Partially Splitting Rings for Faster Lattice-Based Zero-Knowledge Proofs

Vadim Lyubashevsky and Gregor Seiler

IBM Research – Zurich

Abstract. When constructing zero-knowledge proofs based on the hardness of the Ring-LWE or the Ring-SIS problems over the ring $R_p^n = \mathbb{Z}_p[X]/(X^n+1)$, it is often necessary that the challenges come from a set $\mathcal{C}$ that satisfies three properties: the set should be exponentially large (around 2^{256}), the elements in it should have small norms, and all the non-zero elements in the difference set $\mathcal{C}-\mathcal{C}$ should be invertible. The first two properties are straightforward to satisfy, while the third one requires us to make efficiency compromises. We can either work over rings where the polynomial X^n+1 only splits into two irreducible factors modulo p which makes speed of the multiplication operation in R_p^n sub-optimal, or we can limit our challenge set to polynomials of smaller degree which requires them to have (much) larger norms.

In this work we show that one can use the optimal challenge sets $\mathcal{C}$ and still have the polynomial X^n+1 split into more than two factors. For the most common parameters that are used in such zero-knowledge proofs, we show that X^n+1 can split into 8 or 16 irreducible factors. Experimentally, having the rings split in this fashion, increases the speed of polynomial multiplication by around 30%. This is a modest improvement, but it comes completely for free simply by working over the ring R_p^n with a different modulus p. In addition to the speed improvement, the splitting of the ring into more factors is useful in scenarios where one embeds information into the Chinese Remainder representation of the ring elements.

1 Introduction

Cryptography based on the presumed hardness of the Ring / Module-SIS and Ring / Module-LWE problems [PR06,LM06,LPR10,LS15] is seen as a very likely replacement of traditional cryptography after the eventual coming of quantum computing. There already exist very efficient basic public key primitives, such as encryption schemes and digital signatures, based on the hardness of these problems. For added efficiency, most practical lattice-based constructions work over polynomial rings $\mathbb{Z}_p[X]/(f(X))$ where $f(X) = X^n+1$ and p is chosen in such a way that the polynomial X^n+1 splits into n linear factors modulo p. With such a choice of parameters, multiplication in the ring can be performed very efficiently via the Number Theoretic Transform, which is an analogue of the Fast Fourier Transform that works over a finite field. Some examples of practical implementations that utilize NTT implementations of digital signatures and public key encryption based on the Ring-LWE problem can be found in [GLP12,PG13,ADPS16].

The one part of traditional cryptography that does not easily translate into the lattice world is zero-knowledge proofs. Abstractly, in a zero-knowledge proof the prover wants to prove the knowledge of s that satisfies the relation $f(s) = t$, where f and t are public. In the lattice setting, the function

$$f(s) := As \tag{1}$$

where A is a random matrix over some ring (which is commonly $\mathbb{Z}_p$ or $\mathbb{Z}_p[X]/(X^n+1)$) and s is a vector over that same ring, where the coefficients of all (or almost all) the elements comprising s are bounded by some small value $\ll p$.

There are currently three known approaches for constructing lattice-based zero-knowledge proofs when one has a secret s satisfying $f(s) = t$. The first approach is to use the adaptation of the Stern protocol [Ste93] to lattice constructions [KTX08,LNSW13]. The main disadvantage of this approach is that each round of the proof has soundness error $2/3$ and the size of the matrix A used in the proof may grow with the polynomial p. The end result is that these proofs are often Megabytes in length and are unsuitable for most practical applications. Another type of zero-knowledge proof is one in which we would like to simultaneously prove the knowledge of many $s_1, \ldots, s_k$ such that $f(s_i) = t_i$. Recent works showed that such proofs can in fact be quite practical [BDLN16,CDXY17,DL17] when one needs to simultaneously prove a few thousand such equations. A caveat is that rather than proving the knowledge of an s_i, the Prover can only show that he knows some $\bar{s}_i$ whose coefficients are slightly larger.

The third type of zero-knowledge proof is an "approximate" one, where the prover can only show that he knows an $\bar{\mathbf{s}}$ such that $f(\bar{\mathbf{s}}) \approx t$. These are the types of proofs that are affected by our result and we discuss them in greater detail below.

1.1 Approximate Zero-Knowledge Proofs

The function f in (1) satisfies the property that $f(s_1) + f(s_2) = f(s_1 + s_2)$ and for any c in the ring and any vector s over the ring we have $f(sc) = c \cdot f(s)$. The zero-knowledge proof for attempting to prove the knowledge of s proceeds as follows:

The Prover first chooses a "masking parameter" y and sends $w := f(y)$ to the Verifier. The Verifier picks a random challenge c from a subset of the ring and sends it to the prover (in a non-interactive proof, the Prover himself would generate $c := \mathrm{H}(t, w)$, where H is a cryptographic hash function). The Prover then computes $z := sc + y$ and sends it to the Verifier.[1]

The Verifier checks that $f(z) = ct + w$ and, crucially, it also checks to make sure that the coefficients of z are small. If these checks pass, then the Verifier accepts the proof. To show that the protocol is a proof of knowledge, one can rewind the Prover to just after his first move and send a different challenge c', and get a response z' such that $f(z') = c't + w$. Combined with the first response, we extract the equation

$$f(\bar{s}) = \bar{c}t \tag{2}$$

where $\bar{s} = z - z'$ and $\bar{c} = c - c'$.

Notice that while the prover started with the knowledge of an s with small coefficients such that $f(s) = t$, he only ends up proving the knowledge of an $\bar{s}$ with larger coefficients such that $f(\bar{s}) = \bar{c}t$. If $\bar{c}$ also has small coefficients, then this type of proof is good enough for some purposes.

1.2 Applications of Approximate Zero-Knowledge Proofs

As a simple example of the utility of approximate zero-knowledge proofs, we consider commitment schemes where a commitment to a message m involves choosing some randomness r, and outputting $f(s) = t$, where s is defined as $\begin{bmatrix} r \\ m \end{bmatrix}$ where r and m have small coefficients.[2] Using the zero-knowledge

[1] In lattice-based schemes, it is important to keep the coefficients of z small, and so y must be chosen to have small coefficients as well. This can lead to the distribution of z being dependent on sc, which leaks some information about s. This problem is solved in [Lyu09,Lyu12] via various rejection-sampling procedures. How this is done is not important to this paper, and so we ignore this step.

[2] It was shown in [BKLP15,BDOP16] that one actually does not need the message m to have small coefficients, but for simplicity we assume here that it still has them.

proof from Section 1.1, one can prove the knowledge of an $\bar{s}$ and $\bar{c}$ such that $f(\bar{s}) = \bar{c}t$. If $\bar{c}$ is invertible in the ring, then we can argue that this implies that if t is later opened to any valid commitment s' where $f(s') = t$, then it must be $s' = \bar{s}/\bar{c}$.

The sketch of the argument is as follows: If we extract $\bar{s}, \bar{c}$ and the commitment is opened with s' such that $f(s') = t$, then multiplying both sides by $\bar{c}$ results in $f(\bar{c}s') = \bar{c}t$. Combining this with what was extracted from the zero-knowledge proof, we obtain that $f(\bar{c}s') = f(\bar{s})$. If $s' \neq \bar{s}/\bar{c}$, then $\bar{c}s' \neq \bar{s}$ and we found a collision (with small coefficients) for the function f. Such a collision implies a solution to the (Ring-)SIS problem, or, depending on the parameters, may simply not exist (and the scheme can thus be based on (Ring-)LWE).

There are more intricate examples involving commitment schemes (see e.g. [BKLP15,BDOP16]) as well as other applications of such zero knowledge proofs, (e.g. to verifiable encryption [LN17]) which require that the $\bar{c}$ be invertible.

1.3 The Challenge Set and its Effect on the Proof

The challenge c is drawn uniformly from some domain $\mathcal{C}$ which is a subset of R_p^n. In order to have small soundness error, we would like $\mathcal{C}$ to be large. When building non-interactive schemes that should remain secure against quantum computers, one should have $|\mathcal{C}|$ be around 2^{256}. On the other hand, we also would like c to have a small norm. The reason for the latter is that the honest prover computes $z := sc + y$ and so the $\bar{s}$ that is extracted from the Prover in (2) is equal to $z - z'$, and must also therefore depend on $\|sc\|$. Thus, the larger the norms of c, c' are, the larger the extracted solution $\bar{s}$ will be, and the easier the corresponding (Ring-)SIS problem will be.

As a running example, suppose that we're working over the polynomial ring $R_p^{256} = \mathbb{Z}_p[X]/(X^{256}+1)$. If invertibility were not an issue, then a simple and nearly optimal way to choose $\mathcal{C}$ of size 2^{256} would be to define

$$\mathcal{C} = \{c \in R_p^{256} \,:\, \|c\|_\infty = 1, \|c\|_1 = 60\}. \tag{3}$$

In other words, the challenges consist of ring elements consisting of exactly 60 non-zero elements which are ± 1.[3] The l_2 norm of such elements is $\sqrt{60}$.

If we take invertibility into consideration, then we need the difference set $\mathcal{C} - \mathcal{C}$ (excluding 0) to consist only of invertible polynomials. There are some folklore ways of creating such a set. If the polynomial $X^{256} + 1$ splits into k irreducible polynomials modulo p, then all of these polynomials must have degree $256/k$. It is then easy to see, via the Chinese Remainder Theorem that every non-zero polynomial of degree less than $256/k$ is invertible in the ring $\mathbb{Z}_p[X]/(X^{256}+1)$. We can therefore define the set

$$\mathcal{C}' = \{c \in R_p^{256} \,:\, \deg(c) < 256/k,\, \|c\|_\infty \leq \gamma\},$$

where $\gamma \approx 2^{k-1}$ in order for the size of the set to be greater than 2^{256}. The ℓ_2 norm of elements in this set is $\sqrt{256/k} \cdot \gamma$. If we, for example, take $k = 8$, then this norm becomes $\sqrt{32} \cdot 2^7 \approx 724$, which is around 90 times larger than the norms of the challenges in the set defined in (3). It is therefore certainly not advantageous to increase the norm of the challenge by this much only to decrease the running time of the computation. In particular, the security of the scheme will decrease and one will need to increase the ring dimension to compensate, which will in turn negate any savings in

[3] The size of this set is $\binom{256}{60} \cdot 2^{60} > 2^{256}$.

running time. A much more desirable solution would be to have the polynomial $X^n + 1$ split, but still be able to use the optimal challenge set from (3).

1.4 Our Contribution

We show that the polynomial $X^n + 1$ can split into several (in practice up to 8 or 16) irreducible factors and we can still use the optimal challenge sets, like ones of the form from (3). We also show some methods for creating challenge sets that are slightly sub-optimal, but allow for the polynomial to split further. This generalizes a result in [LN17] that showed that one can use the optimal set when $X^n + 1$ splits into two factors.

The simplest way to use our results is via the Theorem below, whose proof is given in Section 3.2. The theorem states that if a non-zero polynomial has small coefficients (where "small" is related to the prime p and the number of prime factors of $X^n + 1$ modulo p), then it's invertible in the ring $\mathbb{Z}_p[X]/(X^n + 1)$.

Theorem 1.1. *Let $n \geq k > 1$ be powers of 2 and $p = 2k + 1 (\mathrm{mod}\, 4k)$ be a prime. Then the polynomial X^n+1 factors as $X^n+1 = \prod_{j=1}^{k} (X^{n/k} - r_j) \bmod p$, and any $\mathbf{y}$ in the ring $\mathbb{Z}_p[X]/(X^n+1)$ that satisfies $0 < \|\mathbf{y}\|_\infty < \frac{1}{\sqrt{k}} \cdot p^{1/k}$ has an inverse in the ring.*

As an application of the above result, suppose that we choose $k = 8$ and a prime p congruent to $17 (\mathrm{mod}\, 32)$ such that $p > 2^{20}$. Furthermore, suppose that we perform our zero-knowledge proofs over the ring $\mathbb{Z}_p[X]/(X^n + 1)$ (where n is a power of 2 greater than 8), and prove the knowledge of $\bar{s}, \bar{c}$ such that $f(\bar{s}) = \bar{c}t$ where $\|\bar{c}\|_\infty \leq 2$ (i.e. the challenges c are taken such that $\|c\|_\infty = 1$). Then the above theorem states that $X^n + 1$ factors into 8 polynomials and $\bar{c}$ will be invertible in the ring since $\frac{1}{\sqrt{8}} \cdot p^{1/8} > 2$.

Having $p > 2^{20}$ is quite normal for the regime of zero-knowledge proofs, and therefore having the polynomial $X^n + 1$ split into 8 factors should be possible in virtually every application. If we would like it to split further into 16 or 32 factors, then we would need $p > 2^{48}$ or, respectively, $p > 2^{112}$. In Section 3.3 we describe how our techniques used to derive Theorem 1.1 can also be used in a somewhat "ad-hoc" fashion to create different challenge sets $\mathcal{C}$ that are nearly-optimal (in terms of the maximal norm), but allow $X^n + 1$ to split with somewhat smaller moduli than implied by Theorem 1.1.

In Section 4, we describe how one would combine the partially-splitting FFT algorithm with the highly optimized polynomial multiplication using the FLINT library [HJP13] to efficiently compute multiplication in a partially-splitting ring. For primes of size approximately 2^{29}, the speed-up of working over rings where $X^n + 1$ splits into 8 versus 2 factors is approximately 30%. It should be noted that even if the ring splits fully, the speed-up obtained by FFT over FLINT (when FLINT is used in a way that takes advantage of the fact that the polynomial moding operation is done by a very sparse polynomial) is only 50%. It is plausible that by using complex optimizations for modern processors (which FLINT does, but our FFT algorithm does not) one could achieve a further 50% speed-up for a full FFT (e.g. as in [ADPS16]) which would also result in further improvements for FFT when $X^n + 1$ is partially-splitting.

In addition to the speed improvement, there are applications whose usability can be improved by the fact that we work over rings R_p^n where X^n+1 splits into more factors. For example, [BKLP15] constructed a commitment scheme and zero-knowledge proofs of knowledge that allows to prove

the fact that $\mathbf{c} = \mathbf{ab}$ when Commit($\mathbf{a}$), Commit($\mathbf{b}$), Commit($\mathbf{c}$) are public (the same holds for addition). An application of this result is the verifiability of circuits. For this application, one only needs commitments of 0's and 1's, thus if we work over a ring where $X^n + 1$ splits into k irreducible factors, one can embed k bits into each Chinese Remainder coefficient of $\mathbf{a}$ and $\mathbf{b}$, and therefore proving that $\mathbf{c} = \mathbf{ab}$ implies that all k multiplications of the bits were performed correctly. Thus the larger k is, the more multiplications one can prove in parallel. Unfortunately k cannot be set too large without ruining the necessary property that the difference of any two distinct challenges is invertible or increasing the ℓ_2-norm of the challenges as described in Section 1.3. Our result therefore allows to prove products of 8 (or 16) commitments in parallel without having to increase the parameters of the scheme to accommodate the larger challenges.

Acknowledgements

We thank Rafaël del Pino for pointing out an improvement to Lemma 3.3. This work is supported by the SNSF ERC Transfer Grant CRETP2-166734 – FELICITY and the H2020 Project Safecrypto.

2 Preliminaries

2.1 Notation

We will denote by R^n the polynomial ring $\mathbb{Z}[X]/(X^n + 1)$ and by R_p^n, the ring $\mathbb{Z}_p[X]/(X^n + 1)$, with the usual polynomial addition and multiplication operations. We will denote by normal letters elements in $\mathbb{Z}$ and by bold letters elements in R^n. For an odd p, an element $\mathbf{w} \in R_p^n$ can always be written as $\sum_{i=0}^{n-1} w_i X^i$ where $|w_i| \leq (p-1)/2$. Using this representation, for $\mathbf{w} \in R_p^n$ (and in R^n), we will define the lengths of elements as

$$\|\mathbf{w}\|_\infty = \max_i |w_i| \text{ and } \|\mathbf{w}\| = \sqrt{\sum_i |w_i|^2}.$$

Notice that in R^n and R_p^n, we have that for any element $\mathbf{w}$, $\|\mathbf{w}\| = \|\mathbf{w}X\|$ and $\|\mathbf{w}\|_\infty = \|\mathbf{w}X\|_\infty$.

2.2 Factorization of $X^n + 1$ Modulo p

It is known that when n is a power of 2, then the polynomial $X^n + 1$ is irreducible over the integers and we will only be considering rings R^n and R_p^n with such n. The lemma below gives the condition on p when $X^n + 1$ splits into n linear terms modulo p.

Lemma 2.1. *If n is a power of two and $p = 1 (\mathrm{mod}\ 2n)$, then $X^n + 1 = \prod_{i=1}^{n} (X - r_i) \bmod p$ where r_i are the n elements in $\mathbb{Z}_p^*$ of multiplicative order $2n$.*

In this work, we will also need to know the necessary conditions that the prime p must satisfy so that the polynomial $X^n + 1$ doesn't fully split, but just splits into k irreducible factors of the form $X^{n/k} - r_j$. These conditions are stated in Theorem 2.5, which requires some lemmas and definitions pertaining to the irreducibility of such polynomials.

Definition 2.2. *Let* $\mathbf{y}$ *be a polynomial in* $\mathbb{Z}[X]$ *with a non-zero constant term. The order of* $\mathbf{y}$ *modulo a prime* p*, denoted* $ord_p(\mathbf{y})$*, is defined as the smallest positive integer* e *such that* $\mathbf{y}$ *divides* $X^e - 1$ *modulo* p*.*

Lemma 2.3. *[LN86, Theorem 3.3, page 75] For any prime* p*,* $ord_p(X - r)$ *is equal to the multiplicative order of* r *in* $\mathbb{Z}_p^*$*.*

Lemma 2.4. *[LN86, Theorem 3.35, page 88] Let* p *be a prime congruent to* $1 (\mathrm{mod}\, 4)$ *and* $X - r$ *be a polynomial such that* $ord_p(X - r) = e$*. Let* t *be an integer whose prime factors divide* e*, but not* $(p - 1)/e$*. Then* $X^t - r$ *is irreducible modulo* p*.*

Theorem 2.5. *If* $n \geq k > 1$ *are powers of two and* p *is a prime congruent to* $2k + 1 (\mathrm{mod}\, 4k)$*, then there exist distinct* $r_i \in Z_p^*$ *such that* $X^n + 1 = \prod_{i=1}^{k} (X^{n/k} - r_i) \bmod p$ *where the polynomials* $X^{n/k} - r_i$ *are irreducible modulo* p*.*

Proof. Because $p = 1 (\mathrm{mod}\, 2k)$, Lemma 2.1 implies that there exist distinct $r_i \in \mathbb{Z}_p^*$ that have multiplicative order $2k$ such that

$$X^k + 1 = \prod_{i=1}^{k} (X - r_i) \bmod p. \tag{4}$$

Since r_i have multiplicative order $2k$ in $\mathbb{Z}_p^*$, Lemma 2.3 says that $ord_p(X - r_i) = 2k$. Let t be any positive power of 2. The prime factors of t (i.e. 2) divide $2k$, yet do not divide $(p-1)/2k$ since the latter is odd. Lemma 2.4 therefore implies that $X^t - r_i$ is irreducible modulo p.

Plugging in $X^{n/k}$ for X in (4), we obtain $X^n + 1 = \prod_{i=1}^{k} (X^{n/k} - r_i) \bmod p$. And because n/k is a power of two, we already proved that $X^{n/k} - r_i$ is irreducible modulo p. □

2.3 Lattices

An integer lattice of dimension n is an additive sub-group of $\mathbb{Z}^n$. For the purposes of this paper, all lattices will be full-rank. The determinant of a full-rank integer lattice Λ of dimension n is the size of the quotient group $|\mathbb{Z}^n / \Lambda|$. If $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are linearly independent vectors in Λ, then $\prod_i \|\mathbf{v}_i\| \geq \det(\Lambda)$. If $\mathbf{z}$ is a non-zero vector in $\mathbb{Z}^n$, then it's easy to see that the lattice

$$\Lambda = \{\mathbf{y} \in \mathbb{Z}^n : \langle \mathbf{y}, \mathbf{z} \rangle \bmod p = 0\}$$

is full-rank and has determinant p. We write $\lambda_1(\Lambda)$ to denote the Euclidean length of the shortest non-zero vector in Λ. Minkowski's Theorem states that for any n-dimensional lattice Λ, $\lambda_1(\Lambda) \leq \sqrt{n} \cdot \det^{1/n}$.

If $\mathcal{I}$ is an ideal in the polynomial ring R^n, then it is also an additive sub-group of $\mathbb{Z}^n$, and therefore an n-dimensional lattice. It is therefore sometimes referred to as an ideal lattice. For an ideal lattice Λ of the ring R^n, in addition to the upper bound on the length of $\lambda_1(\Lambda)$ given by Minkowski's theorem, there also exists a lower bound. The below lemma is well-known, and we sketch its proof for completeness.

Lemma 2.6. *If Λ is an ideal lattice in the ring R^n, then*

$$det^{1/n}(\Lambda) \leq \lambda_1(\Lambda) \leq \sqrt{n} \cdot det^{1/n}(\Lambda).$$

Proof. The upper bound is due to Minkowski's theorem. To prove the lower bound, consider $\mathbf{w}$ to be a polynomial in Λ such that $\|\mathbf{w}\| = \lambda_1(\Lambda)$. Now consider the elements $\mathbf{w}X^i$ for $0 \leq i \leq n-1$. All of these n elements have $\|\mathbf{w}X^i\| = \lambda_1(\Lambda)$, and they are furthermore linearly-independent over $\mathbb{Z}$. The latter is due to the fact that if there exist $a_0, \ldots, a_{n-1} \in \mathbb{Z}$ such that $\mathbf{0} = \sum a_i \mathbf{w}X^i = \mathbf{w} \cdot (\sum_i a_i X^i)$, it implies that the product of two non-zero polynomials in R^n is $\mathbf{0}$, which is impossible because $X^n + 1$ being irreducible over the integers implies that R^n is an integral domain. Since all the $\mathbf{w}X^i$ are linearly independent, we have $\lambda_1(\Lambda)^n = \prod_i |\mathbf{w}X^i| \geq \det(\Lambda)$. □

3 Invertible Elements in Rings

The main goal of this section is to prove Theorem 1.1. To this end, we first prove Lemma 3.1 that can be seen as a special case of the Theorem when the polynomial $X^n + 1$ splits completely modulo p. In Section 3.2 we consider rings R_p^n where $X^n + 1$ only partially splits modulo p and describe how to interpret polynomials $\mathbf{y} \in R_p^n$ as a combination of polynomials $\mathbf{y}_i'$ over a smaller, but fully-splitting ring. We then prove in Lemma 3.2 that if any of the $\mathbf{y}_i'$ is invertible in the fully-splitting ring, then the polynomial $\mathbf{y}$ is invertible in R_p^n. The proof of Theorem 1.1 will follow from these two Lemmas.

3.1 Fully-Splitting Rings

Lemma 3.1. *Let p and k be integers such that $X^k + 1 = \prod_{i=1}^{k} (X - r_i) \bmod p$ for some distinct $r_i \in \mathbb{Z}_p^*$ and let $\mathbf{y}$ be any element in the ring R_p^k. If $0 < \|\mathbf{y}\| < p^{1/k}$, then $\mathbf{y}$ is invertible in R_p^k.*

Proof. Suppose that $\mathbf{y}$ is not invertible in R_p^k. By the Chinese Remainder Theorem, this implies that (for at least) one i, $\mathbf{y} \bmod (X - r_i) = \mathbf{y}(r_i)$ is 0 modulo p. For an i for which $\mathbf{y}(r_i) \bmod p = 0$, (if there is more than one such i, pick one of them arbitrarily) define the set

$$\Lambda = \{\mathbf{z} \in R^k : \mathbf{z}(r_i) \bmod p = 0\}.$$

Notice that Λ is an additive group and for any polynomial $\mathbf{z} \in \Lambda$, the polynomial $\mathbf{z} \cdot X^j \in R^k$ is also in Λ for any integer j. This implies that Λ is an ideal of R^k, and so an ideal lattice in the ring R^k. By Lemma 2.6, we know that $\lambda_1(\Lambda) \geq \det^{1/k}(\Lambda)$.

If we consider the polynomials $\mathbf{z} = \sum_{i=0}^{k-1} z_i X^i \in R^k$ as vectors

$$\mathbf{z} = (z_0, z_1, \ldots, z_{k-1}) \in \mathbb{Z}^k,$$

and define the vector $\mathbf{r} = (1, r_i, r_i^2, \ldots, r_i^{k-1})$, then the lattice Λ can be rewritten as

$$\Lambda = \{\mathbf{z} \in \mathbb{Z}^k : \langle \mathbf{z}, \mathbf{r} \rangle \bmod p = 0\},$$

which implies that $\det(\Lambda) = p$, and so $\lambda_1(\Lambda) \geq p^{1/k}$.

Since we said that $\mathbf{y}(r_i) \bmod p = 0$ and $0 < \|\mathbf{y}\|$, we know that $\mathbf{y}$ is a non-zero vector in Λ. But we also have that $\|\mathbf{y}\| < p^{1/k} \leq \lambda_1(\Lambda)$, which is impossible.

□

At this point, one might be tempted to prove Theorem 1.1 by a simple generalization of Lemma 3.1. The proof sketch would proceed as follows: suppose that n and p are integers such that $X^n+1 = \prod_{j=1}^{k}(X^{n/k} - r_j) \bmod p$ where $(X^{n/k} - r_j)$ are irreducible. Then one can define a lattice

$$\Lambda = \{\mathbf{z} \in R^n : \mathbf{z} \bmod (X^{n/k} - r_j) \bmod p = 0\},$$

and similarly conclude that Λ is an ideal lattice in R^n with $\det(\Lambda) = p^{n/k}$ and $\lambda_1(\Lambda) \geq \det^{1/n}(\Lambda) = p^{1/k}$. This would in turn imply that any polynomial $\mathbf{y} \in R_p^n$ such that $0 < \|\mathbf{y}\| < p^{1/k}$ is invertible. This gives a weaker bound in the ℓ_∞ norm than what is claimed in Theorem 1.1 – we can only conclude that all vectors $\mathbf{y}$ such that $\|\mathbf{y}\|_\infty < \frac{1}{\sqrt{n}} \cdot p^{1/k}$ are invertible. Since n is normally at least 256 and k is a smaller number (like 8), this is a significant difference. In particular, for $k = 8$, rather than having a lower bound $p > 2^{20}$ for the sample application in Section 1.4, we would only obtain $p > 2^{40}$.

Generalizing Lemma 3.1 to rings R_p^n where $X^n + 1$ only "partially splits" is therefore not the right approach for achieving the tightest bounds. In Section 3.2, we instead prove a lemma showing that only some parts of $\mathbf{y}$, which happen to correspond to elements of the smaller ring R_p^k, need to be invertible in R_p^k in order for the entire element $\mathbf{y}$ to be invertible in R_p^n.

3.2 Partially-Splitting Rings

In this section, we will be working with rings R_p^n where p is chosen such that the polynomial X^n+1 factors into k irreducible polynomials of the form $X^{n/k} - r_i$. Theorem 2.5 states the necessary condition on p in order to obtain such a factorization. Throughout this section, we will use the following notation: suppose that $\mathbf{y} = \sum_{j=0}^{n-1} y_j X^j$ is an element of the ring R_p^n, where the value p is chosen as above. Then for all integers $0 \leq i < n/k - 1$, we define the polynomials $\mathbf{y}'_i$ as

$$\mathbf{y}'_i = \sum_{j=0}^{k-1} y_{jn/k+i} X^j. \tag{5}$$

For example, if $n = 8$ and $k = 4$, then for $\mathbf{y} = \sum_{i=0}^{7} y_i X^i$, we have $\mathbf{y}'_0 = y_0 + y_2 X + y_4 X^2 + y_6 X^3$ and $\mathbf{y}'_1 = y_1 + y_3 X + y_5 X^2 + y_7 X^3$.

The intuition behind the definition in (5) is that one can write $\mathbf{y}$ in terms of the $\mathbf{y}'_i$ as

$$\mathbf{y} = \sum_{i=0}^{n/k-1} \mathbf{y}'_i(X^{n/k}) \cdot X^i.$$

Then to calculate $\mathbf{y} \bmod (X^{n/k} - r_j)$ where $(X^{n/k} - r_j)$ is one of the irreducible factors of $X^n + 1$ modulo p, we have

$$\mathbf{y} \bmod (X^{n/k} - r_j) = \sum_{i=0}^{n/k-1} \mathbf{y}'_i(r_j) \cdot X^i \tag{6}$$

simply because we plug in r_j for every $X^{n/k}$.

Lemma 3.2. *Let $n \geq k > 1$ be powers of two such that the polynomial $X^n + 1$ factors as*

$$X^n + 1 = \prod_{j=1}^{k} (X^{n/k} - r_j) \bmod p \tag{7}$$

where $(X^{n/k} - r_j)$ are irreducible modulo p. Let $\mathbf{y}$ be a polynomial in R_p^n and define the associated $\mathbf{y}'_i$ as in (5). If some $\mathbf{y}'_i$ is invertible in R_p^k, then $\mathbf{y}$ is invertible in R_p^n.

Proof. By the Chinese Remainder Theorem, the polynomial $\mathbf{y}$ is invertible in R_p^n if and only if $\mathbf{y} \bmod (X^{n/k} - r_j) \neq 0$ for all $r_1, \ldots, r_k$. From (6), to show that $\mathbf{y}$ is invertible, it is therefore sufficient to show that

$$\exists i \text{ s.t } \forall j, \;\; \mathbf{y}'_i(r_j) \bmod p \neq 0.$$

Let i be such that $\mathbf{y}'_i$ is invertible in the ring R_p^k. From (7), it is clear that

$$X^k + 1 = \prod_{j=1}^{k} (X - r_j) \bmod p,$$

and so the ring R_p^k is fully-splitting. Since $\mathbf{y}'_i$ is invertible in R_p^k, the Chinese Remainder Theorem implies that for all $1 \leq j \leq k$, $\mathbf{y}'_i(r_j) \bmod p \neq 0$, and therefore $\mathbf{y}$ is invertible in R_p^n.

□

Theorem 1.1 now follows from the combination of Theorem 2.5, and Lemmas 3.1 and 3.2.

Proof (Theorem 1.1). For the conditions on n, k, and p, it follows from Theorem 2.5 that the polynomial $X^n + 1$ can be factored into irreducible factors as modulo p as $\prod_{j=1}^{k} (X^{n/k} - r_j)$. For any $\mathbf{y} \in R_p^n$, let the $\mathbf{y}'_i$ be defined as in (5). If $0 < \|\mathbf{y}\|_\infty < \frac{1}{\sqrt{k}} \cdot p^{1/k}$, then because each $\mathbf{y}'_i$ consists of k coefficients, we have that for all i, $\|\mathbf{y}'_i\| < p^{1/k}$. Since $\mathbf{y} \neq 0$, it must be that for some i, $\mathbf{y}'_i \neq 0$. Lemma 3.1 then implies that for that particular i, $\mathbf{y}'_i$ is invertible in R_p^k. In turn, Lemma 3.2 implies that $\mathbf{y}$ is invertible in R_p^n. □

3.3 Example of "Ad-hoc" Applications of Lemma 3.2

Using Lemma 3.2 as we did in the proof of Theorem 1.1 above gives a very clean statement as to a sufficient condition under which polynomials are invertible in a partially-splitting ring. One thing to note is that putting a bound on the ℓ_∞ norm does not take into account the other properties that our challenge space may have. For example, our challenge space in (3) is also sparse, in addition to having the ℓ_∞ norm bounded by 1. Yet we do not know how to use this sparseness to show that one can let $X^n + 1$ split further while still maintaining the invertibility of the set $\mathcal{C} - \mathcal{C}$.

In some cases, however, there are ways to construct challenge sets that are more in line with Lemma 3.2 and will allow further splitting. There does not to be a clean way to systematize these ideas, and so one would have to work out the details on a case-by-case basis. Below, we give such an example for the case in which we are working over R_p^{256} and would like to have the polynomial $X^{256}+1$ split into 16 irreducible factors. If we would like to have X^n+1 split into 16 factors modulo p and the set $\mathcal{C} - \mathcal{C}$ to have elements whose infinity norm is bounded by 2, then applying Theorem 1.1 directly implies that we need to have $2 < \frac{1}{\sqrt{16}} \cdot p^{1/16}$, which implies $p > 2^{48}$.

We will now show how one can lower the requirement on p in order to achieve a split into 16 factors by altering the challenge set $\mathcal{C}$ in (3).

For a polynomial $\mathbf{y} \in R_p^n$, define the $\mathbf{y}'_i$ as in (5). Define $\mathcal{D}$ as

$$\mathcal{D} = \{\mathbf{y} \in R_p^{256} \,:\, \|\mathbf{y}_i\|_\infty = 1 \text{ and } \forall\, 1 \le i \le 16\,, \|\mathbf{y}'_i\| = 2\} \tag{8}$$

In other words, $\mathcal{D}$ is the set of polynomials $\mathbf{y}$, such that every $\mathbf{y}'_i$ has exactly 4 non-zero elements that are ± 1. The size of $\mathcal{D}$ is $\left(\binom{16}{4} \cdot 2^4\right)^{16} \approx 2^{237}$, which should be enough for practical quantum security. The ℓ_2 norm of every element in $\mathcal{D}$ is exactly $\sqrt{64} = 8$. For a fair comparison, we should redefine the set $\mathcal{C}$ so that it also has size 2^{237}. The only change that one must make to the definition in (3) is to lower the ℓ_1 norm to 53 from 60. Thus all elements in $\mathcal{C}$ have ℓ_2 norm $\sqrt{53}$. The elements in set $\mathcal{D}$ therefore have norm that is larger by a factor of about 1.1. It then depends on the application as to whether having $X^n + 1$ split into 16 rather than 8 factors is worth this modest increase. We will now prove that for primes $p > 2^{30.5}$ of a certain form, $X^{256} + 1$ will split into 16 irreducible factors modulo p and all the non-zero elements in $\mathcal{D} - \mathcal{D}$ will be invertible. Therefore if our application calls for a modulus that is larger than $2^{30.5}$ but smaller than 2^{48}, we can use the challenge set $\mathcal{D}$ and the below lemma.

Lemma 3.3. *Suppose that $p > 2^{16 \log_2 \sqrt{14}} \approx 2^{30.5}$ is a prime congruent to $33 (\mathrm{mod}\, 64)$. Then the polynomial $X^{256} + 1$ splits into 16 irreducible polynomials of the form $X^{16} + r_j$ modulo p, and any non-zero polynomial $\mathbf{y} \in \mathcal{D} - \mathcal{D}$ (as defined in (8)) is invertible in the ring $\mathbb{Z}_p[X]/(X^{256} + 1)$.*

Proof. The fact that $X^{256} + 1$ splits into 16 irreducible factors follows directly from Theorem 2.5. Notice that for any $\mathbf{y} \in \mathcal{D} - \mathcal{D}$, the maximum ℓ_2 norm of $\mathbf{y}'_i$ is bounded by 4. Furthermore, the degree of each $\mathbf{y}'_i$ is $256/16 = 16$. Thus an immediate consequence of Lemmas 3.2 and 3.1 is that if $p > 2^{32}$, then any non-zero element in $\mathcal{D} - \mathcal{D}$ is invertible. To slightly improve the lower bound, we can observe that the $\mathbf{y}'_i$ of norm 4 are polynomials in R_p^{16} with exactly four 2's in them. But such elements can be written as a product of 2 and a polynomial with 4 ± 1's in it. So if both of those are invertible, so is the product. The maximum norm of these polynomials is 2 and so they are not the elements that set the lower bound. The next largest element in $\mathcal{D} - \mathcal{D}$ is one that has three 2's and two ± 1's. The norm of such elements is $\sqrt{14}$. Thus for all $p > 2^{16 \cdot \log_2(\sqrt{14})} \approx 2^{30.5}$, the $\mathbf{y}'_i$ will be invertible in R_p^{16}, and thus every non-zero element in $\mathcal{D} - \mathcal{D}$ will be invertible in R_p^{256}. □

4 Polynomial Multiplication Implementation

We now describe in more detail the computational advantage of having the modulus $X^n + 1$ split into as many factors as possible and our experimental results. Our aim is to speed up a general multiplication algorithm provided by the FLINT library [HJP13] by making use of the factorization of the modulus. Suppose that $\mathbb{Z}_p$ contains a fourth root of unity r so that we can write

$$X^n + 1 = (X^{n/2} + r)(X^{n/2} - r).$$

In algebraic language, the FFT (or NTT) is based on the Chinese remainder theorem, which says that $R_p^n = \mathbb{Z}_p[X]/(X^n + 1)$ is isomorphic to the product of $\mathbb{Z}_p[X]/(X^{n/2} + r)$ and $\mathbb{Z}_p[X]/(X^{n/2} - r)$. So, to multiply two polynomials in R_p^n one can first reduce them modulo the two factors of the modulus, then multiply the resulting polynomials in the smaller rings, and finally recombine the product by inverting the Chinese remainder map in order to obtain the product of the original

	Primes				
Number of FFT levels	$2^{20}-2^{14}+1$	$2^{23}-2^{13}+1$	$2^{25}-2^{12}+1$	$2^{27}-2^{11}+1$	$2^{29}-2^{9}+1$
0	29647	32338	34456	36149	44097
1	27021	29553	30610	31927	39035
2	24088	25631	26799	28018	28210
3	24871	26519	27345	28491	27975
4	29730	30702	31967	31622	32710
5	33397	33906	34049	40464	40391
6	24439	24261	24116	24046	24070
7	28016	27973	27778	27811	27943
8	21344	21335	21397	21496	21415

Table 1. CPU cycles used by our FFT-accelerated multiplication algorithm for $\mathbb{Z}_p[X]/(X^{256}+1)$.

polynomials. This is called the FFT-trick (see [Ber01] for a very good survey). Note that reducing a polynomial of degree $n-1$ modulo two sparse polynomials $X^{n/2} \pm r$ is very easy and takes only $\frac{n}{2}-1$ multiplications, $\frac{n}{2}-1$ additions and $\frac{n}{2}-1$ subtractions. If $\mathbb{Z}_p$ contains higher roots so that X^n+1 splits further, then one can apply the FFT-trick recursively to the smaller rings. What is usually referred to as the number theoretic transform (NTT) is the case where $\mathbb{Z}_p$ contains a $2n$-th root of unity so that X^n+1 splits completely into linear factors. This reduces multiplication in R_p^n to just multiplication in $\mathbb{Z}_p$. As we are interested in the case where the modulus does not split completely, we need to be able to multiply in rings of the form $\mathbb{Z}_p[X]/(X^{n/k}-r_i)$ with $k<n$. For this we use the FLINT library [HJP13]. It is the standard back-end for arithmetic of polynomials over finite fields in the Sage computer algebra system. For this purpose, FLINT employs various highly optimized forms of Kronecker substitution.

We have implemented this FFT-acceleration of FLINT using C in both the straight-forward recursive fashion and in an iterative way. In the iterative implementation we first apply the bit reversing permutation to the input polynomials in order to obtain better locality. When X^n+1 splits completely, one usually makes use of the so-called twisting trick and computes the isomorphism

$$X \mapsto rY \colon \mathbb{Z}_p[X]/(X^n+1) \to \mathbb{Z}_p[Y]/(Y^n-1).$$

Then the FFT-trick modulo Y^n-1 is easy to compute as $Y^n-1=(Y^{n/2}-1)(Y^{n/2}+1)$. One can then apply the twisting to $Y^{n/2}+1$ and continue as before. The resulting transform is slightly easier to implement but unfortunately not possible in our case as our ground field lacks the necessary $2n$-th root of unity. In our tests it turned out that the more complex iterative implementation is not better then the recursive one.

If one naïvely uses the general FLINT function nmod_poly_mulmod for the base case multiplication, which implements multiplication of polynomials over $\mathbb{Z}_p$ modulo arbitrary polynomials, then the resulting algorithm will be very slow. The reason is that FLINT does not know how to do the fast reduction modulo our sparse polynomials. Therefore we have used the nmod_poly_mul family of flint functions that provide arithmetic in $\mathbb{Z}_p[X]$ and have implemented our own reduction for the resulting polynomials.

In Table 1 we give the measurements of our experiments. We have performed multiplications in $R_p^{256}=\mathbb{Z}_p[X]/(X^{256}+1)$ for 5 completely splitting primes between 2^{20} and 2^{30}. For each prime we have used between 0 to 7 levels of FFT before using FLINT. 0 levels of FFT means that FLINT was used directly on the input polynomials. On the other hand, in the case of 8 levels of FFT, FLINT

was not used and the corresponding measurements reflect the speed of our full number theoretic transform down to linear factors. As one more example, when performing 3 levels of FFT, we were using FLINT to multiply polynomials of degree 32. The listed numbers are numbers of CPU cycles. They are the medians of 10000 multiplications each. The tests where performed on a computer equipped with an Intel Skylake i7 CPU running at 3.4 GHz. The cycle counter in this CPU ticks at a constant rate of 2.6GHz. As one can see, being able to use a prime p so that $X^n + 1$ splits into more than two factors is clearly advantageous. For instance, by allowing $X^n + 1$ to split into 8 factors compared to just 2, we achieve a speedup of about 30%, which we have measured with the prime $2^{29} - 2^9 + 1$. This should be compared to the only 50% speedup when using the optimal NTT all the way down to 256 linear factors.

It is maybe worth noting that the FLINT library is very highly optimized for modern CPUs, which our basic FFT-implementation is not. Therefore, we expect that one can get larger speedups by, for instance, using a vectorized AVX2-based implementation of the FFT, which would still be a fair comparison with the FLINT library.

The gap between using FLINT after 7 levels of FFT rather than after 6 is explained by the fact that FLINT switches from the Kronecker substitution technique to classical multiplication when multiplying polynomials of degree 4 instead of 8.

References

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In *USENIX*, pages 327–343, 2016.

[BDLN16] Carsten Baum, Ivan Damgård, Kasper Green Larsen, and Michael Nielsen. How to prove knowledge of small secrets. In *CRYPTO*, pages 478–498, 2016.

[BDOP16] Carsten Baum, Ivan Damgård, Sabine Oechsner, and Chris Peikert. Efficient commitments and zero-knowledge protocols from ring-sis with applications to lattice-based threshold cryptosystems. *IACR Cryptology ePrint Archive*, 2016:997, 2016.

[Ber01] Daniel J. Bernstein. Multidigit multiplication for mathematicians, 2001.

[BKLP15] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS*, pages 305–325, 2015.

[CDXY17] Ronald Cramer, Ivan Damgård, Chaoping Xing, and Chen Yuan. Amortized complexity of zero-knowledge proofs revisited: Achieving linear soundness slack. In *EUROCRYPT*, pages 479–500, 2017.

[DL17] Rafaël Del Pino and Vadim Lyubashevsky. Amortization with fewer equations for proving knowledge of small secrets. *IACR Cryptology ePrint Archive*, 2017:280, 2017. To appear in CRYPTO 2017.

[GLP12] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, pages 530–547, 2012.

[HJP13] W. Hart, F. Johansson, and S. Pancratz. FLINT: Fast Library for Number Theory, 2013. Version 2.4.0, http://flintlib.org.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, pages 372–389, 2008.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP (2)*, pages 144–155, 2006.

[LN86] Rudolph Lidl and Harald Niederreiter. *Introduction to Finite Fields and their Applications*. Cambridge University Press, 1986.

[LN17] Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT*, pages 293–323, 2017.

[LNSW13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC*, pages 107–124, 2013.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, 75(3):565–599, 2015.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616, 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755, 2012.

[PG13] Thomas Pöppelmann and Tim Güneysu. Towards practical lattice-based public-key encryption on reconfigurable hardware. In *SAC*, pages 68–85, 2013.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, pages 145–166, 2006.

[Ste93] Jacques Stern. A new identification scheme based on syndrome decoding. In *CRYPTO*, pages 13–21, 1993.