyle\prod_{i=1}^{\ell} g_{w_i}\right)\right)^{-1} \cdot \psi\left(\varphi\left(\textstyle\prod_{i=1}^{\ell} g_{w_i}\right)\right) \cdot \psi(\tau)^\beta \\ &= \psi(\tau)^\beta. \end{aligned}$$

The correctness then follows since τ is not in the kernel of ψ. □

5.3 Properties

Despite being inspired in the traditional LWE cryptosystem, there are several differences between this and the construction described in the previous subsection that may yield to different useful properties, as well as different lines of cryptanalysis.

Noise accumulation and decryption errors Due to the geometric nature of LWE, it is necessary to be careful when handling the noise. Large noise yields to decryption errors, which in turn give way to key recovery attacks. Noise may accumulate during encryption, making decryption errors difficult to mitigate—unless an error correcting code is implemented alongside. Moreover, noise accumulation has been the main obstacle for the design of effective homomorphic cryptosystems based on lattices, making it necessary the use of bootstrapping to achieve unbounded depth fully-homomorphic encryption.

A cryptosystem build as in the previous subsection does not suffer from noise accumulation or decryption errors. Elements sampled form the noise distribution χ are all contained in the kernel of the secret homomorphism ψ.

Unbounded homomorphic Suppose that H is a group with non-trivial center Z, and assume that τ is a non-trivial central element of H of order 2 in the set $H \setminus \mathrm{Ker}(\psi)$. Then $\psi(\tau)$ is also a non-trivial central element in the image of ψ. Let β, β' be two messages and (g, h), (g', h') their corresponding encryptions. Then

$$\begin{aligned} hh' &= \left(\textstyle\prod_{\mathbf{w}} \varphi(g_{w_i}) h_{w_i}\right) \tau^{\beta} \left(\textstyle\prod_{\mathbf{w}} \varphi(g_{w'_i}) h_{w'_i}\right) \tau^{\beta'} \\ &= \left(\textstyle\prod_{\mathbf{w}} \varphi(g_{w_i}) h_{w_i}\right) \cdot \left(\textstyle\prod_{\mathbf{w}} \varphi(g_{w'_i}) h_{w'_i}\right) \cdot \tau^{\beta} \cdot \tau^{\beta'} \\ &= \left(\left(\textstyle\prod_{\mathbf{w}} \varphi(g_{w_i}) h_{w_i}\right) \cdot \left(\textstyle\prod_{\mathbf{w}} \varphi(g_{w'_i}) h_{w'_i}\right)\right) \cdot \tau^{\beta'+\beta'}. \end{aligned}$$

It follows that the coordinate-wise product $(g, h) \cdot (g', h') = (gg', hh')$ is a valid encryption of $\beta + \beta'$.

(Potentially) small keys The encryption mechanism used in traditional LWE mixes elements of the public key by taking a random linear combination of them, where the coefficients are in $\{0, 1\}$. Such restriction is necessary in order to keep the noise small. This is, however, not necessary in this case since noise accumulation does not induce decryption errors. In particular, the number of possible linear combinations of elements $g_1, \ldots, g_m$ of an Abelian group increases according to their order. In the case of non-Abelian groups, however, the number of combinations obtained—words in the set $S = \{g_1, \ldots g_m\}$—is strictly greater, and depends on the relations that hold for the set S.

(Potentially) large message space Suppose that a central element $\tau \in Z(G)$ is such that the discrete logarithm can be solved efficienticiently in the group generated by $\psi(\tau)$, then there is a way to modify the decryption procedure in 5.1 to increase the size of the message space. In particular this is true whenever the discrete logarithm is solvable in K. This allows for the message space to be of the size of $O\big(\psi(\tau)\big)$. Notice, however, that this depends on ψ, which is part of the secret key.

6 Obtaining instances

In the previous section we described way to obtain public-key encryption form the normal-LHN problem over a generic group. However, the feasibility of the construction, as well as the security of it, depend on the specific group that is chosen to instantiate it. In this case, the chosen groups

G, H and K, the homomorphisms $\varphi\colon G \to H$, $\psi\colon H \to K$, and the corresponding probability distributions must have certain desired properties.

Large key space. The groups $\mathrm{Hom}(G, H)$ and $\mathrm{Hom}(H, K)$ must be of exponential size on the security parameter.

Feasibility. There is an efficient algorithm to sample from distribution χ. Since the support of χ (the set of elements where χ is non-zero) must be contained in the kernel of ψ, there must be an efficient algorithm to sample from $\mathrm{Ker}(\psi)$.

One way to ensure that the first condition is satisfied is to choose a group G with a large number of normal subgroups, which holds in a trivial manner for Abelian groups. In this section we present two instances of the construction described in Section 4 using Abelian groups. We remark that both constructions are vulnerable to the attacks described in Section 8, moreover, the attack to the first example, the instance using polynomials, does not require the use of a quantum algorithm, rendering the scheme completely insecure. The second condition is slightly more difficult to guarantee since the difficulty of finding the kernel of a homomorphism depends on the way that the homomorphism is described, and this, in general, might be a difficult task. In the following constructions this problem is addressed by describing the homomorphisms through the description of their corresponding kernels.

6.1 A polynomial ring instance

Let $\mathbb{F}$ be a finite field and let $f(x) \in \mathbb{F}[x]$ be a polynomial of degree n. For $g(x) \in \mathbb{F}[x]$ let $[g(x)]$ denote the coset in $R = \mathbb{F}[x]/f(x)$ containing $g(x)$, and let $\overline{g}(x)$ denote the residue of $g(x)$ divided by $f(x)$. Notice that $\overline{g}(x)$ is the unique polynomial of degree less than n in the coset $[g(x)]$. We have that for every $\alpha \in \mathbb{F}$, the function

$$\psi\colon [g(x)] \mapsto \overline{g}(\alpha)$$

is a group homomorphism from the additive group of $R = \mathbb{F}[x]/f(x)$ to the additive group of $\mathbb{F}$. Notice that this is not a ring homomorphism. The kernel of this homomorphism can be described by the set of polynomials in $\mathbb{F}[x]$ of degree less than n that have α as a root,

$$\begin{aligned} \mathrm{Ker}(\psi) &= \big\{[g(x)]\colon g(\alpha) = 0, \deg(g) < n\big\} \\ &= \big\{[(x-\alpha)p(x)]\colon \deg(p) < n-1\big\}. \end{aligned}$$

If $\mathbb{F}$ is a finite field, the previous description yields an efficient procedure to sample from the uniform distribution over $\mathrm{Ker}(\psi)$, by sampling uniformly a polynomial $p(x)$ of degree less than $n-1$ and returning $(x-\alpha)p(x)$.

KeyGen(1^λ)**:** Pick a polynomial $f(x) \in \mathbb{F}[x]$. Choose $\alpha, s_0, \ldots, s_{n-1}$ from the uniform distribution over $\mathbb{F}$ and let $s(x) = \sum_{j=0}^{n-1} s_j x^j$. For $i \in \{1, \ldots, m\}$ choose $a_i(x)$ uniformly from R and $p(x)$ uniformly from the set of polynomials in $\mathbb{F}[x]$ of degree less than $n-1$. Compute

$$b_i(x) = a_i(x)s(x) + p_i(x)(x-\alpha).$$

The private key is the pair $\big(s(x), \alpha\big)$. The public key is the set of pairs $\big(a_i(x), b_i(x)\big)$.

Enc(μ)**:** Given a message μ, encode it as an element of the field $\mathbb{F}$. Choose a random subset $J \subseteq \{1, \ldots, m\}$ and compute the ciphertext

$$\big(a(x), b(x)\big) = \left(\sum_{i \in J} a_i(x), \sum_{i \in J} b_i(x) + \mu\right) \in \mathbb{F}[x] \times \mathbb{F}[x].$$

Dec $\big(a(x), b(x)\big)$**:** Compute $d(x) = b(x) - a(x)s(x)$ and output $\mu' = d(\alpha)$.

6.2 **Isogeny** LWE

Keeping the kernel of a homomorphism secret is the main idea behind isogeny-based cryptography. The isogeny problem is the problem of computing an isogeny between two curves E_1, E_2 just by knowing the equations that describe the curves, provided that this isogeny exists (that the curves are *isogenous*). In constructions such as SIKE [12], it is assumed that this problem remains hard even after giving away the image of two points in the curve (specifically the generators of the 2 or 3 torsion subgroup of E_1).

Let p be a prime number and $\mathbb{F}_{p^2}$ be the field with p^2 elements.

KeyGen(1^λ)**:** For simplicity we divide this section into the isogeny generation and the point generation.

- **Isogenies:** Choose $k_1, k_2 \in \mathbb{Z}_3^n$ uniformly at random. Set $G_0 = [k_1]R_0 + [k_2]S_0 \in E_0[3^n]$, and find a point $H_0 \in E_0[3^n]$ which is independent from G_0. Use G_0 to compute the isogeny $\phi : E_0 \to E_1$ with $\ker(\phi) = \langle G_0 \rangle$, along with $\phi(H_0)$.

  Next, compute a basis $R_1,\ S_1$ for $E_1[3^n]$. Choose $k_3, k_4 \in \mathbb{Z}_3^n$ uniformly at random. Set $G_1 = [k_1]R_1 + [k_2]S_1 \in E_1[3^n]$, and test that G_1 is independent from $\phi(H_0)$. Otherwise choose k_3 and k_4 again and repeat the previous line. Once G_1 and $\phi(H_0)$ are independent, compute the isogeny use G_1 to compute the isogeny $\psi : E_1 \to E_2$ with $\ker(\phi) = \langle G_1 \rangle$.
- **Points:** Construct points $P_1,\ Q_1 \in E_1[2^n]$ such that $\langle P_1,\ Q_1 \rangle = E_1[2^n]$. Choose $2m$ points at random:

$$X_1, \ldots, X_m \in_R E_0(\mathbb{F}_{p^2}),$$

$$Y_1, \ldots, Y_m \in_R \mathrm{Ker}(\psi) \subseteq E_1[3^n].$$

  For each $i \in \{1, \ldots m\}$ compute the image of $X_1, \ldots, X_m$ under ϕ. The public key is $P_1,\ Q_1$ and the tuples $\big(X_i,\ \phi(X_i) + Y_i\big)$, for $i \in \{1, \ldots m\}$. The private key is $k_1, k_2, k_3, k_4 \in \mathbb{Z}_3^n$ and $\psi(P_1),\ \psi(Q_1)$.

Enc(μ)**:** Encode the message μ into $(M_1,\ M_2) \in (\mathbb{Z}_2^m)^2$, where not both M_1 and M_2 are divisible by 2. Choose a random subset $J \subseteq \{1, \ldots, t\}$ and compute the ciphertext:

$$(X, Y) = \left(\sum_{i \in J} X_i,\ \left(\sum_{i \in J} \phi(X_i) + Y_i \right) + [M_1]P_1 + [M_2]Q_1 \right) \in E_0(\mathbb{F}_{p^2}) \times E_1(\mathbb{F}_{p^2}).$$

Dec(X, Y)**:** Given a ciphertext $(X, Y) \in E_0(\mathbb{F}_{p^2}) \times E_1(\mathbb{F}_{p^2})$, compute

$$Z = \psi\big(Y - \phi(X)\big).$$

Using the knowledge of $\psi(P_1),\ \psi(Q_1)$ solve the two dimensional elliptic curve discrete logarithm problem:

$$Z = [M_1']\psi(P_1) + [M_2']\psi(Q_1)$$

and recover the message M from $(M_1',\ M_2')$.

7 Generic solutions to the morphism learning problem

7.1 Order attack

The *order finding* problem is the problem of finding the order of a group element, given oracle access to the group, where the allowed operations are the group law and inverse. This problem, to the best of our knowledge, is hard to solve classically. A quantum algorithm, however, can

solve the order finding problem by using phase estimation [15]. The solution for this problem is at the core of Shor's algorithm for factoring and solving discrete logarithm over $\mathbb{Z}_n$.

The simplest case to solve the learning problem is when the samples have not been altered by random noise, in other words, where the input of the problem is a collection of samples of the form $\big(g, \varphi(g)\big)$. Let G and H be groups and let $\varphi\colon G \to H$ be a homomorphism. By definition, $\varphi(e_G) = e_H$. Then, for $g \in G$, the order of g, $O(g)$, is bounded below by the order of $\varphi(g)$. Moreover, $O(g)$ is a multiple of $O\big(\varphi(g)\big)$. Hence, given a collection of samples $\big(g_i, \varphi(g_i)\big) \in G \times H$, an attacker can distinguish this distribution from $U(G \times H)$, by observing that the order of the left coordinate is always a multiple of the order of the right coordinate.

7.2 Noise in a known normal subgroup

In [3], the authors remark that, for the distribution $\Gamma_{\varphi,\chi}$ to be indistinguishable from $U(G \times H)$, the support of φ should not be contained in a proper normal subgroup of H, as otherwise an attacker can "factor out" this subgroup obtain a noiseless distribution, on which the attacker can perform the order attack previously described to distinguish it from the uniform distribution. In more detail, let $N \trianglelefteq H$ be a normal subgroup of H containing the support of φ. Then the mapping

$$\bar{\varphi}\colon g \mapsto \varphi(g)N$$

is a homomorphism $\bar{\varphi}$ from G to the quotient group H/N. The distribution $\big(g, \bar{\varphi}(g)\big)$ is a noiseless distribution over $G \times H/N$.

Notice that in order to define $\bar{\varphi}$, and to be able to perform operations in the group H/N, it is necessary to know what the group N is. Therefore, performing this attack requires the knowledge of the normal subgroup on which the support of the noise is contained.

8 Solving normal-LHN for Abelian Groups

In this section we prove the impossibility of constructing a quantum-resistant cryptosystem based on the hardness of normal-LHN for Abelian groups. As a warm-up, we start by recalling the standard way to reduce LWE to SIS. Suppose that we are given m LWE samples $\big(\mathbf{a}_i, b_i = \langle \mathbf{a}_i, \mathbf{s}\rangle + e_i\big)$. Finding the secret $\mathbf{s}$ is equivalent to solving the equation

$$A\mathbf{s} + \mathbf{e} = \mathbf{b},$$

where the matrix A and the vectors $\mathbf{e}$ and $\mathbf{b}$ are formed with the entries of the samples. To solve this one may try to "get rid of the action of $\mathbf{s}$" by computing a vector $\mathbf{t}$ in the null-space of A^T and multiplying $\mathbf{b}$ by $\mathbf{t}^T$. This way we obtain

$$\mathbf{t}^T\mathbf{b} = \mathbf{t}^T A\mathbf{s} + \mathbf{t}^T\mathbf{e} = \mathbf{t}^T\mathbf{e}.$$

When $\mathbf{t}$ is a small vector, the product $\mathbf{t}^T\mathbf{e}$ is also small hence it is possible to solve the decisional version of LWE.

The previous idea can also be used to solve LHN in the case of Abelian groups. When the number of samples in the public key exceeds the rank of the group it is possible to mount a key-recovery attack from the public key. In the other case, when the number of samples that constitute the public-key is less than or equal to the rank of the group, it is possible to recover a message from any encryption of it. Observe that any group homomorphism is constant on the cosets of its kernel; hence a group homomorphism is a hiding function of its kernel.

8.1 Secret key recovery

Let G, H, K be Abelian groups (denoted additively) and let $\varphi\colon G \to H$ and $\psi\colon H \to K$ be two secret homomorphisms. Let ℓ be the rank of G and suppose that we are given $m > \ell$ samples of the form

$$\big(g_i, \varphi(g_i) + h_i\big) \in G \times H$$

with $h_i \in \mathrm{Ker}(\psi)$. Now consider the map $f\colon \mathbb{Z}^m \to G$ given by

$$f\colon (a_1, \ldots, a_m) \mapsto \sum_{i=1}^{m} a_i g_i \in G.$$

This map is a group homomorphism. Using Shor's algorithm it is possible to find a generating set for the kernel of f. If $(a_1, \ldots, a_m) \in \mathrm{Ker}(f)$, we have that

$$\sum_{i=1}^{m} a_i\big(g_i, \varphi(g_i) + h_i\big) = \left(0, \sum_{i=1}^{m} a_i h_i\right) \in \{0\} \times \ker(\psi),$$

obtaining a random element in $\ker \psi$. By repeating this process we can obtain a generating set of $\ker \psi$.

8.2 Message recovery

Suppose that we have the same setup as before, but this time $m \leq \ell$. Let $\big\{\big(g_i, \varphi(g_i) + h_i\big) : i = 1, \ldots, m\big\}$ be the public key and let $(g, h) = \sum_{i=1}^{m} r_i\big(g_i, \varphi(g_i) + h_i\big) + (0, \beta\tau)$ be an encryption of β. Consider the function $f\colon \mathbb{Z}^{m+1} \to G$ given by

$$f\colon (a_1, \ldots, a_m, a_{m+1}) \mapsto -a_{i+1} g + \sum_{i=1}^{m} a_i g_i.$$

As before, this is a group homomorphism. Using Shor's algorithm it is possible to find a generating set for the kernel of f; moreover, this has rank one and is generated by the tuple $(r_1, \ldots, r_m, 1)$. Using these recovered coefficients and the public key, it is possible to recover $\beta\tau$ from the given ciphertext.

9 Unexplored Paths and Future Work

Cryptanalysis Like any new problem, the learning homomorphism with noise problem requires a deeper understanding to be used for cryptographic applications. However, the problem is stated generically, hence every instance of this should be studied separately. In Section 8 we describe quantum procedures to recover the secret key or plaintext by using Shor's algorithm, whenever an Abelian group is used to instantiate the normal-LHN problem. The classical hardness of this problem, however, still remains unaddressed.

Relation to other problems As mentioned before, the popularity of cryptosystems based on LWE and SIS started with the average-case to worst-case reductions from well-known problems in geometry of numbers to these. Unfortunately, since normal-LHN lacks the geometric aspect, we do not expect the existence of a relation with this to some abstraction of GapSVP or SIVP that preserves the geometric idea. Nevertheless, the algebraic nature of this problem opens the possibility to find a relation to classical problems in group theory such as the hidden subgroup problem. It is worth recalling here that the general LHN immediately generalizes the conjugacy problem.

Non-Abelian instances The constructions described in Section 6 make use of Abelian groups—polynomial rings and elliptic curves—to instantiate the normal-LHN problem. Nonetheless, one interesting aspect of the algebraic approach described in Section 4 is the ability to use non-Abelian groups, doing away with the need for an efficiently computable metric in the group. While searching for instances of the normal-LHN problem, there are some properties that one must keep in mind.

References

[1] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM.

[2] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 719–737, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[3] Gilbert Baumslag, Nelly Fazio, Antonio R. Nicolosi, Vladimir Shpilrain, and William E. Skeith. Generalized learning problems and applications to non-commutative cryptography. In Xavier Boyen and Xiaofeng Chen, editors, *Provable Security*, pages 324–339, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[4] Carl Bootland, Wouter Castryck, Alan Szepieniec, and Frederik Vercauteren. A framework for cryptographic problems from linear algebra. Cryptology ePrint Archive, Report 2019/282, 2019. `https://eprint.iacr.org/2019/282`.

[5] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.

[6] Qi Cheng, Jun Zhang, and Jincheng Zhuang. Lwe from non-commutative group rings. Cryptology ePrint Archive, Report 2016/1169, 2016. `https://eprint.iacr.org/2016/1169`.

[7] Andrew M. Childs and Gábor Ivanyos. Quantum computation of discrete logarithms in semigroups. *J. Mathematical Cryptology*, 8:405–416, 2014.

[8] Özgür Dagdelen, Sebastian Gajek, and Florian Göpfert. Learning with errors in the exponent. In Soonhak Kwon and Aaram Yun, editors, *Information Security and Cryptology - ICISC 2015*, pages 69–84, Cham, 2016. Springer International Publishing.

[9] Luke Demarest, Benjamin Fuller, and Alexander Russell. Handling correlated errors: Hardness of lwe in the exponent. Cryptology ePrint Archive, Report 2018/1005, 2018. `https://eprint.iacr.org/2018/1005`.

[10] Nelly Fazio, Kevin Iga, Antonio R. Nicolosi, Ludovic Perret, and William E. Skeith. Hardness of learning problems over burnside groups of exponent 3. *Designs, Codes and Cryptography*, 75(1):59–70, Apr 2015.

[11] Nicolas Gama, Malika Izabachène, Phong Q. Nguyen, and Xiang Xie. Structural lattice reduction: Generalized worst-case to average-case reductions and homomorphic cryptosystems. In *Proceedings, Part II, of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9666*, pages 528–558, New York, NY, USA, 2016. Springer-Verlag New York, Inc.

[12] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular isogeny key encapsulation. Supporting documentation for the NIST PQC project submission, 2017. `https://www.cs.ru.nl/~jrenes/publications/sike.pdf`.

[13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'10, pages 1–23, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, mar 2002.

[15] Michele Mosca. *Quantum Computer Algorithms*. PhD thesis, Oxford University, 1999. http://cacr.uwaterloo.ca/ mmosca/moscathesis.ps.

[16] Chris Peikert. On error correction in the exponent. In Tal Halevi, Shaicand Rabin, editor, *Theory of Cryptography*, pages 167–183, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[17] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10:283–424, 03 2016.

[18] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[19] Vitaly Roman'kov, Alexei Miasnikov, Alexander Ushakov, and Anatoly Vershik. The word and geodesic problems in free solvable groups. *Transactions of the American Mathematical Society*, 362:4655–4682, 01 2010.

[20] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, pages 617–635, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[21] Jacques Vel'u. Isogénies entre courbes elliptiques. *Comptes rendus de l'Académie des Sciences, Paris, Serie A*, 273:238–241, 1971.