CS 294. Ideal Lattices, Ring-SIS and Ring-LWE

This chapter is adapted from notes by Noah Stephens-Davidowitz.

1 Hash Functions

The SIS problem yields a very simple collision-resistant hash function that is provably secure if worst-case lattice problems are hard:

$$h_{\mathbf{A}}(\mathbf{e}) = \mathbf{A}\mathbf{e} \pmod q$$

where the key $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is uniformly random and the input is $\mathbf{e} \in \{0,1\}^m$. Recall that finding an $h_{\mathbf{A}}$ collision is equivalent to solving the SIS problem, whose definition we reproduce below.

Definition 1. *For parameters n, m, q, the (average-case, homogenous) Short Integer Solutions (SIS) problem is defined as follows. The input is a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. The goal is to find a non-zero vector $\mathbf{e} \in \{-1,0,1\}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{0} \pmod q$.*

$h_{\mathbf{A}}$ has a lot going for it as a hash function. It is remarkably simple—a linear collision-resistant hash function! And, we saw that it is provably secure under the assumption that certain well-studied worst-case lattice problems are hard. If those two things are not enough, $h_{\mathbf{A}}$ is also worthy of study because of its close relationship with LWE, the topic of this course and an extremely important problem for cryptographers.

Unfortunately, $h_{\mathbf{A}}$ is quite inefficient, since just reading the public hash description $\mathbf{A}$ takes time roughly $nm \log q > n^2$ (where the inequality follows from the fact that we must have $m > n$ in order for $h_{\mathbf{A}}$ to be a compressing function). But, $h_{\mathbf{A}}$ is breakable in time $2^{O(m)}$ (even by brute-force search).

*Ideal*ly, we would hope for a hash function that can be broken in time $2^{O(m)}$ to run in time roughly linear in $m \approx n$. Our goal is therefore to show a variant of $h_{\mathbf{A}}$ whose running time is in fact roughly linear in n.

2 The cyclic shift matrix, and the ring $\mathbb{Z}[x]/(x^n - 1)$

Since just reading the key of $h_{\mathbf{A}}$ requires time greater than n^2, any attempt to speed up the computation of $h_{\mathbf{A}}$ will presumably have to first compress the key size. E.g., we could take some short uniformly random seed r (with bit length, say, $O(n)$) and set $\mathbf{A} = H(r)$ for some suitable expanding function H. If H is modeled as a random oracle, then the resulting hash function $h_{H(r)}$ retains its security. (This idea is actually quite useful in practice [BCD$^+$16] in the context of LWE.) However, if H is an arbitrary function, then we do not expect to be able to compute $h_{H(r)}(\boldsymbol{e})$ in time faster than n^2. So, though this idea immediately yields a hash function with a smaller key, we need to do more work to get a faster hash function.

3 The cyclic shift matrix, and the ring $\mathbb{Z}[x]/(x^n - 1)$

Since just reading the key of $h_{\mathbf{A}}$ requires time greater than n^2, any attempt to speed up the computation of $h_{\mathbf{A}}$ will presumably have to compress the key size. E.g., we could take some short uniformly random seed r (with bit length, say, $O(n)$) and set $\mathbf{A} = H(r)$ for some suitable expanding function H. If H is modeled as a random oracle, then the resulting hash function $h_{H(r)}$ retains its security. (This idea is actually quite useful in practice [BCD+16] in the context of LWE.) However, if H is an arbitrary function, then we do not expect to be able to compute $h_{H(r)}(\boldsymbol{e})$ in time faster than n^2. So, though this idea immediately yields a hash function with a smaller key, we need to do more work to get a faster hash function.

Even so, assuming that we are happy with a small key and quadratic runtime, I do not know how to prove the security of this approach from standard assumptions, e.g., SIS.

Open Problem. Construct a hash function with a linear-size key which is as secure as SIS or LWE (in particular, without relying on the random oracle assumption.)

In order to speed up our computation, we presumably need our matrix $\mathbf{A}$ to be a very special function of the seed. To that end, we take our short random seed to be $\ell = m/n$ uniformly random vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\ell \in [q]^n$, and we take the columns of our matrix A to be the vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\ell$ together with all "cyclic rotations" of the $\boldsymbol{a}_i$. I.e., for $\boldsymbol{a} = (a_1, \ldots, a_n)^T \in \mathbb{Z}^n$, we define

$$\mathrm{Rot}(\boldsymbol{a}) := \begin{pmatrix} a_1 & a_n & \cdots & a_3 & a_2 \\ a_2 & a_1 & \cdots & a_4 & a_3 \\ a_3 & a_2 & \cdots & a_5 & a_4 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & \cdots & a_n & a_{n-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_n \\ a_n & a_{n-1} & \cdots & a_2 & a_1 \end{pmatrix} \in \mathbb{Z}^{n \times n} ,$$

where each column is a simple cyclic permutation of the previous column.[1] Matrices of the form $\mathrm{Rot}(\boldsymbol{a})$ are sometimes referred to as "cyclic matrices" or "circulant matrices." We then take

$$A = (\mathrm{Rot}(\boldsymbol{a}_1), \mathrm{Rot}(\boldsymbol{a}_2), \ldots, \mathrm{Rot}(\boldsymbol{a}_\ell)) \in \mathbb{Z}^{n \times m} .$$

We claim that for A with this structure, we can compute $A\boldsymbol{e} \bmod q$ in time $n\ell \cdot \mathrm{polylog}(n, q) = \widetilde{O}(m)$.

[1] Notice that the definition of Rot does not depend at all on q. It is convenient to forget about q for now and to think of $\boldsymbol{a}$ as some arbitrary vector in $\mathbb{Z}^n$.

We claim that for A with this structure, we can compute $A\boldsymbol{e} \bmod q$ in time $n\ell \cdot \mathrm{polylog}(n,q) = \widetilde{O}(m)$. This is because the set of all integer cyclic matrices, $\widetilde{R} := \{\mathrm{Rot}(\boldsymbol{a}) \ : \boldsymbol{a} \in \mathbb{Z}^n\}$ is actually a very nice set with nice algebraic structure. In particular, we can write

$$\mathrm{Rot}(\boldsymbol{a}) = (\boldsymbol{a}, X\boldsymbol{a}, \ldots, X^{n-1}\boldsymbol{a}) \ ,$$

where

$$X := \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \in \{0,1\}^{n \times n}$$

is the "cyclic shift" matrix. Notice that $\widetilde{R} \subset \mathbb{Z}^{n \times n}$ is a lattice in $n \times n$ dimensions with rank n and basis $I_n, X, X^2, \ldots, X^{n-1}$. Indeed, for any $\boldsymbol{a} = (a_1, \ldots, a_n)^T \in \mathbb{Z}^n$, we can write

$$\mathrm{Rot}(\boldsymbol{a}) = a_1 I_n + a_2 X + \cdots + a_n X^{n-1} \ .$$

This identity immediately shows us that $\widetilde{R}$ is actually closed under (matrix) multiplication (note that $X^n = I_n$) and that multiplication is commutative over $\widetilde{R}$. I.e., $\widetilde{R}$ is a commutative ring!

In fact, $\widetilde{R}$ is isomorphic to the polynomial ring $R := \mathbb{Z}[x]/(x^n - 1)$. I.e., R is the ring of polynomials in the variable x of degree at most $n-1$ and integral coefficients, with addition defined in the obvious way and multiplication defined by the distributive law together with the identity

$$x \cdot x^i = \begin{cases} x^{i+1} & i < n-1 \\ 1 & i = n-1 \end{cases} .$$

(The polynomial $x^n - 1$ is the characteristic polynomial of the cyclic shift matrix X, which is why it arises in this context.) To see that these two rings are isomorphic, one only needs to check that the map $X \mapsto x$ is a bijection that preserves addition and multiplication of basis elements.

So, there's no reason to drag these $n \times n$ matrices around, and we can instead think of $\mathrm{Rot}(\boldsymbol{a}) \in \widetilde{R}$ as the corresponding polynomial $a \in R$ of degree at most $n-1$. (I.e., we change notation slightly.) We can therefore identify our matrix $A \in [q]^{n \times m}$ with a tuple of ring elements $(a_1, \ldots, a_\ell)^T \in R_{[q]}^\ell$, and similarly the input $\boldsymbol{e} \in \{0,1\}^m$ is a tuple of ring elements $(e_1, \ldots, e_\ell)^T \in R_{\{0,1\}}^\ell$, where we use the notation R_S to represent the set of polynomials in R with coefficients in S. Therefore, our hash function is now $h_{a_1,\ldots,a_\ell}(e_1, \ldots, e_\ell) = a_1 e_1 + \cdots + a_\ell e_\ell \bmod qR$.[2] For convenience, we abbreviate this by $h_a(e)$.

Now, to gain in efficiency, we simply recall that we can multiply two elements in $R_{[q]}$ in time $n \cdot \mathrm{polylog}(n, q)$ via the fast Fourier transform. Therefore, we can compute h_a in time $\ell n \cdot \mathrm{polylog}(n, q) = m \cdot \mathrm{polylog}(n, q)$, which is a tremendous speedup over the $nm \cdot \mathrm{polylog}(q)$ running time of the original h_A. Indeed, we typically think of $q = \mathrm{poly}(n)$ and $\ell = \mathrm{polylog}(n)$, so that this running time is quasilinear in n.

[2] Here, we have chosen to think of the e_i as ring elements as well. This is formally justified by the identity

$$\mathrm{Rot}(\boldsymbol{a}) \cdot \mathrm{Rot}(\boldsymbol{e}) = \mathrm{Rot}\big(\mathrm{Rot}(\boldsymbol{a}) \cdot \boldsymbol{e}\big) .$$

The reduction mod qR simply means that we reduce the coefficients of the result of our polynomial multiplication modulo q.

3.1 Towards Ring-SIS

Of course, this is not very useful if h_a is not secure. In fact, Micciancio showed that h_a is secure *as a one-way function* (under a plausible worst-case lattice assumption) [Mic07]. I.e., with certain reasonable parameters, it is difficult to invert h_a on a random input. This result is really quite remarkable, but we will not state it formally.

Unfortunately, h_a is *not* a collision-resistant hash function. To see this, it helps to define the Ring-SIS problem, which is the analogue of SIS in this setting.

Definition 2. *For a ring R, integer modulus $q \geq 2$, and integer $\ell \geq 1$, the (average-case) Ring-SIS problem is defined as follows. The input is $a_1, \ldots, a_\ell \in R_{[q]}$ sampled independently and uniformly at random. The goal is to output $e_1, \ldots, e_\ell \in R_{\{-1,0,1\}}$ not all zero such that $a_1 e_1 + \cdots + a_\ell e_\ell = 0 \bmod qR$.*

One can easily see that finding a collision in h_a is equivalent to solving Ring-SIS, just like finding a collision in h_A is equivalent to solving SIS.

Unfortunately, Ring-SIS over $\mathbb{Z}[x]/(x^n - 1)$ is not hard. The issue is that the ring $\mathbb{Z}[x]/(x^n - 1)$ has non-trivial zero divisors (i.e., it is not an integral domain). To see this, let $\widetilde{e} = 1 + x + x^2 + \cdots + x^{n-1} \in \mathbb{Z}[x]/(x^n - 1)$, and notice that $(x-1)\widetilde{e} = x^n - 1 = 0$. (In terms of Rot and $\widetilde{R}$, this corresponds to the fact that $\mathrm{Rot}(\boldsymbol{u})$ is singular, where $\boldsymbol{u} = (1, 1, \ldots, 1)^T \neq \mathbf{0}$.) This leads to an attack.

Claim 3. *For any integer modulus $q \geq 2$ and integer $n \geq 1$, let $R := \mathbb{Z}[x]/(x^n - 1)$ and let $\widetilde{e} = 1 + x + x^2 + \cdots + x^{n-1} \in R_{-1,0,1}$. Then, $a\widetilde{e} = 0 \bmod qR$ with probability $1/q$ when $a \in R_{[q]}$ is sampled uniformly at random.*

In particular, $\widetilde{e}, 0, \ldots, 0 \in R_{\{-1,0,1\}}$ is a solution to Ring-SIS over R with probability $1/q$, and the hash function h_a can be broken efficiently with probability $1/q$.

Proof. Suppose that $a \in R_{[q]}$ is divisible by $x - 1$ modulo qR. I.e., $a = (x-1)a' \bmod qR$. Then, $\widetilde{e}a = \widetilde{e}(x-1)a' = 0 \bmod qR$. The result follows by noting that $a \in R_{[q]}$ is divisible by $x-1$ modulo qR with probability $1/q$. (Notice that being divisible by $x - 1$ is equivalent to having coefficients that sum to zero mod q.) □

If our original hash function h_A is in fact $2^{\Omega(n)}$ secure, then this result makes h_a uninteresting as a collision-resistant hash function. In particular, in order for h_a to have a chance of matching this security, we would need to take $q = 2^{\Omega(n)}$, in which case h_a is actually a slower hash function than h_A.

4 The ring $\mathbb{Z}[x]/(x^n + 1)$, ideal lattices, and a secure collision-resistant hash function

Recall that our attack on h_a over $\mathbb{Z}[x]/(x^n - 1)$ relied on the fact that $x^n - 1$ has a nontrivial factor over the integers, $x^n - 1 = (x - 1)(x^{n-1} + x^{n-2} + \cdots + 1)$. So, it is natural to try replacing $x^n - 1$ with an irreducible polynomial. Indeed, one can easily show that $\mathbb{Z}[x]/(p(x))$ for some polynomial $p(x) \in \mathbb{Z}[x]$ is an integral domain if and only if p is irreducible.

We strongly prefer sparse polynomials with small coefficients (both because they are easy to work with and because this ensures that our ring has nice "geometric" properties). Since $x^n - 1$ failed, we try $x^n + 1$. This is irreducible over $\mathbb{Z}$ if and only if n is a power of two.[3] So, we take $R := \mathbb{Z}[x]/(x^n + 1)$ for n some power of two. I.e., R is the ring of polynomials over $\mathbb{Z}$ of degree at most $n - 1$ with addition defined in the obvious way and multiplication defined by

$$x \cdot x^i = \begin{cases} x^{i+1} & i < n-1 \\ -1 & i = n-1 \end{cases} .$$

From the matrix perspective of the previous section, this corresponds to taking

$$\mathrm{Rot}(\boldsymbol{a}) = (\boldsymbol{a}, X\boldsymbol{a}, \ldots, X^{n-1}\boldsymbol{a}) = \begin{pmatrix} a_1 & -a_n & \cdots & -a_3 & -a_2 \\ a_2 & a_1 & \cdots & -a_4 & -a_3 \\ a_3 & a_2 & \cdots & -a_5 & -a_4 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & \cdots & -a_n & -a_{n-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & -a_n \\ a_n & a_{n-1} & \cdots & a_2 & a_1 \end{pmatrix} \in \mathbb{Z}^{n \times n} ,$$

where

$$X := \begin{pmatrix} 0 & 0 & \cdots & 0 & -1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \in \{0,1\}^{n \times n} .$$

Notice that X differs in just one entry from our choice in the previous section. Matrices of the form $\mathrm{Rot}(\boldsymbol{a})$ as above are occasionally called "anti-cyclic."

[3] If $p > 1$ is a non-trivial odd factor of n, then $x^{n/p} + 1$ is a non-trivial factor of $x^n + 1$. If n has no odd factors, then $x^n + 1$ is the $2n$th cyclotomic polynomial—i.e., the minimal polynomial over $\mathbb{Z}$ of any primitive $2n$th root of unity.

As before, we define our hash function $h_a(e) = a_1 e_1 + \cdots + a_\ell e_\ell \mod qR$, where the a_i are chosen uniformly $a_i \in R_{[q]}$ and $e_i \in R_{\{0,1\}}$. But, we stress that the underlying ring has changed from $\mathbb{Z}[x]/(x^n - 1)$ to $R = \mathbb{Z}[x]/(x^n + 1)$, so that this is not the same hash function as before. (Formally, we should include the ring as a parameter in h, i.e. $h_{a,\mathbb{Z}[x]/(x^n+1)}$, to distinguish it, but we prefer to keep the notation uncluttered.) As before, finding a collision for this hash function is equivalent to solving Ring-SIS, now over this new ring, $\mathbb{Z}[x]/(x^n + 1)$.

Ring-SIS is in fact hard over this ring, under a reasonable *worst-case* complexity assumption. We will describe this complexity assumption (which will lead us to the topic of ideal lattices) but will not prove the worst-case to average-case reduction for Ring-SIS.

Remark The ring R is rather special; it is the ring of integers of the cyclotomic number field $\mathbb{Q}[x]/(x^n + 1)$. Number fields and their rings of integers are very well-studied and very interesting objects, and these notes stop short of presenting some of the beautiful mathematics that is lurking beneath the surface here. (The fact that R is such a rich mathematical object also seems relevant for the security of h_a. In particular, there are algorithmic results for related problems that exploit rather deep properties of R [Ber14, CGS14, CDPR16, CDW17].)

4.1 Ideal lattices

In order to present the worst-case hardness assumption that will imply the security of our hash function, we will need to introduce a special class of lattices known as *ideal lattices*. Recall that a lattice is an additive subgroup of $\mathbb{Z}^n$. I.e., a subset of $\mathbb{Z}^n$ closed under addition and subtraction. An ideal $\mathcal{I} \subseteq R$ is an additive subgroup of a ring R that is closed under multiplication by any ring element. I.e., $\mathcal{I}$ is closed under addition and subtraction, *and* for any $y \in \mathcal{I}$ and $r \in R$, we have $ry \in \mathcal{I}$.

For our choice of ring, we can view $\mathcal{I}$ as a lattice by embedding R in $\mathbb{Z}^n$ via the trivial embedding that maps x^i to the unit vector $\boldsymbol{e}_i$. So, $\mathcal{I}$ can equivalently be viewed as a lattice $\mathcal{I} \subseteq \mathbb{Z}^n$ that is invariant under the linear transformation X. I.e., $\mathcal{I} \subseteq \mathbb{Z}^n$ is a lattice such that $(y_1, \ldots, y_n)^T \in \mathcal{I}$ if and only if $(-y_n, y_1, y_2, \ldots, y_{n-1})^T \in \mathcal{I}$. Such lattices are sometimes called "anti-cyclic," and the corresponding lattices over $\mathbb{Z}[x]/(x^n - 1)$ are often called "cyclic."

In particular, this embedding allows us to consider the geometry of an ideal $\mathcal{I}$, as a subset of $\mathbb{Z}^n$. E.g., we can define the ℓ_2 norm and the inner product over $\mathcal{I}$ by taking the ℓ_2 norm and the inner product over $\mathbb{Z}^n$.[4] We then see that ideal lattices $\mathcal{I}$ are a strange class of lattices in which non-zero lattice elements $y \in \mathcal{I}$ can be divided into groups of n linearly independent elements, $y, xy, x^2y, \ldots, x^{n-1}y$, all with the same length, $\|x^i y\| = \|x^j y\|$. In particular $\lambda_1(\mathcal{I}) = \lambda_n(\mathcal{I})$. (Notice that we move freely between the representation of R as $\mathbb{Z}^n$ and the representation of R as a polynomial ring. I.e., we can think of $y_1, y_2 \in R$ as scalars, written in plain font, as opposed to boldface vectors $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{Z}^n$. We can still talk about their norms $\|y_1\|, \|y_2\|$ and inner product $\langle y_1, y_2 \rangle$.)

Remark Ideals are very important objects in the study of rings, and they have a rich history that we do not discuss here. In fact, much of the early study of lattices was motivated by the study of the geometry of ideals, going back all the way to the seminal work of Minkowski, Dirichlet, and others in the middle of the 19th century.

[4] For more general rings of integers over number fields, there is actually a different notion of geometry obtained via the "canonical embedding" of $\mathcal{I}$ into $\mathbb{C}^n$, which has very nice properties. E.g., in the canonical embedding, ring multiplication is coordinate-wise. For our very special choice of ring, $\mathbb{Z}[x]/(x^n + 1)$ for n a power of two, these two embeddings actually yield the same geometry.

4.2 SVP over ideal lattices and worst-case hardness

For our purposes, this view of ideals as lattices is useful because it allows us to extend computational lattice problems to ideals. I.e., for some fixed ring R, we can define the computational problems γ-IdealSVP, γ-IdealSIVP, γ-GapIdealSVP, etc., as the corresponding computational problems restricted to ideal lattices. In fact, the above discussion shows that γ-IdealSVP and γ-IdealSIVP are equivalent over our ring $R = \mathbb{Z}[x]/(x^n + 1)$. A slightly more sophisticated argument shows that γ-GapIdealSVP is easy over R for $\gamma > \sqrt{n}$ because the length of the shortest vector in an ideal can be approximated up to a factor of $\sqrt{n}$ by the determinant. We therefore only present a formal definition of γ-IdealSVP.

Definition 4. *For a ring R (with an associated norm $\|\cdot\|$) and approximation factor $\gamma \geq 1$, γ-IdealSVP over R is the approximate search problem defined as follows. The input is (a basis for) an ideal lattice $\mathcal{I}$ over R. The goal is to output a non-zero element $y \in \mathcal{I}$ with $\|y\| \leq \gamma\lambda_1(\mathcal{I})$.*

With this, we can present the worst-case to average-case hardness of Ring-SIS, which was discovered independently by Peikert and Rosen [PR06] and by Lyubashevsky and Micciancio [LM06].

Theorem 5 ([PR06, LM06])**.** *For any power of two n, integer $\ell \geq 1$, and integer modulus $q \geq 2n^2\ell$, γ-Ideal SVP over $R = \mathbb{Z}[x]/(x^n + 1)$ can be efficiently reduced to Ring-SIS over R, where $\gamma = \ell \cdot \mathrm{poly}(n)$.*

4.3 Regarding the hardness of IdealSVP

Of course, Theorem 5 is only interesting if γ-IdealSVP is hard over the ring $\mathbb{Z}[x]/(x^n+1)$. Until very recently, our best algorithms for this problem were essentially no better than our generic algorithms for γ-SVP over general n-dimensional lattices. However, very recently, polynomial-time quantum algorithms for γ-IdealSVP with the very large approximation factor $\gamma = 2^{\widetilde{O}(\sqrt{n})}$ were discovered in a series of works [Ber14, CGS14, CDPR16, CDW17]. (The best known algorithms for $2^{\sqrt{n}}$-SVP run in time roughly $2^{\sqrt{n}}$, even on a quantum computer. And, our best polynomial-time algorithms for γ-SVP only achieve an approximation factor of $\gamma = 2^{\widetilde{\Theta}(n)}$. So, this is a very big improvement.)

These algorithms are not known to extend to attacks on Ring-SIS for two reasons. First, the approximation factor $\gamma = 2^{\widetilde{O}(\sqrt{n})}$ is much larger than the approximation factors that are relevant to Ring-SIS. Second, Ring-SIS is not exactly an ideal lattice problem. Instead, notice that a solution to Ring-SIS consists of a *vector* of ring elements $(e_1, \ldots, e_\ell) \in R^\ell$. Indeed, Ring-SIS is technically a lattice problem over rank ℓ *modules*. It is therefore not currently known how to efficiently reduce Ring-SIS to IdealSVP.

As a result of all of this, the status of Ring-SIS is a bit unclear at the moment. The barriers mentioned in the previous paragraph seem to be quite hard to overcome, so perhaps this new line of research will not lead to an attack. As far as we know, Ring-SIS is just as hard as SIS, and indeed, as far as we know, it could yield a collision-resistant hash function that is computable in $\widetilde{O}(n)$ time and only breakable in time $2^{\Omega(n)}$.

We will not present the worst-case to average-case reduction for Ring-SIS. It is a slight variant of the reduction that we have already seen for SIS, and is included in the posted lecture notes (`http://people.csail.mit.edu/vinodv/CS294/lecturenotes.pdf`).

5 Ring-LWE basics and some properties of $\mathbb{Z}[x]/(x^n+1)$

5.1 From Ring-SIS to Ring-LWE

Now, we do unto LWE what we just did to SIS. In particular, the problem (search) LWE asks us to find $\boldsymbol{s} \in \mathbb{Z}_q^n$ given $(A, \boldsymbol{s}^T A + \boldsymbol{e}^T \bmod q)$, where $A \in \mathbb{Z}_q^{n \times m}$ and $\boldsymbol{s} \in \mathbb{Z}_q^n$ are uniformly random and $\boldsymbol{e} \in \mathbb{Z}^m$ is chosen from some error distribution on short vectors. We will define Ring-LWE in a similarly natural way. We will see that the hardness of Ring-LWE implies more efficient public-key cryptography, and that this hardness can be based on the worst-case hardness of the worst-case ideal lattice problem γ-IdealBDD (which we will define later). Because we will rely very heavily on special properties of our specific ring $R = \mathbb{Z}[x]/(x^n+1)$ for n a power of two, we only define Ring-LWE over this specific ring. Everything presented here can be generalized, but doing so requires quite a bit more work [LPR10].[5]

Definition 6. *For integers $\ell, q \geq 2$, power of two n, and an error distribution χ over short elements in R, the (average-case, search) Ring-LWE problem is defined as follows. The input is $a_1, \ldots, a_\ell \in R_q$ sampled independently and uniformly at random together with $b_1, \ldots, b_n \in R_q$, where $b_i := a_i \cdot s + e_i \bmod qR$ for $s \in R_q$, and $e_i \sim \chi$. The goal is to output s.*

Notice that we take s to be worst-case, rather than uniformly random. This is without loss of generality, since we can trivially randomize s if necessary. Just like before, we will also need the decisional version of the problem, which asks us to distinguish the (a_i, b_i) from uniformly random and independent elements of R_q.

[5] The "right" notion of Ring-LWE for more general rings has a more sophisticated definition based on the canonical embedding of a number field. In particular, the naive coefficient embedding in which the norm of a ring element is just the norm of its coefficient vector does not behave nicely for general rings. In the special case when $R = \mathbb{Z}[x]/(x^n+1)$ for n a power of two, the canonical embedding and coefficient embedding are identical (up to scaling and rotation), so we can largely ignore these issues.

5.2 Basic properties

Ring-LWE inherits many of LWE's nice properties. In particular, Ring-LWE is equivalent to the planted variant of Ring-SIS, and the hardness of Ring-LWE (both search and decision) remains unchanged if we sample the secret s from the error distribution χ (at the expense of one sample). One can prove both of these facts in more-or-less the same way that we proved the corresponding facts for plain LWE, at least for appropriate choices of q.

For example, given ℓ Ring-LWE samples $(a_1, b_1), \ldots, (a_\ell, b_\ell)$ with $b_i := a_i s + e_i$, we can try to convert them into $\ell - 1$ Ring-LWE samples with the secret sampled from the error distribution as follows. We assume that one of the a_i is invertible in R_q (i.e., there exists an element $a_i^{-1} \in R_q$ such that $a_i a_i^{-1} = 1$, which happens with non-negligible probability, as shown in [LPR13, Claim 2.25]). Then, $a_j a_i^{-1} b_i = a_j s + a_j a_i^{-1} e_i$, and $a_j a_i^{-1} b_i - b_j = a_j a_i^{-1} e_i + e_j$. We can therefore create the new samples $(a_j a_i^{-1}, a_j a_i^{-1} b_i - b_j)$ for all $j \neq i$, which are $\ell - 1$ valid Ring-LWE samples with secret e_i and error e_j, as needed.

5.3 Encryption

Recall that we saw both a secret-key encryption scheme and a public-key encryption scheme from plain LWE. Both of these schemes have natural analogues in the Ring-LWE world. Just like our Ring-SIS-based hash function, these schemes are remarkably efficient.

The secret-key encryption scheme is as follows. Both this scheme and the public-key scheme naturally use $R_{\{0,1\}}$ as their message space, i.e., polynomials with $\{0,1\}$ coefficients. (Compare this to the one-bit message space that we obtained for LWE.)

- **Key generation:** The secret key is simply a uniformly random element $s \in R_q$.
- **Encryption:** To encrypt $m \in R_{\{0,1\}}$, compute (a, b) for $b := a \cdot s + e + \lfloor q/2 \rceil \cdot m \bmod qR$, where $a \in R_q$ is chosen uniformly at random and $e \sim \chi$.
- **Decryption:** To decrypt (a, b), compute $b - a \cdot s \bmod qR = \lfloor q/2 \rceil \cdot m + e \bmod qR$. Round each coefficient to either $q/2$ or zero, whichever is closest (where we assume that our representation modulo qR uses coefficients in $[q]$), and interpret 0 as 0 and $q/2$ as 1.

Clearly, this scheme is correct if and only if the coefficients of e are smaller than roughly $q/4$. Furthermore, the CPA-security of the scheme is immediate from Ring-LWE. And, this scheme is quite efficient:

- secret keys have size $n \log q$;
- encrypting n-bit messages using roughly $n \log q$-bit ciphertexts; and
- encryption and decryption run in time $n \cdot \mathrm{polylog}(n, q)$.

As far as we know, this scheme is $2^{\Omega(n)}$ secure for appropriate parameters, so that we may take n only linear in the security parameter.

As a parenthetical remark, we can achieve such short ciphertexts from LWE as well (as shown by Peikert, Vaikuntanathan and Waters.) with the following properties:

- secret keys have size $n^2 \log q$;
- encrypting n-bit messages using roughly $n \log q$-bit ciphertexts; and
- encryption and decryption run in time $n^2 \cdot \mathrm{polylog}(n, q)$.

The public-key encryption scheme is as follows.

- **Key generation:** The secret key is a short secret $s \sim \chi$. The public key is $(\widehat{a}, y)$ for $\widehat{a} \in R_q$ uniformly random and $y := \widehat{a} \cdot s + e \bmod qR$, where $e \sim \chi$.
- **Encryption:** To encrypt $m \in R_{\{0,1\}}$, compute (a, b), where $a := \widehat{a}r + x \bmod q$ and $b := yr + x' + \lfloor q/2 \rceil\, m \bmod q$ for $r, x, x' \sim \chi$.
- **Decryption:** To decrypt (a, b), compute $b - a \cdot s \bmod qR = \lfloor q/2 \rceil\, m + er + x' - xs \bmod q$ and again do our rounding procedure to find m.

Clearly, this scheme is correct if and only if $er + x' - xs$ is less than $q/4$. (So, we can take our error to have size roughly $\sqrt{q}/2$.) Security follows from a proof similar to the one for plain LWE in our first lecture. I.e., we use the hardness of decisional Ring-LWE with short secrets once to show that the public key can be replaced by uniformly random ring elements and then again to show that the element b in the ciphertext can also be replaced by a uniformly random ring element.

Again, we note the remarkable efficiency of this scheme. As far as we know, it is $2^{\Omega(n)}$ secure and all operations are computable in time $n \cdot \mathrm{polylog}(n, q)$. Taking $q = \mathrm{poly}(n)$ gives a public-key encryption scheme with key generation, encryption, and decryption all computable in time quasilinear in the security parameter. And, Lyubashevsky, Peikert, and Regev proved that breaking this scheme is at least as hard as a certain worst-case ideal lattice problem [LPR10]—even an ideal lattice problem that is plausibly $2^{\Omega(n)}$ hard.

5.4 Reduction modulo ideals and Chinese Remainder Theorem

Recall that for an element $r \in R$ in some ring R (e.g., $R = \mathbb{Z}$), we define equivalence of $s_1, s_2 \in R$ modulo r by $s_1 = s_2 \bmod r$ if and only if there exists an $r' \in R$ with $s_1 = s_2 + r'r$. Equivalently, $s_1 = s_2 \bmod r$ if and only if there exists an *ideal* element $y \in rR := \{r' \cdot r \ : \ r' \in R\}$ in the ideal rR generated by r such that $s_1 = s_2 + y$. This is an equivalence relation because the ideal is closed under addition, which also implies that it respects addition. It respects multiplication because the ideal is closed under multiplication by any ring element. I.e., if $s_1 = s_2 + y$ for $y \in rR$, then $xs_1 = xs_2 + xy$, which implies that $xs_1 = xs_2 \bmod r$, since $xy \in rR$ also.

This immediately shows that we can also reduce modulo an arbitrary ideal $\mathcal{I}$, not just an ideal generated by a single element. I.e., we define $s_1 = s_2 \bmod \mathcal{I}$ if and only if there exists $y \in \mathcal{I}$ such that $s_1 = s_2 + y$. (This is a big part of the reason why ideals are such important objects in the study of rings, as opposed to, say, subrings.) Just like before, addition and multiplication are well defined modulo $\mathcal{I}$, and we write $R/\mathcal{I}$ for the ring of equivalence classes modulo $\mathcal{I}$.[6]

[6] In fact, we have already been sneakily using this notation, writing $\bmod qR$, rather than $\bmod q$.

We will need something slightly more general. For an ideal $\mathcal{J} \subseteq \mathcal{I}$ (e.g, $\mathcal{J} = q\mathcal{I}$), we can again define the quotient $\mathcal{I}/\mathcal{J}$. This quotient is also a ring, and we can define multiplication by x in $\mathcal{I}/\mathcal{J}$ in the obvious way.

We can now present the Chinese Remainder Theorem over R. (A far more general theorem holds here over a very large class of rings.) We say that two ideals $\mathcal{I}$ and $\mathcal{J}$ are *coprime* if there exists $y \in \mathcal{I}, z \in \mathcal{J}$ such that $y + z = 1$.

Theorem 7 (Chinese Remainder Theorem for R)**.** *For any pairwise coprime ideals $\mathcal{I}_1, \ldots, \mathcal{I}_k \subseteq R$ over R, let $\mathcal{I} := \bigcap \mathcal{I}_j$. Then, $R/\mathcal{I}$ is isomorphic (as a ring) to the direct product*

$$\frac{R}{\mathcal{I}_1} \times \frac{R}{\mathcal{I}_2} \times \cdots \times \frac{R}{\mathcal{I}_k} \ .$$

Indeed, an isomorphism is given by the natural map

$$r \mapsto (r \bmod \mathcal{I}_1, r \bmod \mathcal{I}_2, \ldots, r \bmod \mathcal{I}_k) \ ,$$

and it can be efficiently inverted.

Furthermore, in the special case when $\mathcal{I} = qR$ for a prime q, we may take the $\mathcal{I}_j$ to be the ideal generated by q and the jth irreducible factor of $x^n + 1$ modulo q. Then, the quotients $R/\mathcal{I}_j$ are actually fields of characteristic q.

In particular, turning back to the question of invertibility of a_i from Section 5.2, we see that at least for prime q, $a \in R_q$ is invertible unless $a = 0 \bmod \mathcal{I}_j$ for some j (since the quotients are fields and therefore do not have zero divisors). Since the quotient has size at least q, this happens with probability at most $1/q$. Because of the product structure guaranteed by the Chinese Remainder Theorem, we then see that a is invertible with probability at least $(1 - 1/q)^n$.

6 Search to decision

We will prove the following search-to-decision reduction for Ring-LWE, which was originally proven by Lyubashevsky, Peikert, and Regev [LPR10]. We say that a polynomial *splits mod q* if it is the product of distinct linear factors modulo q. We say that an error distribution χ over R is *spherically symmetric* if the probability of sampling a ring element from χ depends only on its norm.

Theorem 8. *For a prime $q \geq 2$, integer $\ell \geq 2$, power of two n such that $x^n + 1$ splits mod q, and spherically symmetric error distribution χ, there is a reduction from search Ring-LWE to decision Ring-LWE that runs in time $q \cdot \mathrm{poly}(n, \ell)$*

Many new issues arise in the ring setting. Therefore, the proof is quite a bit more difficult than the relatively easy proof for plain LWE. Indeed, lurking behind this reduction is quite a bit of Galois theory. (We refer the reader to [LPR10] for a much more thorough discussion.) Furthermore, the result is not entirely satisfying for at least two reasons.

First, the running time proportional to q is unfortunate, since our worst-case to average-case reduction will only work for exponentially large moduli q. Recall that we had this issue in the plain LWE case as well, but there we saw that modulus-switching techniques can be used to reduce exponential q to polynomial q (with a large loss in parameters) [BLP$^+$13]. However, nothing similar is known in the Ring-LWE setting. Indeed, the only hardness results known for Ring-LWE with small q use a quantum reduction [LPR10, PRS17], which we will not present here.

Second, the fact that our polynomial $x^n + 1$ splits mod q is a bit worrisome, since we saw an attack on Ring-SIS when the polynomial modulus has a high-degree factor with small coefficients over the integers. *That* attack does extend to Ring-LWE, but as far as we know, there is no attack that *exploits a modulus q over which $x^n + 1$ factors.* Indeed, the worst-case to average-case reduction in [LPR10] shows that, if Ideal-SVP with appropriate parameters is hard for a quantum computer, then Ring-LWE is also hard for any sufficiently large modulus q, regardless of whether $x^n + 1$ splits modulo q.

6.1 Where we're going

Recall that our search-to-decision reduction for plain LWE worked by guessing the coordinates of the secret vector $\boldsymbol{s} \in \mathbb{Z}_q^n$ one by one. One might therefore hope to find a similar reduction for Ring-LWE that works by guessing the *coefficients* of the secret ring element $s \in R_q$ one by one. However, it is not at all clear how to do this. In the plain LWE case, we crucially used the fact that knowing a coordinate of $\boldsymbol{s}$ allows us to compute $\langle \boldsymbol{a}, \boldsymbol{s} \rangle \bmod q$ for some $\boldsymbol{a} \in \mathbb{Z}_q^n$. (Namely, the standard basis vector corresponding to the relevant coordinate.) However, knowing just one coefficient of s (or even $n-1$ coefficients of s) does not allow us to compute $a \cdot s \bmod qR$ for any non-zero $a \in R_q$.

We will need to develop a few tools in the next few subsections to correct this. The high-level structure is as follows. First, we show how to use a different coordinate system, based on the Chinese Remainder Theorem, to make multiplication coordinate-wise. This is nice because it allows us to guess a coordinate in a meaningful way. However, when we guess wrong, we will not end up with uniformly random samples. Instead, we will get Ring-LWE samples that are uniform in just one coordinate, and it is not immediately clear how to use a decision oracle to distinguish these two cases. In order to get around this, we will show the existence of very special functions that essentially allow us to "swap" coordinates. Finally, we will use a hybrid argument together with these tools to prove that hardness of search Ring-LWE implies hardness of decision Ring-LWE.

6.2 The CRT embedding (which is very different from the coefficient embedding!)

Our first task is to find a coordinate system in which multiplication is coordinate-wise. E.g., in these coordinates, the product of $(s_1, s_2, \ldots, s_n)$ with $(1, 0, 0, \ldots, 0)$ should simply be $(s_1, 0, 0, \ldots, 0)$. Indeed, since $x^n + 1$ splits modulo q, the Chinese Remainder Theorem tells us that R_q is isomorphic as a ring to the ring $\mathbb{Z}_q^n$ under coordinate-wise multiplication. So, we can in fact write ring elements $a, s \in R_q$ in a coordinate system such that $a \cdot s = (a_1 s_1, \cdots a_n s_n)$. We call this the *CRT embedding*, in contrast to the *coefficient embedding* in which we view the ring elements as polynomials. We recall that the Chinese Remainder Theorem guarantees that we can move efficiently between these two embeddings. (Indeed, this is accomplished via an invertible linear map over the field $\mathbb{Z}_q$.)

It might seem a bit silly to have gone through all of the trouble of defining Ring-LWE over polynomial rings just to end up working with $\mathbb{Z}_q^n$ under coordinate-wise multiplication! But, we stress that the error distribution looks quite different in the CRT embedding. (If we used as an error distribution that is short in the CRT embedding, the resulting Ring-LWE problem would be easy.)

To see this, let's consider the smallest non-trivial example. The polynomial x^2+1 splits modulo 13 as $x^2+1=(x+5)(x-5) \bmod 13$, so an element $ax+b \in \mathbb{Z}_{13}/(x^2+1)$ has CRT representation $(5a+b, b-5a) \in \mathbb{Z}_{13}^2$. (Check this!) Therefore, if our initial error distribution is, say, uniform over polynomials with $a, b \in \{-1,0,1\}$, then in the CRT embedding, our error distribution is uniform over the rather strange set $\{(0,0), \pm(5,-5), \pm(1,1), \pm(6,-4), \pm(6,-4)\}$, which in particular contains quite long elements, relative to $q=13$. I.e., the mapping from the coefficient embedding to the CRT embedding is a linear transformation with large distortion (to the extent that one can define "distortion" over a finite vector space).

So, while we *can* equivalently define Ring-LWE in terms of $\mathbb{Z}_q^n$ (when x^n+1 splits modulo q), we would end up with a much less natural error distributions. In particular, the error distributions obtained from our worst-case to average-case reduction would be rather strange and depend on q in complicated ways.

6.3 When and how $x^n + 1$ splits modulo q

We now consider when $x^n + 1$ splits modulo q and show that the factors take a nice form. Notice that x^n+1 is the minimal polynomial over $\mathbb{Z}$ of the (complex) primitive $2n$th roots of unity $e^{k\pi i/(2n)}$ for odd k. I.e., x^n+1 splits over $\mathbb{C}$ precisely because $\mathbb{C}$ contains such elements. In analogy with this, suppose that $z \in \mathbb{Z}_q^*$ is a primitive $2n$th root of unity modulo q. That is, suppose $z^{2n} = 1 \bmod q$ but $z^k \neq 1 \bmod q$ for all $0 < k < 2n$. Then, clearly $z^n \neq 1 \bmod q$ is a square root of 1 in $\mathbb{Z}_q$. Since $\mathbb{Z}_q$ is a field, the only square roots of 1 are ± 1, so we must have $z^n = -1 \bmod q$. I.e., $z^n + 1 = 0 \bmod q$.

Furthermore, for any odd k, z^{kn} is also a primitive $2n$th root of unity. So, $z, z^3, z^5, \ldots, z^{2n-1} \in \mathbb{Z}_q^n$ are all roots of $x^n + 1$ modulo q. Indeed, they are distinct because $z^k \neq 1$ for $0 < k < 2n$. Finally, since $\mathbb{Z}_q$ is a field, there is only one non-zero polynomial over $\mathbb{Z}_q$ of degree n with these roots, and we must have $x^n + 1 = (x - z)(x - z^3)(x - z^5) \cdots (x - z^{2n-1}) \bmod q$. I.e., $x^n + 1$ splits modulo q.

So, $x^n + 1$ splits modulo a prime q if (and only if) there is an element of order $2n$ in $\mathbb{Z}_q^*$ (i.e., a primitive $2n$th root of unity modulo q). To find such a prime, we recall that $\mathbb{Z}_q^*$ is cyclic of order $q - 1$, so that it has an element of order $2n$ if and only if $2n$ divides $q - 1$. Therefore, $x^n + 1$ splits modulo a prime q if (and only if) $q = 1 \bmod 2n$. The Prime Number Theorem in arithmetic progressions guarantees that such primes exist and can be found efficiently. And, when this is the case, the factors of $x^n + 1$ modulo q can be written as $x - z^k$ for all odd $1 \leq k \leq 2n - 1$.

6.4 Some very special automorphisms τ_k

The above discussion shows a very natural way to think of the coordinates CRT embedding. Each coordinate in the CRT embedding of a polynomial $p(x)$ is simply $p(x) \bmod \mathcal{I}_i = p(z^{2i-1}) \bmod \mathcal{I}_i$ for some $i \in [n]$, where z is some fixed primitive $2n$th root of unity modulo q and $\mathcal{I}_i$ is the ideal generated by q and $x - z^{2i-1}$. It is therefore natural to order the coordinates in the CRT embedding so that the ith coordinate is $p(z^{2i-1})$. We then observe a nice symmetry of the CRT embedding. Let $k := (2i-1)^{-1}(2j-1) \bmod 2n$ (where we have used the fact that all odd numbers have an inverse modulo $2n$). Then, we see that the ith CRT coordinate of $p(x) \in R_q$ is the jth CRT coordinate of $p(x^k)$.

So, we define $\tau_k : R_q \to R_q$ for odd k such that $\tau_k(p(x)) := p(x^k)$. We see that τ_k can be viewed as a certain permutation of the coordinates in the CRT embedding. It is therefore a ring automorphism (i.e., it is a bijection respecting addition and multiplication). In fact, it also preserves norms in the coefficient embedding! I.e., $\|\tau_k(p(x))\| = \|p(x^k)\| = \|p(x)\|$, which can be seen by observing that τ_k simply permutes the coordinates of $p(x)$ (and flips some of their signs). Such maps are very rare,[7] and very useful. The next lemma extracts the specific property that we will need from them.

Lemma 9. *The maps $\tau_k : R_q \to R_q$ as described above are efficiently computable ring automorphisms preserving the norm (in the coefficient embedding). Furthermore, τ_k acts on the CRT embedding by permuting the coordinates, and for each $i, j \in [n]$, there is an efficiently computable k such that τ_k maps the ith CRT coordinate to the jth CRT coordinate.*

[7] As we've described these maps here, they only exist for our specific choice of R_q! They can, however, be generalized to more rings if we work in the canonical embedding rather than the coefficient embedding [LPR10].

6.5 The reduction

We can now finally present our reduction. As we discussed above, we can guess the coordinate s_1 and replace the Ring-LWE sample (a_i, b_i) by

$$(a_i + \alpha_i v_1 \bmod qR, b + \alpha_i \sigma_1 v_1 \bmod qR)$$

where $v_1 = (1, 0, 0, \ldots, 0)^T$ in the CRT embedding, $\sigma_1 \in \mathbb{Z}_q$ is our guess for the first coordinate s_1 of s in the CRT embedding, and $\alpha_i \in \mathbb{Z}_q$ is uniformly random. Clearly, when our guess σ_1 is correct, the result is still a valid Ring-LWE sample with the same secret s, and the same error. However, when σ_1 is not correct, the result is not uniformly random. Instead, the first coordinate in the CRT embedding is uniformly random, but the remaining coordinates are completely unchanged.

To fix this, we use a hybrid argument together with the special maps τ_k. In particular, we let $\mathsf{Ring\text{-}LWE}_j$ be the variant of decision Ring-LWE that asks us to distinguish Ring-LWE samples in which the first $j - 1$ coordinates in the CRT embedding are replaced by uniformly random noise from Ring-LWE samples in which the first j CRT coordinates are replaced by uniformly random noise. To show the hardness of decision Ring-LWE, it suffices to show the hardness of $\mathsf{Ring\text{-}LWE}_j$ for each j.

Notice that the above argument lets us use an oracle for $\textsf{Ring-LWE}_1$ to learn the first coordinate s_1 in the CRT embedding of the secret s of a Ring-LWE instance. More generally, we can use an oracle for $\textsf{Ring-LWE}_j$ to find the jth coordinate s_j. So, to finish our proof, we need to show how the ability to find the jth coordinate s_j in the CRT embedding allows us to find all coordinates s_i. This is where we use the maps τ_k. In particular, Lemma 9 lets us find a k such that τ_k maps the ith coordinate to the jth coordinate in the CRT embedding. Since τ_k is a ring automorphism, it converts Ring-LWE samples with secret s to Ring-LWE samples with secret $\tau_k(s)$. Furthermore, since τ_k preserves the norm and the error distribution χ is spherically symmetric, τ_k preserves the error distribution.

So, our full reduction from search Ring-LWE to $\mathsf{Ring\text{-}LWE}_j$ behaves as follows. (We will denote ring elements by bold-face.) For each $i = 1, \ldots, n$, we use our $\mathsf{Ring\text{-}LWE}_j$ oracle to find the ith coordinate s_i of $\boldsymbol{s}$ in the CRT embedding by first computing $k = (2i-1)^{-1}(2j-1) \bmod 2n$ such that τ_k maps the ith CRT coordinate to the jth CRT coordinate, as in Lemma 9.

Let $\boldsymbol{v}_j \in R_q$ be the element whose coordinates in the CRT embedding are $(0, 0, \ldots, 1, 0, \ldots, 0)$, where the 1 is in the jth position. For each $\sigma \in \mathbb{Z}_q$, we replace our Ring-LWE samples $(\boldsymbol{a}_\ell, \boldsymbol{b}_\ell)$ by

$$(\tau_k(\boldsymbol{a}_\ell) + \alpha_\ell \boldsymbol{v}_j, \tau_k(\boldsymbol{b}_\ell) + \sigma\alpha_\ell \boldsymbol{v}_j + \boldsymbol{u}_\ell)$$

where $\alpha_\ell \in \mathbb{Z}_q$ is uniformly random, and $\boldsymbol{u}_\ell \in R_q$ has its first $j-1$ coordinates uniformly random in the CRT embedding and last $n-j+1$ coordinates equal to zero. If $\sigma = s_i$, then the resulting distribution

$$(\tau_k(\boldsymbol{a}_\ell) + \alpha_\ell \boldsymbol{v}_j, \tau_k(\boldsymbol{a}_\ell)\tau_k(\boldsymbol{s}_\ell) + \alpha_\ell \sigma \boldsymbol{v}_j + \tau_k(\boldsymbol{e}_\ell) + \boldsymbol{u}_\ell)$$

will be exactly the YES case of $\mathsf{Ring\text{-}LWE}_j$ with secret $\tau_k(\boldsymbol{s})$—i.e., the first $j-1$ coordinates will be uniformly random and the last $n-j+1$ coordinates will correspond to valid Ring-LWE samples. Otherwise, the distribution will be exactly the NO case—i.e., the jth coordinate will also be uniformly random.

7 NTRU

Finally, we mention a different elegant way to build public-key encryption using polynomial rings, such as R_q, the NTRU encryption scheme, due to Hoffstein, Pipher, and Silverman [HPS98]. Historically, NTRU predates LWE by nearly a decade and Ring-LWE by about 15 years. As far as we know, it is more-or-less as secure as Ring-LWE-based schemes for most reasonable parameter settings. However, unlike Ring-LWE-based schemes, NTRU comes with no worst-case hardness guarantee. We present it here because (1) it is pretty; (2)one of the relatively few concrete assumptions known to imply public-key cryptography; and (3) people who work in lattice-based cryptography should know what NTRU is.

As before, we work over $R := \mathbb{Z}[x]/(x^n + 1)$ for power-of-two n with $R_q := R/(qR)$ for some modulus $q = \mathrm{poly}(n)$. A "typical" element in R is invertible modulo $3R$ (i.e., the polynomial $x^n + 1$ does not have low-degree factors modulo 3),[8] and we may, e.g., take q to be prime to guarantee the same modulo qR. (NTRU can be defined over any polynomial ring, and it is often actually defined over $\mathbb{Z}[x]/(x^n - 1)$. This causes some annoying issues related to those that we observed in the context of Ring-SIS. They can be overcome, but we ignore this issue by using our preferred ring.)

- **Key generation:** Sample two short polynomials $g, f \in R$. E.g., sample them uniformly at random from $R_{\{0,1\}}$. If f is not invertible modulo both qR and $3R$, we resample it. Otherwise, we denote these respective inverses by f_q^{-1} and f_3^{-1}. The public key is $h := gf_q^{-1} \bmod qR$, and the private key is f, g.

- **Encryption:** Let $m \in R_{\{-1,0,1\}}$ be some ternary message. The encryption algorithm computes the ciphertext $c := hr + 3e + m \bmod qR$, where r and e are some random short polynomials.

- **Decryption:** Given a ciphertext c, we compute $fc = 3(fe + rg) + fm \bmod qR$. As long as q is sufficiently large, this element $3(fe + rg) + fm$ should have small coefficients relative to q. I.e., by choosing our representative of $3(fe + rg) + fm \bmod qR$ to have coefficients in the interval $(-q/2, q/2]$, we can actually recover $3(fe + rg) + fm \in R$, not just its coset in R_q. This allows us to reduce the result modulo $3R$ to recover fm. Finally, we multiply by f_3^{-1} to find m, which is uniquely determined by its coset modulo $3R$.

[8] It's factorization into irreducible polynomials is $x^n + 1 = (x^{n/2} + x^{n/4} - 1)(x^{n/2} - x^{n/4} - 1)$ modulo 3. The fact that these polynomials are irreducible is equivalent to saying that a finite field of characteristic 3 contains a primitive $2n$th root of unity if and only if it has size 3^m for m divisible by $n/2$, i.e., that $2n$ divides $3^m - 1$ if and only if $n/2$ divides m (since the multiplicative group of a finite field is cyclic). I was frustrated by my inability to find a nice enough proof of this, so I asked on Math StackExchange and got some very nice answers [Nic]—three very nice proof as of the last time I checked, as well as my own rather clunky proof.

The security of NTRU is typically proven under the assumption that the public key h is indistinguishable from random. However, there is no known reduction from a more standard computational problem to the problem of distinguishing h from random. For most choices of parameters, however, our best attack on NTRU is a lattice attack that searches for a short vector in the so-called NTRU lattice, spanned by the basis

$$\begin{pmatrix} I_n & 0 \\ \mathrm{Rot}(\boldsymbol{h}) & qI_n \ , \end{pmatrix} \in \mathbb{Z}^{2n \times 2n}$$

where

$$\mathrm{Rot}\begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{pmatrix} := \begin{pmatrix} h_1 & -h_n & -h_{n-1} & \cdots & -h_2 \\ h_2 & h_1 & -h_n & \cdots & -h_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & h_{n-3} & \cdots & -h_n \\ h_n & h_{n-1} & h_{n-2} & \cdots & h_1 \end{pmatrix} ,$$

as in the previous lecture, and $\boldsymbol{h}$ is the coefficient vector of the public key h. Notice that the NTRU lattice contains the short vector $(\boldsymbol{f}, \boldsymbol{g}) \in \mathbb{Z}^{2n}$ corresponding to the secret key. Indeed, any short enough vector in this lattice can be used to break the NTRU encryption scheme.

References

[BCD$^+$16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! practical, quantum-secure key exchange from lwe. In *CCS*, 2016.

[Ber14] Daniel J. Bernstein. A subfield-logarithm attack against ideal lattices, 2014.

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, 2013.

[CDPR16] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In *EUROCRYPT*, 2016.

[CDW17] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short stickelberger class relations and application to ideal-svp. In *Eurocrypt*, 2017.

[CGS14] Peter Campbell, Michael Groves, and Dan Shepherd. Soliloquy: a cautionary tale. ETSI 2nd Quantum-Safe Crypto Workshop, 2014.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. In *ANTS*, 1998.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP*, 2006.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Eurocrypt*, 2010.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. In *Eurocrypt*, 2013.

[Mic07] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4), 2007.

[Nic] Nicer proof that \$2{̂n+2}\$ divides \$3m̂-1\$ if and only if \$2{̂n}\$ divides \$m\$.

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

[PRS17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *STOC*, 2017.