INSTRUCTIONS—Read Carefully

- Time: 3 hours
- There are 10 problems.
- Write your name *legibly* at the top of this page.
- No calculators or other electronic devices are allowed. (You won’t need them.)
- When a box is provided, please put your answer in the box.
- Show all your work. Partial credit will be given for substantial progress towards the solution. No credit will be given for answers with no explanation.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Value</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>140</strong></td>
<td></td>
</tr>
</tbody>
</table>
Problem 1. (10 points) Alice and Bob communicate using the
Elgamal Elliptic Curve Public Key Cryptosystem,
with Bob sending messages to Alice.
(a) Describe what they do, including public parameters, key creation,
encryption, and decryption. (You do not have to verify that decryption
works.)
(b) Their adversary Eve listens in on their communications. Describe
as many ways as you can that Eve might try to break their system and
read Bob’s messages. (The methods that Eve tries might take too long
to be practical.)

Solution. (a) Public Parameters: Alice and Bob agree on a (large)
prime $p$, and elliptic curve $y^2 = x^3 + Ax + B$, and a point $P \in E(\mathbb{F}_p)$
(of large order).
Key Creation: Alice picks a random multiplier $a$ as her private
key. Her public key is the point $Q = aP$.
Encryption: Bob plaintext is a point $M \in E(\mathbb{F}_p)$. He also chooses
a random number $k$. Bob’s ciphertext is the pair of points $(C_1, C_2)$
with $C_1 = kP$ and $C_2 = M + kQ$.
Decryption: Alice computes the quantity $C_2 - aC_1$, which equals
Bob’s plaintext $M$.
(b) I gave full credit if people listed at least some of these.
(1) (Brute Force) Eve can compute $2P, 3P, 4P, \ldots$ until she finds
the multiple of $P$ that equals $Q$.
(2) (Giant Step–Baby Step) Let be the order of $P$ and $n = \lceil \sqrt{N} \rceil$.
Eve computes $P' =nP$, and then computes lists $P, 2P, \ldots, nP$ and
$P' - Q, 2P' - Q, \ldots, nP' - Q$. There will be a match between the two
lists, yielding an expression for $Q$ as a multiple of $P$.
(3) (Pollard rho) Use Pollard’s rho method, which also gives a way
of expressing $Q$ as a multiple of $P$ in roughly $\sqrt{N}$ steps, but requiring
very little storage.
(4) (Pairings) Depending on the parameters that Alice and Bob
are using, it may be possible to use pairings to transfer the problem to
a easier DLP in the finite field $\mathbb{F}_p^*$.
(5) (Quantum Computer) If she’s very ambitious, Eve could build
a quantum computer that is large enough to break the ECDLP. (Congrats
to the student in Math 1580 who thought of that one; I hadn’t!)
(6) Note that the problem says only that Eve listens to Alice and
Bob’s conversation. She is not able to interfere with their communica-
tions, so she can’t mount a woman-in-the-middle attack, nor can she
send Alice a message to decipher (chosen ciphertext attack).
Problem 2. (15 points) For each of the following, circle the answer that correctly completes the statement. You do not have to provide any justification for your answer. (2 points/part, except (h) is 1 point.)
(a) The fastest known way to compute \(\gcd(a, b)\) takes time that is \[\text{polynomial} \quad \text{subexponential} \quad \text{exponential}.\]
(b) The fastest known way to compute discrete logarithms in \(\mathbb{F}_p^*\) takes time that is \[\text{polynomial} \quad \text{subexponential} \quad \text{exponential}.\]
(c) The fastest known way to compute a multiple of a point on an elliptic curve \(E(\mathbb{F}_p)\) takes time that is \[\text{polynomial} \quad \text{subexponential} \quad \text{exponential}.\]
(d) The fastest known way to compute discrete logarithms on an elliptic curve \(E(\mathbb{F}_p)\) takes time that is \[\text{polynomial} \quad \text{subexponential} \quad \text{exponential}.\]
(e) The fastest known way to factor integers takes time that is \[\text{polynomial} \quad \text{subexponential} \quad \text{exponential}.\]
(f) The fastest known way to solve the closest vector problem in a lattice takes time that is \[\text{polynomial} \quad \text{subexponential} \quad \text{exponential}.\]
(g) The fastest known way to compute the inverse of a number modulo \(m\) takes time that is \[\text{polynomial} \quad \text{subexponential} \quad \text{exponential}.\]
(h) If someone were to build a (sufficiently large) working quantum computer, which of the above answers would definitely change?

Solution.
(a) The fastest known way to compute \(\gcd(a, b)\) takes \text{polynomial time}.
(b) The fastest known way to compute discrete logarithms in \(\mathbb{F}_p^*\) takes \text{subexponential time}.
(c) The fastest known way to compute a multiple of a point on an elliptic curve \(E(\mathbb{F}_p)\) takes \text{polynomial time}.
(d) The fastest known way to compute discrete logarithms on elliptic curves takes \text{exponential time}.
(e) The fastest known way to factor integers takes \text{subexponential time}.
(f) The fastest known way to solve the closest vector problem in a lattice takes exponential time.

(g) The fastest known way to compute the inverse of a number modulo \( m \) takes polynomial time.

(h) If someone were to build a (sufficiently large) working quantum computer, then (b), (d), and (e) would change from subexponential time to polynomial time.

Problem 3. (10 points) For this problem we work over the field \( \mathbb{F}_5 \). Consider the elliptic curve \( E \) and points \( P \) and \( Q \) in \( E(\mathbb{F}_5) \) given by

\[
E : y^2 = x^3 + 2x - 1, \quad P = (0, 2), \quad Q = (4, 1).
\]

(a) Compute \( P + Q \).

(b) Compute \( 2P \).

Solution. If you’ve memorized the formulas for addition and doubling, you can of course use those. But I find it easier to just do the computation directly.

(a) The slope of \( PQ \) is \( \lambda = (2 - 1)/(0 - 4) = 1/2 = 3 \). (Remember we’re working mod 5.) So the line \( PQ \) is \( y - 2 = 1 \cdot (x - 0) \), i.e., the line \( y = x + 2 \). Substituting this into the equation for \( E \) gives

\[
(x + 2)^2 = x^3 + 2x - 1 \\
x^2 + 4x + 4 = x^3 + 2x - 1 \\
0 = x^3 - x^2 - 2x - 5 \\
0 = x^3 + 4x^2 + 3x \quad \text{remember, 5 = 0.}
\]

We know that 0 and 4 are roots (these are the \( x \)-coordinates of \( P \) and \( Q \)), so this cubic factors as

\[
x^3 + 4x^2 + 3x = x(x - 4)(x + 2) = x^3 - (4 + 2)x^2 + 4cx.
\]

Comparing either the \( x^2 \) or the \( x \) terms, we find that \( c = 2 \). Then using the equation \( y = x + 2 \) of the line \( PQ \), the \( y \)-coordinate of the third intersection point is \( c + 2 = 4 \), so \( PQ \) intersects \( E \) in the three points \( P \), \( Q \), and (2, 4). Then \( P + Q \) is the reflection, i.e., we need to change the sign of the \( y \)-coordinate. Hence \( P + Q = (2, -4) = (2, 1) \).

(b) Again, I’ll do the computation from scratch. Implicit differentiation gives \( 2y(dy/dx) = 3x^2 + 2 \), so the slope of the tangent line at \( P = (0, 2) \) is \( \lambda = (3x^2 + 2)/2y = (3 \cdot 0^2 + 2)/(2 \cdot 2) = 1/2 = 3 \). So the
tangent line $L$ is $y - 2 = 3(x - 0)$, i.e., the line $y = 3x + 2$. Substituting this into the equation for $E$ gives

$$(3x + 2)^2 = x^3 + 2x - 1$$
$$9x^2 + 12x + 4 = x^3 + 2x - 1$$
$$0 = x^3 - 9x^2 - 10x - 5$$
$$0 = x^3 + x^2$$

This has 0 as a double root, as it should since it is tangent to $E$ at $P$, and it factors as $x^3 + x^2 = x^2(x + 1)$. So the other intersection point in $E \cap L$ has $x = -1 = 4$. We compute the $y$-coordinate using the equation of $L$, thus $y = 3x + 2 = 3 \cdot 4 + 2 = 4$. Hence $E \cap L$ consists of the two points $P$ and $(4, 4)$. Reflecting this last point gives the answer, $2P = (4, -4) = (4, 1)$.

**Problem 4.** (10 points) Write your answer to this problem is words, not formulas or numbers. Write legibly.

(a) State one way that Elliptic Curve Public Key Cryptosystems are better than RSA or Lattice-Based Public Key Cryptosystems at equal security levels.
(b) State two ways that Lattice-Based Public Key Cryptosystems are better than RSA and Elliptic Curve Public Key Cryptosystems at equal security levels.
(c) State one way that the NTRU Public Key Cryptosystem is better than the GGH Public Key Cryptosystem at equal security levels.

**Solution.**

(a) Keys and ciphertexts are smaller (fewer bits).
(b) (1) Encryption and decryption are much faster.
   (2) Lattice-based systems are not (known to be) breakable by quantum computers
(c) NTRU keys are (much) smaller than GGH keys (fewer bits).

**Problem 5.** (10 points) There are three urns, each of which contains 10 balls numbered from 1 to 10. Alice randomly chooses 2 balls from the first urn, Bob randomly chooses 2 balls from the second urn, and Carl randomly chooses 2 balls from the third urn.

(a) What is the probability that at least one of Alice’s numbers matches one of Bob’s numbers?
(b) What is the probability that, among the 6 numbers chosen by Alice, Bob, and Carl, at least two of the numbers match?
Solution. (a) Think of Alice’s two numbers as the “winning” numbers. Then

\[
\Pr(\text{Bob wins}) = 1 - \Pr(\text{Bob loses})
\]

\[
= 1 - \Pr(\text{Bob’s first choice loses}) \cdot \Pr(\text{Bob’s second choice loses} \mid \text{Bob’s first choice already lost})
\]

\[
= 1 - \frac{8}{10} \cdot \frac{7}{9}
\]

\[
= \frac{34}{90} = \frac{17}{45}.
\]

(b) Note the problem is asking the following: Alice, Bob, and Carl are holding six numbers, and you supposed to compute the probability that among those six numbers, at least two of them are the same. Some people instead tried to compute the probability of the six numbers containing at least two matches. That’s a different problem.

Again it’s easier to compute the complementary probability. But you have to keep track of the fact that when you choose two numbers from the same urn, they’re always different. So here’s what happens.

(1) Alice choose two numbers. They’re automatically different.

(2) Bob chooses a number. The probability it is different from Alice’s two chosen numbers is \(\frac{8}{10}\).

(3) Bob chooses another number from the 9 remaining numbers in his urn. It never matches Bob’s already chosen number, and the probability that it is different from Alice’s chosen two numbers is \(\frac{7}{9}\). So far, we’ve just redone the computation from (a).

(4) Carl chooses a number. Since Alice and Bob already have 4 different numbers, the probability that Carl’s number doesn’t match any of theirs is \(\frac{6}{10}\).

(5) Carl chooses another number from the 9 remaining numbers in his urn. Since Alice and Bob already have 4 different numbers and Carl has already chosen a 5th different, the probability that Carl’s second number doesn’t match any of 5 previously chosen numbers is \(\frac{5}{9}\).
Using this information, we can compute
\[ \Pr(\text{get a match}) = 1 - \Pr(\text{all 6 numbers are different}) \]
\[ = 1 - \frac{8 \cdot 7 \cdot 6 \cdot 5}{10 \cdot 9 \cdot 10 \cdot 9} \]
\[ = 1 - \frac{1680}{8100} \]
\[ = \frac{6420}{8100} = \frac{107}{135} \]
So there’s close to an 80% chance of getting a match.

**Problem 6.** (15 points) Let \( N = pq \) be a product of two large primes. The value of \( N \) is public, but the values of \( p \) and \( q \) are not. Consider the following Digital Signature Scheme.

**Key Creation:** Alice chooses numbers \( k \) and \( a \) and computes
\[ b \equiv k^a \pmod{N}. \]
Her private signing key is \( k \), and her public verification key is the pair \((a,b)\).

**Signing:** To sign a document \( D \), Alice chooses a random number \( r \) and computes
\[ S \equiv D^r k^r \pmod{N}. \]
The document and signature consists of the triple \((D,r,S)\).

**Verification:** Bob accepts that the signature is valid if
\[ S^a \equiv D^a b^r \pmod{N}. \]

(a) Prove that if Alice has signed \( D \), then verification works.

(b) Suppose that Eve is given a signed document \((D,r,S)\). Let \( D' \) be a different document. Explain how Eve can create a signature on \( D' \) without knowing Alice’s private signing key.

(c) Suppose that Eve is given two signed documents
\[ (D_1,r_1,S_1) \text{ and } (D_2,r_2,S_2), \]
and suppose further that Alice’s chosen random numbers \( r_1 \) and \( r_2 \) happen to satisfy
\[ \gcd(r_1,r_2) = 1. \]
Explain how Eve can compute Alice’s private signing key \( k \).

**Solution.** (a) We have
\[ S^a \equiv (Dk^r)^a \equiv D^a k^r \equiv D^a b^r \pmod{N}. \]
(b) The triple \( (D', r, D'D^{-1}S) \) is a valid signature on \( D' \), where \( D^{-1} \) means the inverse of \( D \) modulo \( N \). To check that \( (D', r, D'D^{-1}S) \) is a valid signature on \( D' \), we compute

\[
(D'D^{-1}S)^a = (D')^a \cdot D^{-a} \cdot S^a \\
\equiv (D')^a \cdot D^{-a} \cdot (D^a b^r) \pmod N \\
\equiv (D')^a b^r \pmod N.
\]

(c) Since \( \gcd(r_1, r_2) = 1 \), Eve can find integers \( u_1, u_2 \) such that \( r_1 u_1 + r_2 u_2 = 1 \). Then

\[
S_1^{u_1} \cdot S_2^{u_2} \equiv (D_1 k^{r_1})^{u_1} \cdot (D_2 k^{r_2})^{u_2} \pmod N \\
\equiv D_1^{u_1} \cdot D_2^{u_2} k^{r_1 u_1 + r_2 u_2} \pmod N \\
\equiv D_1^{u_1} \cdot D_2^{u_2} k \pmod N.
\]

Since Eve knows all of the quantities \( S_1, S_2, D_1, D_2, u_1, u_2 \), she can compute

\[
k \equiv S_1^{u_1} \cdot S_2^{u_2} \cdot D_1^{-u_1} \cdot D_2^{-u_2} \pmod N.
\]

**Problem 7.** (20 points) Give clear concise statements of the following mathematical problems.

(a) The Diffie-Hellman problem for \( \mathbb{F}_p^* \).

(b) The elliptic curve discrete logarithm problem (ECDLP) for \( E(\mathbb{F}_p) \).

(c) The closest vector problem (CVP) for a lattice \( L \).

(d) A mathematical problem whose solution would enable Eve to break the RSA cryptosystem.

**Solution.** (a) The Diffie-Hellman problem for \( \mathbb{F}_p \) says: Given the values of \( g, g^a \), and \( g^b \) in \( \mathbb{F}_p \), compute the value of \( g^{ab} \).

(b) The elliptic curve discrete logarithm problem (ECDLP) for \( E(\mathbb{F}_p) \) says: Given points \( P, Q \in E(\mathbb{F}_p) \), find an integer \( n \) satisfying \( Q = nP \) in \( E(\mathbb{F}_p) \).

(c) The closest vector problem (CVP) for a lattice \( L \) says: Given an arbitrary vector \( \mathbf{w} \), find the vector in the lattice \( L \) that is closest to \( \mathbf{w} \).

(d) Eve can break RSA if she can factor integers \( N = pq \) that are a product of two (large) primes. She can also break RSA if she can solve congruences of the form \( x^e \equiv m \pmod N \) when \( N \) is as above.

**Problem 8.** (25 points) Let \( L \) be the lattice spanned by the vectors

\[
\mathbf{v}_1 = (1, 2, -1), \quad \mathbf{v}_2 = (-2, 1, 1), \quad \mathbf{v}_3 = (1, 0, 2).
\]

(a) Compute the dimension and the determinant of \( L \).

\[
\dim(L) = \boxdot, \quad \det(L) = \boxdot.
\]

---

**Math 1580—Solutions for Final Exam**

Page 7
(b) Compute the Hermite ratio of the basis \( \{v_1, v_2, v_3\} \) for the lattice \( L \). You may leave your answer in terms of roots of numbers, e.g., square roots, cube roots, etc.

Hermite Ratio = 

(c) Is the vector \( w_1 = (3, 2, 7) \) in the lattice \( L \)? Explain clearly why or why not.

(d) Is the vector \( w_2 = (0, 9, -7) \) in the lattice \( L \)? Explain clearly why or why not.

(e) For whichever of the vectors \( w_1 \) and \( w_2 \) in (c) and (d) that is not in \( L \), use Babai’s method to find a vector \( v \) in \( L \) that is close to the given non-lattice vector.

Solution. (a) \( \dim(L) = 3 \) and

\[
\det(L) = \det \begin{pmatrix}
1 & 2 & -1 \\
-2 & 1 & 1 \\
1 & 0 & 2
\end{pmatrix} = 13.
\]

(b) The formula for the Hermite ratio of a basis \( \{v_1, \ldots, v_n\} \) of a lattice \( L \) is

\[
\text{Hermite Ratio} = \left( \frac{\det(L)}{\|v_1\| \cdots \|v_n\|} \right)^{1/\dim(L)}.
\]

We already computed \( \det(L) = 13 \) and \( \dim(L) = 3 \). The lengths are

\[
\|v_1\| = \sqrt{1 + 4 + 1} = \sqrt{6},
\]

\[
\|v_2\| = \sqrt{4 + 1 + 1} = \sqrt{6},
\]

\[
\|v_3\| = \sqrt{1 + 0 + 4} = \sqrt{5},
\]

so

\[
\text{Hermite Ratio} = \left( \frac{13}{\sqrt{6} \cdot \sqrt{6} \cdot \sqrt{5}} \right)^{1/3} = \left( \frac{13}{6\sqrt{5}} \right)^{1/3}.
\]

You were not asked to compute this value as a decimal, but it turns out that it equals approximately 0.98954.

(c) We express \( w_1 \) as a linear combination of \( v_1, v_2, v_3 \). Thus

\[
(3, 2, 7) = x(1, 2, -1) + y(-2, 1, 1) + z(1, 0, 2).
\]

This leads to three equations

\[
3 = x - 2y + z
\]

\[
2 = 2x + y
\]

\[
7 = -x + y + 2z
\]
Now solve using any method you want. For example, subtract the last equation from twice the first to eliminate $z$. This gives

$$-1 = 3x - 5y.$$ 

So adding this to 5 times the second equation eliminates $y$ and gives

$$9 = 13x.$$ 

Thus $x = 9/13$ is not an integer, so $w_1$ is not in $L$. (To be in $L$, we would need $w_1$ to be a linear combination of $v_1, v_2, v_3$ using integer coefficients.)

(d) This is similar to (c). We set

$$(0, 9, -7) = x(1, 2, -1) + y(-2, 1, 1) + z(1, 0, 2),$$

which gives the three equations

$$0 = x - 2y + z$$
$$9 = 2x + y$$
$$-7 = -x + y + 2z$$

Subtracting the last equation from twice the first to eliminate $z$. This gives

$$7 = 3x - 5y.$$ 

Then adding this to 5 times the second equation eliminates $y$ and gives

$$52 = 13x.$$ 

Hence $x = 4$. Then $7 = 3x - 5y$ gives $5y = 3 \cdot 4 - 7 = 5$, so $y = 1$. Finally using one of the original equations gives $z = -x + 2y = -2$. Thus

$$w_2 = 4v_1 + v_2 - 2v_3 \in L$$

is a vector in the lattice $L$.

(e) We need to complete the computation in (c) and express $w_1$ as a linear combination of $v_1, v_2, v_3$ using non-integer coefficients. We found that $x = 9/13$. Then the equation $-1 = 3x - 5y = 27/13 - 5y$ gives $5y = 40/13$, so $y = 8/13$. And then using $3 = x - 2y + z = 9/13 - 16/13 + z$, we find that $z = 3 + 7/13 = 46/13$. Thus

$$w_1 = \frac{9}{13}v_1 + \frac{8}{13}v_2 + \frac{46}{13}v_3.$$ 

Babai’s method says to round each of the coefficients to the nearest integer. In this example, that means each of the coefficients gets rounded up, so a lattice vector that is close to $w_1 = (3, 2, 7)$ is

$$v = v_1 + v_2 + 4v_3 = (1, 2, -1) + (-2, 1, 1) + 4(1, 0, 2) = (3, 3, 8).$$
A quick inspection shows that this is a plausible answer, since $\mathbf{w}_1 - \mathbf{v} = (0, -1, -1)$ is fairly short; it has length $\sqrt{2}$.

**Problem 9.** (10 points) Alice creates an NTRU private key by choosing polynomials

$$f(X) = f_0 + f_1X + \cdots + f_{N-1}X^{N-1},$$
$$g(X) = g_0 + g_1X + \cdots + g_{N-1}X^{N-1},$$

having small coefficients and uses them to compute her public key

$$h(X) \equiv f(X)^{-1} \cdot g(X) \equiv h_0 + h_1X + \cdots + h_{N-1}X^{N-1} \pmod{q}.$$

(a) Write down a matrix whose rows generate a lattice $L$ that contains the short vector

$$(f_0, f_1, \ldots, f_{N-1}, g_0, g_1, \ldots, g_{N-1}).$$

This should be a matrix that Eve, Alice’s adversary, can write down using only the publicly available information $N$, $q$, and $h(X)$.

(b) Let $i$ be an integer between 1 and $N - 1$. Suppose that Alice instead chooses $X^{N-i} \cdot f(X)$ and $X^{N-i} \cdot g(X)$ as her private key, where remember that we always set $X^N = 1$. What would the coefficients of her private key and her public key look like?

(c) Prove that for every integer $i$ between 1 and $N - 1$, the lattice $L$ in (a) contains the short vector

$$(f_i, f_{i+1}, \ldots, f_{N-1}, f_0, f_1, \ldots, f_{i-1}, g_i, g_{i+1}, \ldots, g_{N-1}, g_0, g_1, \ldots, g_{i-1}).$$

(d) Suppose that Eve manages to find one of the vectors in (c), or equivalently, suppose that she manages to find the polynomials $X^{N-i} \cdot f(X)$ and $X^{N-i} \cdot g(X)$, but that she does not know the value of $i$. She uses this information to decrypt a ciphertext $e(X)$ that Bob has sent to Alice. If Bob’s plaintext is

$$m(X) = m_0 + m_1X + \cdots + m_{N-1}X^{N-1},$$

what polynomial will Eve recover?

**Solution.** (a) This is the NTRU lattice that we discussed in class. It is a $2N$-by-$2N$ matrix that has four $N$-by-$N$ block. Three of the blocks are very simple, either 0, the identity, and $q$ times the identity. The fourth block is a circulant matrix with the coefficients of $h(X)$. Thus $L$
is generated by the rows of the following matrix:

\[
\begin{pmatrix}
1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\
0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\
0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q
\end{pmatrix}.
\]

(b) The effect of multiplying a polynomial by \(X^{N-i}\) is to shift the coefficients \(i\) places to the left, with the bottom coefficients wrapping around to the top. We do the computation explicitly for \(f(X)\).

\[
X^{N-i} \cdot f(X) = X^{N-i} \cdot (f_0 + f_1X + \cdots + f_{i-1}X^{i-1} + f_iX^i + f_{i+1}X^{i+1} + \cdots + f_{N-1}X^{N-1})
\]

\[
= f_0X^{N-i} + f_1X^{N-i+1} + \cdots + f_{i-1}X^{N-i+i-1} + f_iX^{N-i+i} + f_{i+1}X^{N-i+i+1} + \cdots + f_{N-1}X^{N-i+N-1}
\]

\[
= f_0X^{N-i} + f_1X^{N-i+1} + \cdots + f_{i-1}X^{N-i} + f_iX^{N-i+1} + \cdots + f_{N-1}X^{N-i-1}.
\]

Now it is simply a matter of writing the terms starting with the constant term, then the term with \(X\), then \(X^2\), etc. This gives

\[
X^{N-i} \cdot f(X) = f_i + f_{i+1}X + \cdots + f_{N-1}X^{N-i-1} + f_0X^{N-i} + f_1X^{N-i+1} + \cdots + f_{i-1}X^{N-i-1}.
\]

The formula for \(X^{N-i} \cdot g(X)\) is the same,

\[
X^{N-i} \cdot g(X) = g_i + g_{i+1}X + \cdots + g_{N-1}X^{N-i-1} + g_0X^{N-i} + g_1X^{N-i+1} + \cdots + g_{i-1}X^{N-i-1}.
\]

However, the formula for the “new” public key \(h(X)\) is given by

\[
(X^{N-i} \cdot f(X))^{-1} \cdot (X^{N-i} \cdot g(X)) \equiv (X^{-N+i} \cdot f(X)^{-1}) \cdot (X^{N-i} \cdot g(X)) \pmod{q}
\]

\[
\equiv f(X)^{-1} \cdot g(X) \pmod{q},
\]

so Alice’s new public key \(h(X)\) is the same as Alice’s original public key \(h(X)\). This is reasonable, since we changed \(f\) and \(g\) by multiplying them by the same quantity, then we divided \(g\) by \(f\), so that multiplier cancels.
(c) One way to do this is via the same sort of computation that we used in class to show that \((f_0, \ldots, f_{N-1}, g_0, \ldots, g_{N-1})\) is in \(L\). But it’s easier to argue as follows. The computation that we did in part (b) shows that if Alice had used the private key polynomials \(X^{N-i} \cdot f(X)\) and \(X^{N-i} \cdot g(X)\), she would have gotten the same public key polynomial \(h(X)\), so Eve would have created the same lattice \(L\). Hence from (a), the lattice \(L\) contains the vector whose coordinates are the coefficients of \(X^{N-i} \cdot f(X)\) followed by the coefficients of \(X^{N-i} \cdot g(X)\). The computation in (b) shows that this is precisely the vector 
\[
(f_i, f_{i+1}, \ldots, f_{N-1}, f_0, f_1, \ldots, f_{i-1}, g_i, g_{i+1}, \ldots, g_{N-1}, g_0, g_1, \ldots, g_{i-1}).
\]
that we’re supposed to show is in \(L\).

(d) You might think that Eve will recover the coefficients of Bob’s plaintext shifted, but that’s not the case. She will actually get Bob’s plaintext exactly. To see why, we start with Bob’s ciphertext, which has the form
\[
e(X) \equiv p \cdot r(X) \cdot h(X) + m(X) \pmod{q}.
\]
We’re assuming that Eve knows \(X^{N-i} \cdot f(X)\), so she starts by computing
\[
X^{N-i} \cdot f(X) \cdot e(X)
\]
\[
\equiv X^{N-i} \cdot f(X) \cdot p \cdot r(X) \cdot h(X) + X^{N-i} \cdot f(X) \cdot m(X) \pmod{q},
\]
\[
\equiv X^{N-i} \cdot f(X) \cdot p \cdot r(X) \cdot f(X)^{-1} \cdot g(X) + X^{N-i} \cdot f(X) \cdot m(X) \pmod{q},
\]
\[
\equiv X^{N-i} \cdot p \cdot r(X) \cdot g(X) + X^{N-i} \cdot f(X) \cdot m(X) \pmod{q}.
\]
Eve center-lifts this polynomial so that it’s coefficients are between \(-\frac{1}{2}q\) and \(\frac{1}{2}q\), and then reduces modulo \(p\) to recover the polynomial
\[
X^{N-i} \cdot f(X) \cdot m(X) \pmod{p}.
\]
She next multiplies by the mod \(p\) inverse of \(X^{N-i} \cdot f(X)\). This causes both the \(f(X)\) and the \(X^{N-i}\) to cancel, leaving Bob’s plaintext \(m(X)\).

**Problem 10.** (15 points) Let \((N, p, q, d)\) be parameters that are used for the NTRU cryptosystem. For this problem, you can express your answers using binomial coefficients.

(a) Recall that \(T(d_1, d_2)\) denotes the set of polynomials \(a_0 + a_1 X + \cdots + a_{N-1} X^{N-1}\) whose coefficients consist of \(d_1\) ones, \(d_2\) negative ones, and the remaining \(N - d_1 - d_2\) coefficients are zero. How big is the set \(T(d_1, d_2)\)?

\[
\#T(d_1, d_2) =
\]
(b) Recall that an NTRU private key consists of a pair of polynomials \((f, g)\), where \(f\) is randomly chosen from the set \(T(d + 1, d)\) and \(g\) is randomly chosen from the set \(T(d, d)\). How many different NTRU key pairs \((f, g)\) are there?

\[
\text{# of key pairs } (f, g) = \binom{N}{d+1} \cdot \binom{N-d-1}{d}.
\]

(c) Let \((f, g)\) be Alice’s NTRU private key and \(h\) her NTRU public key. One way for Eve to find Alice’s private key is to try all pairs \((f, g)\) and check if \(f^{-1} \cdot g \mod q\) is equal to \(h\). A quicker method is to try all \(f \in T(d+1, d)\) and compute \(f \cdot h \mod q\). Explain how Eve will be able to tell when she has found a good possibility for \(f\).

(d) **Bonus Problem** (This is very challenging. Don’t spend time on it unless you’re done with the rest of the exam and feel like a challenge. Write your answer on the back of this page.) Eve’s attack in (c) requires her to check all of the elements in \(T(d+1, d)\). Devise a collision-style attack that only requires her to check some small multiple of \(\sqrt{\#T(d+1, d)}\) cases.

**Solution.** (a) Choose \(d_1\) of the \(N\) coefficients to be 1’s, and choose \(d_2\) of the remaining \(N - d_1\) coefficients to be \(-1\)’s, so the total number polynomials in \(T(d_1, d_2)\) is \(\binom{N}{d_1} \cdot \binom{N-d_1}{d_2}\). It’s also fine if you write this out with factorials and cancel some terms to get \(\frac{N!}{d_1! \cdot d_2! \cdot (N-d_1-d_2)!}\).

(b) From (a) this is

\[
\#T(d+1, d) \cdot \#T(d, d) = \binom{N}{d+1} \cdot \binom{N-d-1}{d} \cdot \binom{N}{d} \cdot \binom{N-d}{d}.
\]

(c) If Eve has the right \(f\), then \(f \cdot h \mod q\) will equal \(g\), which has only very small coefficients. More precisely, it will be in \(T(d, d)\). If she has the wrong \(f\), then the coefficients of \(f \cdot h \mod q\) will look random mod \(q\), so it’s highly unlikely to be in \(T(d, d)\). More generally, Eve will find all multiples \(X^{N-1} \cdot f(X)\) of \(f(X)\), since they will yield shifts of \(g(X)\).

(d) The idea for this collision algorithm is due to Andrew Odlyzko, a world-class researcher in both cryptography and analytic number theory. I will just sketch how it works. The idea is to write \(f(X)\) as \(f(X) = F_1(X) + X^{(N-1)/2}F_2(X)\), where \(F_1(X)\) and \(F_2(X)\) each has degree (roughly) \(N/2\). One also has to decide how many plus ones and how many minus ones are in \(F_1(X)\), with \(F_2(X)\) being allocated the remaining weights.
others. More precisely, for every $0 \leq i \leq d + 1$ and every $0 \leq j \leq d$, Eve lets $F_1(X)$ vary over all polynomials of degree $N/2$ in $T(i, j)$ and $F_2(X)$ vary over all polynomials of degree $N/2$ in $T(d + 1 - i, d - j)$. Next make two lists.

List #1: $F_1(X) \cdot h(X) \mod q$ for all $F_1(X)$. 

List #2: $-X^{(N-1)/2} F_2(X) \cdot h(X) \mod q$ for all $F_2(X)$.

We observe that there is some choice of $F_1$ and $F_2$ so that 

$$f(X) = F_1(X) + X^{(N-1)/2} F_2(X),$$

and we know that the product 

$$f(X) \cdot h(X) \equiv g(X) \pmod q$$

has small coefficients. It follows that for the right choice of $F_1$ and $F_2$, we have 

$$g(X) \equiv f(X) \cdot h(X) \equiv F_1(X) \cdot h(X) + X^{(N-1)/2} F_2(X) \cdot h(X),$$

and hence the polynomial 

$$-X^{(N-1)/2} F_2(X) \cdot h(X) \equiv F_1(X) \cdot h(X) - g(X) \pmod q$$

is quite close to the polynomial $F_1(X) \cdot h(X) \pmod q$. Thus Eve needs to find an approximate collision between her two lists, i.e., she is looking for a polynomial in List #1 whose coefficients “almost” match the coefficients of a polynomial in List #2. There are various ways one might efficiently do this, but they’re too complicated to describe here, so I’ll let you download and read the original description, which is reference [101] in our textbook.