A Framework for Cryptographic Problems from Linear Algebra

Carl Bootland[1], Wouter Castryck[1,2], Alan Szepieniec[1], Frederik Vercauteren[1]

[1] imec-COSIC, Department of Electrical Engineering, KU Leuven
Kasteelpark Arenberg 10, 3001 Heverlee, Belgium
firstname.lastname@kuleuven.be

[2] Section of Algebra, Department of Mathematics, KU Leuven
Celestijnenlaan 200B, 3001 Heverlee, Belgium

Abstract. We introduce a general framework encompassing the main hard problems emerging in lattice-based cryptography, which naturally includes the recently proposed Mersenne prime cryptosystem, but also code-based cryptography. The framework allows to easily instantiate new hard problems and to automatically construct post-quantum secure primitives from them. As a first basic application, we introduce two new hard problems and the corresponding encryption schemes.

Concretely, we study generalizations of hard problems such as SIS, LWE and NTRU to free modules over quotients of $\mathbb{Z}[X]$ by ideals of the form (f, g), where f is a monic polynomial and $g \in \mathbb{Z}[X]$ is a ciphertext modulus coprime to f. For trivial modules (i.e. of rank one) the case $f = X^n + 1$ and $g = q \in \mathbb{Z}_{>0}$ corresponds to ring-LWE, ring-SIS and NTRU, while the choices $f = X^n - 1$ and $g = X - 2$ essentially cover the recently proposed Mersenne prime cryptosystems. At the other extreme, when considering modules of large rank and letting $\deg f = 1$ one recovers the framework of LWE and SIS.

Keywords: LWE, SIS, NTRU, quotient ring, post-quantum

1 Introduction

Lattice-based and code-based cryptography are rapidly emerging as leading contenders for generating public-key cryptosystems that promise to withstand quantum attacks. The popularity of these branches of cryptography are due in large part to the simplicity and efficiency of their designs, but is certainly underscored by their strong security guarantees. Two hard problems in particular, the Short Integer Solution (SIS) [3] and Learning With Errors (LWE) [37] problems, stand out in this regard. While these hard problems are expressible in the language of simple linear algebra over finite rings, and are hence easy to use, they are also provably hard-on-average, assuming the worst-case hardness of certain problems in lattices.

In response to the quadratic scaling of both operational cost and memory associated with a full matrix representation, many proposals switch to using structured matrices [26,38,27]. In essence, random matrices are replaced by matrices of multiplication by elements of the ring $R_q = \mathbb{Z}[X]/(f(X), q)$ resulting in the ring-based versions ring-SIS (RSIS) and ring-LWE (RLWE) respectively. Similar worst-to-average case reductions apply here, albeit from problems in structured lattices, which are potentially easier. Nevertheless, the low bandwidth requirements and high speed made possible by the designs from this category make their deployment an attractive option, and this in turn mandates careful study.

Some recent constructions have similar features to these ring-based cryptosystems, but rely on modular big integer arithmetic rather than arithmetic involving polynomials. We classify the AJPS cryptosystem [1] and the I-RLWE cryptosystem of Gu [16] as members of this category, as well as several submissions to the NIST PQC project [35] such as Ramstake [39] and ThreeBears [17]. Despite relying on different types of rings, the underlying mechanisms of both categories bear a striking resemblance to each other in that a notion of 'smallness' of elements is preserved under addition and multiplication operations. This operational similarity suggests the possibility of a unifying perspective and a generic framework for design and analysis.

This paper vastly generalizes the above setting by replacing the *ciphertext ring* R_q by a quotient ring of the form $R_g = \mathbb{Z}[X]/(f(X), g(X))$ with $f, g \in \mathbb{Z}[X]$ and some restrictions on which pairs one can take. This description captures both the familiar RLWE setting where $g = q \in \mathbb{Z}_{>0}$ as well as the big integer arithmetic cryptosystems since when $g(X) = X - b$ for some integer b, we have $(f(X), g(X)) = (f(b), X - b)$ so that $R_g = \mathbb{Z}[X]/(f(b), X - b) \cong \mathbb{Z}/(f(b))$. As such, our framework contains both RLWE and AJPS as special cases. To capture plain LWE and module-LWE we will eventually work with free modules over R_g.

On top of the well-known examples it should be clear that our framework will contain many more, possibly hard, problems that can be consider for use in cryptographic applications. A systematic treatment of the exact hardness of these problems would divert attention away from our current focus, hence we defer such analysis to a future work.

1.1 A motivating example

To identify some of the problems we face in this more general setting, consider the following standard noisy key agreement protocol. Let $\mathbf{G} \in R_g$ be a public parameter, typically sampled uniformly at random or generated pseudorandomly from a short seed. Alice samples two small elements $\mathbf{a}, \mathbf{b} \in R_g$ and Bob does the same for $\mathbf{c}, \mathbf{d}$. They then exchange $\mathbf{aG} + \mathbf{b}$ and $\mathbf{cG} + \mathbf{d}$, thus allowing Alice to obtain $\mathbf{a}(\mathbf{cG} + \mathbf{d})$ and Bob to obtain $\mathbf{c}(\mathbf{aG} + \mathbf{b})$ while thwarting any passive eavesdropper. If $\mathbf{ad} - \mathbf{cb}$ is small, then in principle Alice can obtain secret key material identical to Bob's by correcting the errors or extracting an identical template, possibly with the aid of some additional reconciliation data.

Several requirements are needed to make this protocol work. 1) The representation of elements of R_g must be conducive to efficient computation. 2) Sampling small elements must be possible and moreover, whenever $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ are small then so is $\mathbf{ad} - \mathbf{cb}$. 3) The adversary must be unable to obtain $(\mathbf{a}, \mathbf{b})$ from $(\mathbf{G}, \mathbf{aG} + \mathbf{b})$

or $(\mathbf{c}, \mathbf{d})$ from $(\mathbf{G}, \mathbf{cG} + \mathbf{d})$. 4) It must be possible to correct small perturbations like $\mathbf{ad} - \mathbf{cb}$ or at least tolerate them somehow.

These conditions have been studied extensively in the standard case where $g = q \in \mathbb{Z}_{>0}$. This paper initiates the study of these same conditions in our more general setting. As mentioned above, we view the ciphertext ring R_g as the quotient of the *parent ring* $R := \mathbb{Z}[X]/(f(X))$ by the ideal gR. The parent ring is used to define smallness: informally, a small element of R_g is the reduction modulo g of an element of the parent ring having small coordinates (in absolute value) with respect to the *power basis* $1, X, X^2, \ldots, X^{\deg f - 1}$. Furthermore, when computing in R_g, all variables are to be reduced into a set of representatives $\mathrm{Rep}(R_g)$, see Section 2.2 for details; this forces noisy expressions to wrap around, so that they become hard to distinguish from random expressions. Against this framework, we will provide a thorough analysis of key points 1) and 2), thereby providing a new set of tools for the cryptographer's toolbox that are useful for various specific applications. Condition 3) will be addressed in a future work while condition 4) will be discussed only superficially as it has a more ad hoc flavour.

2 A Recipe for Generating Problems

In this section we present a general recipe for concocting problems on which to build cryptosystems. The recipe is given as a number of decisions to be taken before ending up with a problem. When following this recipe it is instructive to think of having a fixed amount of resources (informally this amount is the size of the problem) to allocate to the different ingredients. Here we simply state the choices to be made and do not attempt to answer the more difficult question of how to make the most appetising dish.

Throughout this section, we look at what choices are made in five different cases. Firstly, we start with plain LWE. Secondly ring-LWE together with module-LWE are examined. Thirdly, we consider the problem underlying the NTRU Prime cryptosystem from [6]. Next, we have the problems underlying the two Mersenne prime cryptosystems due to Aggarwal, Joux, Prakash and Santha [1,2]. Finally, we take an example from coding theory, that of the McEliece cryptosystem [29].

2.1 Select the parent ring

The first choice one needs to make is the monic polynomial $f \in \mathbb{Z}[X]$ defining the parent ring $R = \mathbb{Z}[X]/(f)$. If we denote the degree of f by $n \geq 1$, then choosing a larger n requires allotting more of our resources to this ingredient, furthermore the size of the coefficients of f also affects the consumption of resources; one should keep these small in general so that condition 2) holds. The parent ring naturally carries the structure of a free $\mathbb{Z}$-module with (power) basis $1, X, \ldots, X^{n-1}$.

Running example 1 (plain LWE). Here f is taken to be a linear polynomial, the most obvious choice being $f = X$, so that $R = \mathbb{Z}[X]/(f) \cong \mathbb{Z}$. In this case we use the least amount resources possible.

Running example 2 (ring-LWE and module-LWE). Here we let f be irreducible, so that $R = \mathbb{Z}[X]/(f)$ is an order in a number field.[1]

Running example 3 (NTRU Prime). The NTRU Prime cryptosystem sets n be an odd prime and takes $f = X^n - X - 1$, an irreducible polynomial.

Running example 4 (AJPS). The Mersenne prime cryptosystem lets $f = X^n - 1$ be such that $f(2) = 2^n - 1$ is a prime number; note that n is necessarily prime as well.

Running example 5 (McEliece). As with plain LWE one chooses f to be linear and $R = \mathbb{Z}$.

2.2 Select the ciphertext modulus

Next, we must choose a *ciphertext modulus* $g \in \mathbb{Z}[X]$, which defines the ciphertext ring

$$R_g = \mathbb{Z}[X]/(f, g)$$

in terms of which our problem will be formulated. We impose some restrictions on the possible choice of g; throughout this paper we assume that

(i) f and g are coprime, *i.e.*, their only common divisors are ± 1: this ensures that R_g is a finite ring,
(ii) $\mathsf{deg}(g) < n$, which is not really a restriction since one can always replace g by $g \bmod f$,
(iii) there exists a positive integer a and a monic polynomial $r \in \mathbb{Z}[X]$ such that $(f, g) = (a, r)$ as ideals.

Assumption (iii) is the most restrictive, although not as badly as one might fear: a heuristic proportion of $6/\pi^2 \approx 60.8\%$ of all random pairs f and g satisfies this condition, which is confirmed by experiment (if satisfied then r is linear with overwhelming probability). The reason for (iii) is it ensures that the ciphertext ring naturally comes equipped with a nice set of representatives

$$\mathrm{Rep}(R_g) = \left\{ \alpha_{\mathsf{deg}(r)-1} X^{\mathsf{deg}(r)-1} + \ldots + \alpha_1 X + \alpha_0 \;\middle|\; \alpha_i \in \{0, \ldots, a-1\} \right\}, \quad (1)$$

in which all computations are to be reduced; this ensures condition 1) is satisfied. We stress that having such a nice set of representatives is our *only* reason for

[1] More precisely it is an order in the degree n number field $K = \mathbb{Q}[X]/(f)$. In fact the formal definitions of ring-LWE [27] and module-LWE [22] require R to be the *maximal* such order, denoted by $\mathcal{O}_K$, which may not be true in our setting (if K is not monogenic then this is even impossible). However, allowing for arbitrary orders would needlessly complicate our discussion, the more since there is no issue in the common scenario where f is a cyclotomic polynomial.

this assumption: it would be possible to weaken it if one is willing to end up with uglier or less canonical sets of representatives, though we avoid a detailed discussion. In Section 5 we will explain how to decide if such a and r exist, and if so, how to find them.

Just as with f the degree of g and the size of the coefficients of g plays a role in defining how much resources a certain g uses. In fact, it is better to consider the values of $\deg(r)$ and a as this is what defines the size of R_g: $\#R_g = \mathrm{Res}(f, g) = a^{\deg(r)}$. Increasing this value naturally increases the size of the problem.

Running example 1 (plain LWE). Here g is a positive integer, usually denoted by q, so that $R_g \cong \mathbb{Z}_q$. In this case one can take $a = q$ and $r = f$, hence $\#R_g = q^n$.

Running example 2 (ring-LWE and module-LWE). Here again g is a positive integer q so that one can take $a = q$ and $r = f$.

Running example 3 (NTRU Prime). Again g is a positive integer q and one lets $a = q$ and $r = f$.

Running example 4 (AJPS). Here $g = X - 2$ and one can take $a = 2^n - 1$ and $r = g = X - 2$ because indeed $(X^n - 1, X - 2) = (2^n - 1, X - 2)$. Thus taking $a = 2^n - 1$ and $r = X - 2$ we have $\#R_g = 2^n - 1$.

Running example 5 (McEliece). As with plain LWE we take g to be an integer q, but whereas in plain LWE q is relatively large here we take $q = 2$, thus $\#R_g = 2^n$.

2.3 Select the rank

Thirdly, one must select a positive integer m, the *rank*, and construct the free R_g-module

$$M := R_g^m = \underbrace{R_g \times R_g \times \ldots \times R_g}_{m \text{ copies}}$$

consisting of length m vectors with entries in R_g.

As with n (the degree of f), taking a larger m consumes more resources; indeed the size of a element of M is $m \deg(r) \log |a|$.

Running example 1 (plain LWE). Here m is a reasonably large integer and $M = R_q^m \cong \mathbb{Z}_q^m$.

Running example 2 (ring-LWE and module-LWE). In ring-LWE we take $m = 1$ so that $M = R_q$. In module-LWE $m > 1$ is a relatively small integer and the module M is given by R_q^m.

Running example 3 (NTRU Prime). Here $m = 1$ so that $M = R_q$.

Running example 4 (AJPS). Here again $m = 1$ so that $M = R_{X-2}$.

Running example 5 (McEliece). In this case, the value of m is the dimension of the code used.

2.4 Select the family of hard problems

After choosing the rank we select one of the following three problems, which we call Ideal-LWE, Ideal-SIS, and Ideal-NTRU, respectively. Informally, these problems in their basic form are to solve a system of 'noisy' linear equations, to find a non-zero solution to a system of linear equations which is 'small', and to express a matrix as a quotient of two 'small' matrices, respectively.[2] In each case the base ring is $\mathbb{Z}_q$ for some positive integer q. These basic problems refer to standard LWE, standard SIS and a matrix variant of NTRU, alluded to in [18] when comparing NTRU to McEliece.[3]

The simplest way to generalise these basic problems is to replace the random matrix defining the linear system by a *matrix of multiplication*; that is a linear map on a free $\mathbb{Z}_q$-module defined by multiplying by an element of that module. This gives the matrix some structure allowing for a more compact representation and gives rise to the ring versions of the problems. In particular this gives standard the NTRU problem.

The second main way to generalise the basic problem is to take entries from a larger ring than $\mathbb{Z}_q$, such as the ring R_g, which is a $\mathbb{Z}_a$ module itself.[4] Thus, we can replace the ring elements by $\deg(r) \times \deg(r)$ matrices of multiplication with entries in $\mathbb{Z}_a$ which gives a block structure to the original matrix. This is the general module approach which gives rise to the module variants of the problems when $g = a \in \mathbb{Z}$.

Now we have seen the two main generalisations we give the details of how this can be applied to each problem.

Ideal-LWE For the Ideal-LWE problem one chooses two further parameters k, the number of 'keys', and ℓ, the number of samples (which will depend on the application).[5] The problem is then defined as:

Problem 1 (Ideal-LWE Search Problem). Let χ be a distribution on R defining small elements and let k and ℓ be positive integers. Sample a uniformly random element $\mathbf{s}$ from $R_g^{m\times k}$. The Ideal-LWE search problem is to find $\mathbf{s}$ given the tuple $(\mathbf{a}, \mathbf{b}) \in R_g^{\ell\times m} \times R_g^{\ell\times k}$ where $\mathbf{a} \in R_g^{\ell\times m}$ is sampled uniformly at random and $\mathbf{b} = \mathbf{a} \times \mathbf{s} + \mathbf{e} \in R_g^{\ell\times k}$ with $\mathbf{e}$ sampled from $\chi^{\ell\times k}$.

In a number of circumstances one often wants to sample the secret $\mathbf{s}$ not from the whole space but some subset of elements, for example by sampling it using the error distribution. This so-called 'small secret' case allows more powerful cryptographic constructions to be built as multiplying by $\mathbf{s}$ preserves smallness. See [10, Sect. 4] and [32] for a reduction from the general case to the small secret case.

[2] The definition of what exactly 'small' means and what a distribution of small elements is is left to the next section.

[3] See also [33] where this is elaborated in more detail.

[4] Recall we have $(f, g) = (a, r)$ for some $a \in \mathbb{Z}$.

[5] Often one considers ℓ to be simply polynomially bounded in the security parameter rather than fixed.

Ideal-SIS In the Ideal-SIS and Ideal-NTRU problems we require a norm on the ciphertext ring, $\|\cdot\|\colon R_g \to \mathbb{R}_{\geq 0}$. We abuse notation and write $\|\mathbf{a}\| < \rho$ for $\mathbf{a} \in R_g^m$ if, for all components $\mathbf{a}_i$ of $\mathbf{a}$, the relation $\|\mathbf{a}_i\| < \rho$ holds.

Problem 2 (Ideal-SIS Search Problem). Given an integer $\ell > m$ together with a bound ρ. Sample ℓ elements from $M = R_g^m$ uniformly at random, say $\mathbf{a}_1, \dots, \mathbf{a}_\ell$, then the Ideal-SIS problem is to find a non-zero vector $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_\ell) \in R^\ell$ such that $\|\mathbf{z}\| \leq \rho$ and $\sum_{i=1}^{\ell} \mathbf{a}_i \cdot \mathbf{z}_i = \mathbf{0}$.

One often considers the inhomogeneous problem where instead of finding a linear combination summing to zero one is given a target vector which the linear combination must sum to; this is also sometimes called the knapsack problem.

Ideal-NTRU The final problem we consider is that of the Ideal-NTRU problem.

Problem 3 (Ideal-NTRU Search Problem). Let χ be a distribution of small elements on R with appropriate bound ρ. Sample $\mathbf{u} \leftarrow \chi^{m\times m}$ such that it is invertible in $R_g^{m\times m}$ and $\mathbf{v} \leftarrow \chi^{m\times m}$.[6] Set $\mathbf{h} = \mathbf{v}\mathbf{u}^{-1} \in R_g^{m\times m}$.[7] Then given $\mathbf{h}$ and ρ the Ideal-NTRU search problem is to find a pair $(\mathbf{u}', \mathbf{v}')$ with $\mathbf{u}'$ invertible, $\mathbf{h} = \mathbf{v}'\mathbf{u}'^{-1}$, $\|\mathbf{u}'\| < \rho$ and $\|\mathbf{v}'\| < \rho$.

Unlike with the previous choices the cost of picking a certain problem is not so obvious; one could consider, for example, the size of the space which the solution to the set of linear equations belongs but this is not so easy to compute in the Ideal-SIS and Ideal-NTRU cases when the solution is restricted to be small. We point out that the size of the problem is related but directly equivalent to the hardness of a problem. For most choices of parameters, the best known attacks rely on lattice reduction; hence in general the cost will depend on the dimension of the lattice being reduced which need not directly reflect the size of the problem.

Running example 1 (plain LWE). Here we of course select the Ideal-LWE problem.

Running example 2 (ring-LWE and module-LWE). This again amounts to selecting the Ideal-LWE problem.

Running example 3 (NTRU Prime). Here we select the Ideal-NTRU problem.

Running example 4 (AJPS). The version of [1] amounts to selecting the Ideal-NTRU problem while the corresponding NIST submission [2] amounts to selecting Ideal-LWE.

Running example 5 (McEliece). Here we are considering the problem of decrypting a ciphertext using only the public key. One essentially takes the Ideal-LWE problem with a fixed number of samples (the length of the code).

[6] The case of non-square $\mathbf{v}$ can also be considered.

[7] We also have the choice of multiplying $\mathbf{v}$ on the left by $\mathbf{u}^{-1}$ but this leads to the same problem; however there is a third option: to multiply $\mathbf{v}$ by the inverse of two small square matrices, one on the left and one on the right. This is done in [12].

2.5 Distribution of small elements

Finally, we come to the issue of what a *small element* is. Informally spoken, by a small element of R we mean an element having small coordinates (in absolute value) with respect to the power basis. The archetypal example is that each coordinate is sampled from a discrete Gaussian distribution with standard deviation σ. The LWE type problems all typically use this type of distribution. One can also consider the case when the coefficients are not sampled independently. When σ becomes small enough, the coefficients are, with high probability, in the set $\{-1, 0, 1\}$. When not sampled independently, it becomes possible to essentially sample vectors of a specified Hamming weight, this is the distribution used in the NTRU setting.

The question of precisely how small to take small elements is complex and depends on the problem and application. In general larger errors give harder problems but may inhibit functionality and performance of certain cryptographic schemes.

3 A Catalogue of Problems

Now that we have a general outline for our recipe we can consider what problems we can create using it. To this end we start to build a catalogue of problems by looking at examples already in the literature, a number of which we have already seen.

Ideal-LWE We first consider those using the Ideal-LWE problem. If one takes the ciphertext modulus g to be an integer and set $k = 1$ then we get the familiar LWE type problems: when $\deg(f) = 1$ and $m > 1$ we get standard LWE, when $\deg(f) > 1$ and $m = 1$ we have the (poly-)RLWE problem,[8] and bridging them when $\deg(f) > 1$ and $m > 1$ we find module-LWE. An example for when $k > 1$ is the matrix LWE problem from [7] which still takes g to be an integer.

In contrast, if one takes $g(X) = X - b$ for some integer b and $\mathsf{deg}(f) > 1$, then one obtains LWE-like problems but associated with big integer arithmetic. We identify the I-MLWE problem of ThreeBears [17] ($m > 1$, $k = 1$) and I-RLWE problem of Gu [16] ($m = k = 1$) as members of this class. Further, the Mersenne-756839 submission to NIST [34] defines and uses the Mersenne Low Hamming Combination (MLHC) search problem for security; this is essentially the I-RLWE problem when $b = 2$ and the secret $\mathbf{s}$ is not uniformly random but sampled from the distribution χ. The Ramstake submission [39] also makes use of the MLHC problem.

[8] We note that the RLWE problem is usually stated in terms of the codifferent $R^\vee$ [27,28], but this can be avoided by using a different error distribution [11]. Therefore, we do not consider this option in detail.

Ideal-NTRU Next we consider examples of the Ideal-NTRU problem. When $m = 1$ and $\deg(f) > 1$ we capture standard NTRU [19] along with NTRU Prime [6] and many other variants when taking $g(X)$ an integer as well as the Mersenne Low Hamming Ratio (MLHR) problem [1] when $g(X) = X - 2$. Furthermore, for $m > 1$ and $g \in \mathbb{Z}$ we have the basic matrix formulation of NTRU [33] when $\deg(f) = 1$ while MaTRU [12] uses $\deg(f) > 1$.

Ideal-SIS Finally, with the Ideal-SIS problem there are relatively few examples, all take g to be an integer. When $\deg(f) = 1$ and $m > 1$ we have the standard SIS problem [3], when $\deg(f) > 1$ and $m = 1$ we have the ring-SIS problem [31] and when both $\deg(f) > 1$ and $m > 1$ we reach the module-SIS problem [22]. In the case when both $\deg(f)$ and m are taken to be one, the resulting problem is the (homogeneous) modular subset sum problem (SSP).

We can arrange all of these examples in a number of tables classified by the problem family they utilise, the degrees of f and g as well as whether the rank m is one or larger than one. We colour each cell either red (and mark with a $*$), when we don't consider the problem as $\deg(g) \geq \deg(f)$; yellow, when there is a known example in the current literature; or green (marked with a question mark), when the problem has not yet been considered.

Looking at the green entries in the tables we can immediately see a number of empty entries. Firstly, there seems to be no analogue of NTRU over the integers which appears to be hard; the problem can be solved easily by performing lattice reduction on the 2-dimensional lattice spanned by the row vectors $(1, h)$, $(q, 0)$ and $(q, 0)$ where h is the quotient of small elements in $\mathbb{Z}_q$. Secondly, to the best of our knowledge no one has proposed a matrix version of the NTRU problem over the AJPS ring $\mathbb{Z}[X]/(X^n - 1, X - 2) \cong \mathbb{Z}/(2^n - 1)$. Thirdly, the ring and module variants of the SIS problem have also not been considered when using this ring. Finally, as we have already stated, we know of no paper which explicitly considers the case when the modulus g has degree larger than one.

Cryptographic applications In practice, as cryptographers, our end goal is to build cryptographic schemes which rely on the hardness of a given problem. Just as with deriving a problem by following the above recipe, much of the known cryptographic applications can equally be built almost automatically on top of the new problems in much the same way as when building them from the standard problems. The motivating key-exchange example in the introduction essentially forms the basis for most applications.

In this respect we find that the LWE family is the most useful to us, while the SIS family has the fewest known applications to date.

From the problems belonging to the LWE family we can build basic primitives such as public key encryption [37,36], key exchange [21,5], digital signatures [25,4][9] and oblivious transfer [36,8], as well as more advanced constructs such as identity-based encryption [15] and fully homomorphic encryption [9,14].

[9] See also the NIST competition for more constructions of these three primitives [35].

Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	1-dimensional LWE [10]	RLWE[27]
	1	$*$	I-RLWE [16], MLHC [2]
	$\vdots$	$*$	?

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE, LPN [37], McEliece [29], matrix LWE [7]	M-LWE [22,9]
	1	$*$	I-MLWE [17]
	$\vdots$	$*$	?

Ideal-NTRU

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	?	NTRU [19], NTRU Prime [6]
	1	$*$	MLHR [1]
	$\vdots$	$*$	?

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	matrix NTRU[33]	MaTRU[12]
	1	$*$	?
	$\vdots$	$*$	?

Ideal-SIS

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	modular SSP	RSIS [31]
	1	$*$	?
	$\vdots$	$*$	?

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	SIS [3]	M-SIS[22]
	1	$*$	?
	$\vdots$	$*$	?

As for the NTRU family, there are known constructions for much the same primitives: public key encryption [19,6], digital signatures [20], oblivious transfer [30], identity based encryption [13] and fully homomorphic encryption [24].

The SIS family has turned out to be far less fruitful, however it has still been used to create a digital signature scheme via hashing [15]. It is also known that one can build zero knowledge proofs from the inhomogeneous SIS problem [23].

We expect that most of the above primitives can be straightforwardly adapted to work using our more general problems and we give some simple examples in the case of public key encryption in the next section.

4 New Examples

4.1 Generalising the Gu Encryption scheme to higher degree g

Here we present a generalisation of the Gu encryption scheme [16] where instead of taking g to be linear we consider g of higher degree. We first define our parent ring as $R = \mathbb{Z}[X]/(X^n + 1)$, that is we take $f(X) = X^n + 1$. Next, we carefully choose our ciphertext modulus $g = X^d + b$ where $b > 1$ such that $d \mid n$, $d < n$ and $q = b^{n/d} + (-1)^{n/d}$ is prime.[10] Then we have that the ideal generated by f and g is also generated by g and the prime q; this is because $f = (X^d)^{n/d} + 1 \equiv (-b)^{n/d} + 1 = (-1)^{n/d} q \bmod g$. Therefore we have that $R_{X^d+b} \cong \mathbb{Z}_q^d$ as abelian groups, by considering a polynomial of degree at most $d-1$ as a vector of d coefficients. We will use this as a set of representatives of R_g, see Equation (1). We also take the rank to be one, to simplify the discussion somewhat but one can easily consider a module version of our scheme. Finally, we choose a plaintext modulus p; the plaintext space will be $\mathbb{Z}_p^n$.

Next, we define a distribution of small elements in R, χ_σ, by sampling n coefficients from a discrete Gaussian distribution with standard deviation σ, and forming a polynomial of degree $n-1$ from these coefficients. This polynomial will then be reduced modulo g in our scheme to one with d coefficients, which need not be small with respect to q, indeed we expect them not to be. We denote by $\overline{\chi}_\sigma$ the distribution on $\mathbb{Z}_q^d$ given by sampling from χ_σ and reducing modulo g. In practice, to sample from $\overline{\chi}_\sigma$ one will, for each of the d entries, sample n/d coefficients from the discrete Gaussian, say ϵ_i, and compute $\sum_{i=0}^{n/d-1} \epsilon_i(-b)^i$ as the entry. Thus we see that σ should be much smaller than b.

Key Generation To generate a key we sample an element $\mathbf{a}$ uniformly at random from $R_{X^d+b} \cong \mathbb{Z}_q^d$ as well as elements $\mathbf{s}, \mathbf{e} \leftarrow \overline{\chi}_\sigma$. Compute $\mathbf{b} = \mathbf{as} + p\mathbf{e}$. The public key is the pair $(\mathbf{a}, \mathbf{b})$ while the private key is $\mathbf{s}$.

Encryption Given a plaintext $\mathbf{m} \in \mathbb{Z}_p^n$, consider it as a polynomial in R with coefficients in $[-p/2, p/2)$ and denote by $\overline{\mathbf{m}}$ the reduction of this polynomial modulo $X^d + b$. Sample elements $\mathbf{r}, \mathbf{e_1}, \mathbf{e_2} \leftarrow \overline{\chi}_\sigma$ and compute $\mathbf{c_1} = \mathbf{ar} + p\mathbf{e_1}$ and $\mathbf{c_2} = \mathbf{br} + p\mathbf{e_2} + \overline{\mathbf{m}}$, where $(\mathbf{a}, \mathbf{b})$ is the public key of the intended recipient. The ciphertext is the pair $(\mathbf{c_1}, \mathbf{c_2})$.

[10] If n/d is odd then $b^{n/d} - 1$ is divisible by $b - 1$ so the only way for it to be prime is when $b = 2$ and n/d is prime, hence q must be a Mersenne prime. In our case we want b to be large so we will always require n/d to be even. The choice of n being a power of two gives generalised Fermat primes and we of course require b to be even.

Decryption Given a ciphertext $(\mathbf{c_1}, \mathbf{c_2})$ and a private key $\mathbf{s}$ one first computes $\mathbf{d} = \mathbf{c_2} - \mathbf{c_1}\mathbf{s}$. For each coefficient d_i consider it an integer in $[-q/2, q/2)$ and compute the balanced expansion with base $-b$, say $d_i = \sum_j \alpha_{i,j}(-b)^j$ where $\alpha_{i,j} \in [-b/2, b/2)$. Then for $k = 0, \ldots, n-1$ define $m_k = \alpha_{i,j} \bmod p$ where $i = k \,\mathsf{mod}\, d$ and $j = \lfloor kd/n \rfloor$. Return the vector $\mathbf{m} = (m_k)$.

Security Just as in Theorem 3.9 from [16], for the specific choices of f and g taken here we can convert a RLWE sample with $f = X^n + 1$ and $g = b$ to a Ideal-LWE sample with the same f but $g = X^d + b$ and conversely transform a Ideal-LWE sample into a RLWE sample, in both cases with a growth in the noise present in the sample. The conversions are simple to write down. To go from RLWE to Ideal-LWE, for each polynomial in R_b (i.e. $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{s}$), lift it to a polynomial in R with coefficients in the symmetric interval around zero and then reduce modulo $X^d + b$. In the reverse direction, for each element in R_{X^d+b} with coefficients in the symmetric interval about zero, lift it to a polynomial in R by expanding the coefficients to the base b with the coefficients of powers of b in the range $[-b/2, b/2)$ and then substituting b with $-X^d$. Reduction modulo b gives an element of R_b.

A proof of the reductions is essentially the same as that given in [16] with the same bound on the growth of the noise.

Somewhat Homomorphic Encryption It is straightforward to transform this scheme into a somewhat homomorphic scheme akin to, for example, the Brakerski-Fan-Vercauteren scheme [14]. Implementing this we found that with the same parameters used in practice we could perform on average between zero and three fewer multiplicative levels than with the original scheme.[11]

4.2 Module-NTRU over the AJPS ring

In this section we briefly describe a cryptosystem employing the Ideal-NTRU problem with rank larger than one and which takes as the underlying ring the AJPS ring; this means we will take f as $X^n - 1$ for some prime n such that $q = 2^n - 1$ is also prime, and g as $X - 2$. We also choose positive integers d and $w \ll n$ where d will be the rank of the module used and w will be the Hamming weight of elements sampled from our distribution of small elements. Formally, we define χ_w to be the uniform distribution over the set $\{\sum_{i\in\mathcal{I}} 2^i \mid \mathcal{I} \subset \{0, 1, \ldots, n-1\},\ \#\mathcal{I} = w\}$. The plaintext space will be $\{0,1\}^d$ and for decryption we will choose two thresholds t_l and t_u satisfying $0 \le t_u < t_l \le n$.

Key Generation To generate keys first sample two matrices $\mathbf{u}$ and $\mathbf{v}$ from $\chi_w^{d\times d}$ with the condition that $\mathbf{u}$ is invertible modulo q. Compute $\mathbf{w} = \mathbf{v}\mathbf{u}^{-1}$. The public key is $\mathbf{w}$ and the private key is $\mathbf{u}$.

[11] We dropped the condition that $b^{n/d} + 1$ must be prime for this.

Encryption Given a public key $\mathbf{w}$ and a message $m \in \{0,1\}^d$, denote by $\mathbf{m}$ the $d \times d$ diagonal matrix with the message bits down the diagonal. To encrypt, sample two matrices $\mathbf{r}$ and $\mathbf{e}$ from $\chi_w^{d\times d}$ and a diagonal matrix $\mathbf{d}$ with uniformly random coefficients modulo q. Compute the ciphertext as $\mathbf{c} = \mathbf{rw} + \mathbf{md} + \mathbf{e}$.

Decryption To decrypt the ciphertext $\mathbf{c}$ with the private key $\mathbf{u}$ first compute the product $\mathbf{p} = \mathbf{cu}$. Then for each i in $\{1, \dots, d\}$ consider the elements in the ith row of $\mathbf{p}$ as binary strings of length n and compute the mean of the Hamming weights of these binary strings. If this mean is at most the threshold t_l set $m_i = 0$, if this mean is no smaller than t_u set $m_i = 1$ and otherwise abort. Return the vector (m_i).

Decryption works since we have $\mathbf{p} = \mathbf{cu} = \mathbf{rv} + \mathbf{mdu} + \mathbf{eu}$ and the entries of $\mathbf{rv}$ and $\mathbf{eu}$ will still have relatively small Hamming weight while the entries of $\mathbf{mdu}$ will be zero in the ith row if $m_i = 0$ and be uniformly random if $m_i = 1$. The probability that d uniformly random elements have a mean Hamming weight smaller that the threshold t_l can be made negligibly small by choosing the parameters appropriately.

5 Generic Moduli

In this final section we look at the structure of the ring R_g for generic g. In this case, our ring $R_g = \mathbb{Z}[X]/(f(X), g(X))$ does not have an obvious canonical set of representatives. In order to have useful representatives we will try to find a pair $a \in \mathbb{Z}_{>0}$ and $r \in \mathbb{Z}[X]$ such that $(f, g) = (a, r)$. When r is monic we can use the set of representatives from Eq. (1). We note that if r is not monic then a set of representatives is still possible to write down but is not so user-friendly. Our choice of g will be constrained by R_g having such a set of representatives.

Now our task is to find such a and r, if they exist. It is natural to choose a to be the smallest positive integer in (f, g) so that $(f, g) \cap \mathbb{Z} = (a)$ which always exists due to the coprimality of f and g. Then r is defined only modulo a and up to units of $\mathbb{Z}_a[X]$. The overall strategy is first to find a. Afterwards, we search for an r using the Euclidean algorithm in the ring $\mathbb{Z}_a[X]$. When a is composite, $\mathbb{Z}_a$ is not an integral domain so that finding inverses modulo a can fail. However in this case we will have found a factor of a and can use this factor, with some work, to either split a into a product of coprime factors, work modulo each of these factors and combine the results using the Chinese Remainder Theorem, or write a as a power and use Hensel lifting to find r. Of course these subroutines can also fail when a division fails but we recurse until an r is found. We remark that if we don't assume r exists then it is only possible to determine no r exists during the lifting procedure. This *ad hoc* recursion strategy allows us to bypass the need to factorize a at the onset.

Lemma 1. *Let $s, t \in \mathbb{Z}[X]$ be such that $sf + tg \in \mathbb{Z}$, with $\mathsf{deg}(s) < \mathsf{deg}(g)$ and $\mathsf{deg}(t) < \mathsf{deg}(f)$, and further assume that the greatest common divisor of s and t is 1. Then $a = sf + tg$ is a generator of the ideal $(f, g) \cap \mathbb{Z}$.*

Proof. We proceed by assuming $(f, g) \cap \mathbb{Z}$ is not generated by $sf + tg$ but some proper divisor and derive a contradiction.

For some prime factor p of $sf + gt$ we must have $(sf + gt)/p \in (f, g) \cap \mathbb{Z}$ and thus $(sf + gt)/p = s'f + t'g$ for some $s', t' \in \mathbb{Z}[X]$. We therefore have

$$sf + tg = ps'f + pt'g$$

and rearranging gives $(s - ps')f = (pt' - t)g$. Since f and g are coprime, we must have $s - ps' = kg$ as well as $pt' - t = kf$ for some polynomial $k \in \mathbb{Z}[X]$.

Denote by $\bar{\cdot} : \mathbb{Z}[X] \to \mathbb{F}_p[X]$ reduction modulo p. Then $\bar{k}\bar{g} = \bar{s}$ and $\bar{k}\bar{f} = -\bar{t}$. Since f is monic and $\mathbb{F}_p[X]$ is an integral domain we have $\mathsf{deg}(\bar{t}) < \mathsf{deg}(\bar{f})$ so that $\bar{k}\bar{f} = -\bar{t}$ can only hold if $\bar{k} = \bar{t} = 0$, which implies $\bar{s} = 0$. But $\bar{t} = \bar{s} = 0$ implies p divides both s and t which contradicts the assumption that s and t have greatest common divisor 1. □

The question is thus how to find such s and t. One way to proceed is by computing, using the extended Euclidean algorithm over $\mathbb{Q}[X]$, rational polynomials s' and t' such that $s'f + t'g = 1$ and $\deg(s') < \deg(g)$ and $\deg(t') < \deg(f)$, then multiplying by the lowest common multiple of all the denominators appearing in the coefficients of both s' and t' we find such s and t. The a we require is this lowest common multiple.

Next we show that, when it does not fail, we can use Euclid's algorithm to find r modulo a positive divisor of a. Thus we assume in the lemma that an r exists.

Lemma 2. *Let d be a positive divisor of a and suppose that applying Euclid's algorithm to f and g in the ring $\mathbb{Z}_d[X]$ does not fail and outputs the polynomial ρ. Then $\rho \equiv r \,\mathsf{mod}\, d$ up to units in $\mathbb{Z}_d[X]$.*

Proof. Denote by $\bar{\cdot}$ the residue modulo d. Since $(f, g) = (a, r)$ we have $(\bar{f}, \bar{g}) = (\bar{a}, \bar{r}) = (\bar{r})$ since $d \mid a$. Now by the properties of Euclid's algorithm we have that $(\bar{f}, \bar{g}) = (\rho)$. Therefore $r \equiv \rho \,\mathsf{mod}\, d$ up to a unit of $\mathbb{Z}_d[X]$. □

If d is taken to be a prime p then Euclid's algorithm never fails so we can use it to find a suitable r modulo p. However it is possible that a larger power of the prime divides a, say p^e, and in this case if Euclid's algorithm fails modulo p^e we need to use Hensel lifting to lift ρ, our solution modulo p, to one modulo p^e. Algorithm 1, shows how to do this iteratively from p^j to p^{j+1}. It is at this point where a solution may fail to exist, showing that no such r exists.

Lemma 3. *Algorithm 1 for Hensel lifting is correct.*

Proof. Firstly we assume that ρ' exists. By the preconditions, there exist α, β, and further μ and ν such that $\rho \equiv \alpha f + \beta g$, $f \equiv \mu\rho$ and $g \equiv \nu\rho$ modulo p^j and we write each of these in p-ary form with the subscript indexing the digit, starting at zero. Note that α_0 and β_0 can be computed from f_0 and g_0 using the extended Euclidean algorithm over $\mathbb{F}_p[X]$. Also μ and ν can easily be computed from f, g

Algorithm 1: Hensel Lifting

Input: Polynomials f, g, ρ in $\mathbb{Z}[X]$ (with f monic), a prime p and a positive integer j; satisfying $\alpha f + \beta g \equiv \rho \,\mathsf{mod}\, p^j$ for some $\alpha, \beta \in \mathbb{Z}[X]$, as well as $f \equiv \rho\mu \,\mathsf{mod}\, p^j$ and $g \equiv \rho\nu \,\mathsf{mod}\, p^j$ for some $\mu, \nu \in \mathbb{Z}[X]$.

Output: A polynomial $\rho' \in \mathbb{Z}[X]$ such that $\rho' \equiv \alpha' f + \beta' g \,\mathsf{mod}\, p^{j+1}$ for some $\alpha', \beta' \in \mathbb{Z}[X]$, as well as $\rho' | f$ and $\rho' | g$ in $\mathbb{Z}_{p^{j+1}}[X]$; or Fail if no such polynomial exists.

$\mu \leftarrow f/\rho$ ▷ Arithmetic in $\mathbb{Z}_{p^j}[X]$.
$\nu \leftarrow g/\rho$ ▷ Arithmetic in $\mathbb{Z}_{p^j}[X]$.
$u \leftarrow ((f - \rho\mu)/p^j) \,\mathsf{mod}\, p$ ▷ Thus $f \equiv \rho\mu + p^j u \,\mathsf{mod}\, p^{j+1}$.
$v \leftarrow ((g - \rho\nu)/p^j) \,\mathsf{mod}\, p$ ▷ Thus $g \equiv \rho\nu + p^j v \,\mathsf{mod}\, p^{j+1}$.
$\gamma, \xi, \zeta = \mathsf{xgcd}_{\mathbb{F}_p[X]}(\rho, \mu)$ ▷ Thus $\gamma = \xi\rho + \zeta\mu \,\mathsf{mod}\, p$.
$\delta, \phi, \psi = \mathsf{xgcd}_{\mathbb{F}_p[X]}(\rho, \nu)$ ▷ Thus $\delta = \phi\rho + \psi\nu \,\mathsf{mod}\, p$.
$\theta \leftarrow \zeta\psi(u\nu - v\mu) \,\mathsf{mod}\, p$
$\rho_0 \leftarrow \rho \,\mathsf{mod}\, p$
if $\gamma \nmid u$ **or** $\delta \nmid v$ **or** $\rho_0 \nmid \theta$ **then**
  return Fail
$\kappa \leftarrow (\theta/\rho_0 + \zeta\phi u - \psi\xi v)\tau$
$\rho_j \leftarrow (\zeta u - \kappa\rho_0)/\gamma \,\mathsf{mod}\, \rho_0$ ▷ Hence $\mathsf{deg}(\rho_j) < \mathsf{deg}(\rho_0)$.
$\rho' \leftarrow \rho + p^j \rho_j$ ▷ Arithmetic in $\mathbb{Z}[X]$.
return ρ'

and ρ. Then $f - \rho\mu$ is divisible by p^j, so defining u via $f - \rho\mu = p^j u \,\mathsf{mod}\, p^{j+1}$, ρ_j and μ_j must satisfy

$$0 \equiv f - (\rho + p^j \rho_j)(\mu + p^j \mu_j) \equiv p^j (u - (\rho_j \mu + \rho\mu_j)) \,\mathsf{mod}\, p^{j+1},$$

or equivalently $\rho_j \mu + \rho\mu_j \equiv u \,\mathsf{mod}\, p$. Hence, $u \in (\rho_0, \mu_0) = (\gamma)$ where γ is the greatest common divisor of ρ_0 and μ_0 in $\mathbb{F}_p[X]$, say with Bézout coefficients ξ and ζ so that $\gamma = \xi\rho_0 + \zeta\mu_0$. So γ divides u and all solutions for ρ_j and μ_j are given by

$$\rho_j = \zeta \frac{u}{\gamma} - \kappa \frac{\rho_0}{\gamma} \quad \text{and} \quad \mu_j = \xi \frac{u}{\gamma} + \kappa \frac{\mu_0}{\gamma} \tag{2}$$

for some $\kappa \in \mathbb{F}_p[X]$. The same computation for g implies that δ must divide v where $\delta = \phi\rho_0 + \psi\nu_0$ is the greatest common divisor of ρ_0 and ν_0 over $\mathbb{F}_p[X]$ and $v = (g - \rho\nu)/p^j \,\mathsf{mod}\, p$. The solutions for ρ_j and ν_j are given by

$$\rho_j = \psi \frac{v}{\delta} - \lambda \frac{\rho_0}{\delta} \quad \text{and} \quad \nu_j = \phi \frac{v}{\delta} + \lambda \frac{\nu_0}{\delta} \tag{3}$$

for some $\lambda \in \mathbb{F}_p[X]$. Equating the two expression for ρ_j in Equations (2) and (3) we see that $(\kappa\delta - \lambda\gamma)\rho_0 = \zeta u\delta - \psi v\gamma$. Now using our expressions for γ and δ we have $(\kappa\delta - \lambda\gamma)\rho_0 = (\zeta u\phi - \psi v\xi)\rho_0 + \zeta\psi(u\nu_0 - v\mu_0)$. Thus we must have that ρ_0 divides $\theta := \zeta\psi(u\nu_0 - v\mu_0)$ and then $\kappa\delta - \lambda\gamma = \zeta u\phi - \psi v\xi + \theta/\rho_0$.

Next we note that $\gcd(\gamma, \delta) = 1$ as otherwise there would be a non-trivial factor of μ_0 and ν_0 and then ρ_0 could not be the highest-degree common factor of f and g modulo p. Therefore we can write $1 = \sigma\gamma + \tau\delta$ for some $\sigma, \tau \in \mathbb{F}_p[X]$ and all solutions for κ and λ are given by

$$\kappa = (\theta/\rho_0 + \zeta\phi u - \psi\xi v)\tau + \epsilon\gamma \quad \text{and} \quad \lambda = -(\theta/\rho_0 + \zeta\phi u - \psi\xi v)\sigma + \epsilon\delta$$

for some $\epsilon \in \mathbb{F}_p[X]$ and each such ϵ will give a valid solution. Algorithm 1 chooses to take $\epsilon = 0$ at first but implicitly changes its value later via modular reduction. We find ρ_j by plugging in the expression for κ in Equation (2) then reducing modulo ρ_0. If this modular reduction subtracts $k\rho_0$, then this is equivalent to choosing $\epsilon = k$.

The post-conditions are satisfied because there is a solution for μ_j and ν_j whenever there is one for ρ_j. Setting $\mu' = \mu + \mu_j p^j$ and $\nu' = \nu + \nu_j p^j$ this shows that necessarily $\rho'\mu' = f$ and $\rho'\nu' = g$ in $\mathbb{Z}_{p^{j+1}}[X]$. Moreover, the requirement

$$\rho' = (\alpha + p^j\alpha_j)\rho'\mu' + (\beta + p^j\beta_j)\rho'\nu' \bmod p^{j+1}$$

is equivalent to $w + \alpha_0\mu_j + \alpha_j\mu_0 + \beta_0\nu_j + \beta_j\nu_0 = 0 \bmod p$, where $w = (\alpha\mu + \beta\nu - 1)/p^j \bmod p$ which always has a solution for α_j and β_j as μ_0 and ν_0 are coprime. Therefore, for any such solution, $\alpha' = \alpha + p^j\alpha_j$ and $\beta' = \beta + p^j\beta_j$ satisfy $\rho' = \alpha' f + \beta' g \bmod p^{j+1}$.

The proof up until this point shows that if a ρ_j exists, then Algorithm 1 finds one. Therefore, if the algorithm fails, such a ρ_j does not exist. □

Remark 1. The algorithm can be modified to avoid computing γ, ξ, ζ and δ, ϕ, ψ every iteration as these variables change only when p does. Also, it is possible to output α', β', μ', and ν' along with ρ', if required, but we opted here for brevity and simplicity.

In practice one will not check whether we are working modulo a prime and the requirement that p is a prime in Algorithm 1 and Lemma 3 is there only to guarantee that the various calls to the Euclidean algorithm return a valid result and will not fail. In practice if the Euclidean algorithm fails it will be because it was unable to invert an integer modulo p and hence we will have found a factor of p and can split it appropriately and try again on each factor until it succeeds.

In more detail, if one is working modulo a and finds a factor d then one can find the largest power of d dividing a, say d^k. Then if a/d^k is coprime to d we can work modulo a/d^k and d^k. Otherwise $h = \mathsf{gcd}(a/d^k, d)$ is such that $1 < h < d$ then we find the largest power of h dividing d and the largest power of h dividing a/d^k, say h^l and h^m respectively. Then h^{kl+m} divides a and recurse using factors h^{kl+m}, $(d/h^l)^k$ and $a/(d^k h^m)$ until all factors are coprime. A solution modulo a is then found by using the Chinese Remainder Theorem.

Our calculations (and some heuristics) suggest that $6/\pi^2 \approx 60.8\%$ of all random pairs f and g satisfy this condition, and that r is linear with overwhelming probability in this case. Of the remaining 39.2%, a little over 25% give non-monic r and in just under 14% of the cases no r exists. We leave open the question whether non-monic r can be useful in ways that a monic r cannot.

Acknowledgements

This work was supported in part by the Research Council KU Leuven grants C14/18/067 and STG/17/019. Carl Bootland is funded by an FWO fellowship. Alan Szepieniec is supported by an IWT doctoral grant.

References

1. Aggarwal, D., Joux, A., Prakash, A., Santha, M.: A New Public-Key Cryptosystem via Mersenne Numbers. IACR Cryptology ePrint Archive, Report 2017/481, version 20170530:072202 (2017), https://eprint.iacr.org/eprint-bin/versions.pl?entry=2017/481
2. Aggarwal, D., Joux, A., Prakash, A., Santha, M.: A New Public-Key Cryptosystem via Mersenne Numbers. Cryptology ePrint Archive, Report 2017/481 (2017), https://eprint.iacr.org/2017/481
3. Ajtai, M.: Generating Hard Instances of Lattice Problems (Extended Abstract). In: Miller, G.L. (ed.) STOC 1996. pp. 99–108. ACM (1996)
4. Akleylek, S., Bindel, N., Buchmann, J., Krämer, J., Marson, G.A.: An Efficient Lattice-Based Signature Scheme with Provably Secure Instantiation. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2016. pp. 44–60. Springer International Publishing (2016)
5. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum Key Exchange—A New Hope. In: USENIX Security 16. pp. 327–343. USENIX Association (2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim
6. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU Prime: Reducing Attack Surface at Low Cost. In: Adams, C., Camenisch, J. (eds.) SAC 2017. pp. 235–260. Springer (2018)
7. Bos, J.W., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. In: ACM SIGSAC 2016. pp. 1006–1018 (2016)
8. Brakerski, Z., Döttling, N.: Two-Message Statistically Sender-Private OT from LWE. In: A. Beimel, S.D. (ed.) Theory of Cryptography. TCC '18, Springer (2018)
9. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) Fully Homomorphic Encryption Without Bootstrapping. ACM Trans. Comput. Theory 6(3), 13:1–13:36 (Jul 2014)
10. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical Hardness of Learning with Errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) STOC 2013. pp. 575–584. ACM (2013)
11. Castryck, W., Iliashenko, I., Vercauteren, F.: On error distributions in ring-based LWE. LMS Journal of Computation and Mathematics 19(A), 130–145 (2016)
12. Coglianese, M., Goi, B.M.: MaTRU: A New NTRU-Based Cryptosystem. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. pp. 232–243. Springer Berlin Heidelberg (2005)
13. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient Identity-Based Encryption over NTRU Lattices. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. pp. 22–41. Springer Berlin Heidelberg (2014)
14. Fan, J., Vercauteren, F.: Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive, Report 2012/144 (2012), https://eprint.iacr.org/2012/144

15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for Hard Lattices and New Cryptographic Constructions. In: Dwork, C. (ed.) STOC 2008. pp. 197–206. ACM (2008)
16. Gu, C.: Integer version of ring-LWE and its applications. Cryptology ePrint Archive, Report 2017/641 (2017), https://eprint.iacr.org/2017/641
17. Hamburg, M.: Post-quantum cryptography proposal: ThreeBears (2018), https://sourceforge.net/p/threebears/code/ci/master/tree/threebears-spec.pdf
18. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A New High Speed Public Key Cryptosystem (1996), preliminary draft available at https://web.securityinnovation.com/hubfs/files/ntru-orig.pfd?
19. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J.P. (ed.) Algorithmic Number Theory. pp. 267–288. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
20. Hoffstein, J., Pipher, J., Whyte, W., Zhang, Z.: A Signature Scheme from Learning with Truncation. IACR Cryptology ePrint Archive, Report 2017/995 (2017), https://eprint.iacr.org/2017/995
21. Jintai Ding, Xiang Xie, X.L.: A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem. Cryptology ePrint Archive, Report 2012/688 (2012), https://eprint.iacr.org/2012/688
22. Langlois, A., Stehlé, D.: Worst-Case to Average-Case Reductions for Module Lattices. Designs, Codes and Cryptography 75(3), 565–599 (Jun 2015)
23. Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. pp. 107–124. Springer Berlin Heidelberg (2013)
24. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. In: Karloff, H.J., Pitassi, T. (eds.) STOC 2012. pp. 1219–1234. ACM (2012)
25. Lyubashevsky, V.: Lattice Signatures without Trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. pp. 738–755. Springer Berlin Heidelberg (2012)
26. Lyubashevsky, V., Micciancio, D.: Generalized Compact Knapsacks Are Collision Resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming. pp. 144–155. Springer Berlin Heidelberg (2006)
27. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. pp. 1–23. Springer Berlin Heidelberg (2010)
28. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors Over Rings. IACR Cryptology ePrint Archive, Report 2012/230 (2012), https://eprint.iacr.org/2012/230
29. McEliece, R.J.: A Public-Key Cryptosystem Based on Algebraic Coding Theory. JPL DSN Progress Report 42–44 pp. 114–116 (01 1978)
30. Mi, B., Huang, D., Wan, S., Mi, L., Cao, J.: Oblivious Transfer based on NTRU-Encrypt. IEEE Access pp. 7019–7028 (2018)
31. Micciancio, D.: Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions from Worst-Case Complexity Assumptions. In: FOCS 2002. pp. 356–365. IEEE Computer Society (2002)
32. Micciancio, D.: On the Hardness of Learning With Errors with Binary Secrets (2018), http://cseweb.ucsd.edu/~daniele/papers/BinLWE.pdf
33. Nayak, R., Sastry, C., Pradhan, J.: A matrix formulation for NTRU cryptosystem. In: Proc. 16th IEEE International Conf. on Networks (ICON-2008) (2008)

34. NIST: Post-quantum crypto standardization (2018), http://csrc.nist.gov/groups/ST/post-quantum-crypto/
35. NIST: Submission to the NIST call for PQC proposals. (2018), https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
36. Peikert, C., Vaikuntanathan, V., Waters, B.: A Framework for Efficient and Composable Oblivious Transfer. In: Wagner, D. (ed.) CRYPTO 2008. pp. 554–571. Springer Berlin Heidelberg (2008)
37. Regev, O.: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In: STOC 2005. pp. 84–93. ACM (2005)
38. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. pp. 617–635. Springer Berlin Heidelberg (2009)
39. Szepieniec, A.: Ramstake. Technical report, National Institute of Standards and Technology (2018), https://csrc.nist.gov/CSRC/media/Presentations/Ramstake/images-media/Ramstake-April2018.pdf