CS 70: RSA, Polynomial Applications, Un__tability

1 RSA

1. Suppose Alice wishes to send Bob a confidential message using RSA. For this, Bob must first set up his public-private key pair. Below, we show the choices Bob made in picking his keys, where he makes at least one mistake.

   Suppose that Bob chooses primes p = 7, q = 13. (Assume these are large enough.)

   He computes N = pq = 91.

   Then Bob chooses e = 3 so his public key is (3, 91).

   Finally Bob chooses d = 61 which is his private key.

   What mistake did Bob make?

   Solution: In RSA, we define $d = e^{-1} \pmod{(p-1)(q-1)}$. This forces e to be relatively prime to $(p-1)(q-1)$. But if $e = 3$ in this case, because $(p-1)(q-1) = 12 * 6 = 72$, 3 is not relatively prime to 72! So Bob needs to pick a better e.

2 Polynomials and (mod p)

1. Why do we use a prime p as a modulus so often?

   Solution: Working modulo p has many advantages, stemming from the fact that primes have very few factors. We get the property that $ab = 0 \pmod{p} \implies a = 0 \pmod{p}$ or $b = 0 \pmod{p}$. In addition, every number that isn't congruent to zero has an inverse. This makes it 'look' a lot like the real numbers or the rationals. This isn't the case for example if our modulus is $n = 12$, because $3 \cdot 4 = 0 \pmod{12}$,

2. Is the polynomial $3x^3 + 5x^2 + 4x + 2$ perfectly divisible by $x - 3 \pmod{7}$?

   Solution: Yes it is! When doing polynomial divison, after you've finished a single step of multiplying, make sure you take all coefficients mod 7. Then you should find that

   $$3x^3 + 5x^2 + 4x + 2 = (x - 3)(3x^2 + 4)$$

3. For $k < 7$, How many unique degree-k polynomials are there mod 7? For $k < p$, Mod p?

   Solution: A degree-k polynomial is completely determined by $k + 1$ points, for example when the polynomial is evaluated at $x = 0, 1, 2, \ldots k$. There are 7 choices for each of these points, so we get 7^{k+1}. In general, we get p^{k+1}. There are

4. Suppose $x = 4 \pmod{7}$ and $x = 7 \pmod{11}$. What is $x \pmod{77}$?

   Solution: The CRT does apply to this problem, but it doesn't help us find a solution (at least what we learned in class). To figure out a number, just enumerate all numbers less than 77 that are congruent to 7 mod 11, and figure out which of those is congruent to 4 mod 11. 18 is a number that comes up relatively early that satisfies this, but CRT does tell us that it is the unique number less than 77 that satistfiest these two congruences.

5. Let $r^2 = 1 \pmod{n}$. Show that if $r - 1$ and n are relatively prime, then $r = n - 1 \pmod{n}$.

   Solution: If $r - 1$ and n are relatively prime, then there must be x and y such that $x(r-1) + yn = 1$. Multiplying both sides by $r + 1$, we have that $x(r^2 - 1) + yn(r + 1) = (r + 1)$. If we take both sides mod n, then we have $0 = r + 1 \pmod{n}$, which means $r = -1 = n - 1 \pmod{n}$.

6. Let $x_1, \ldots x_n$ be integers, and p a prime number. Show that $(x_1 + \ldots + x_n)^p = x_1^p + \ldots + x_n^p \pmod{p}$.

   Solution: Applying Fermat's last theorem on the left, we have that $(x_1 + \ldots + x_n)^p = x_1 + \ldots + x_n$. Applying it to each term on the right, we have that $x_1^p + \ldots + x_n^p = x_1 + \ldots + x_n$. Since these are the same, the two sides must be equal.

7. Consider a two-variable polynomial $Z(x, y) = P(x)Q(y)$ modulo a prime p where $P(x)$ and $Q(y)$ are nonzero degree-d polynomials where $d < p$. What is the maximum number of distinct pairs of (i, j) that satisfy $Z(i, j) = 0 \pmod{p}$?

   Solution: Since we're working modulo a prime, we have that $Z(x, y) = 0$ only if $P(x) = 0$ or $Q(y) = 0$, or perhaps both. Both P and Q can have at most d distinct roots each, because they have degree at most d (this is just a fact of polynomials). Let the roots of P be $x_1, x_2, \ldots x_d$ and the roots of Q be $y_1, y_2, \ldots y_d$.

   Then notice that $Z(x_1, 0) = Z(x_1, 1) = Z(x_1, 2) = 0$, in other words, $Z(x_i, n)$ is zero for any x_i and n! There are d choices for one of the x_i and p choices for the n, so there are pd ways this can happen. Something similar happens lookig at $Z(n, y_i)$. So if we add up all these pairs, we get $2pd$. But notice that we double counted pairs like $Z(x_i, y_i)$, as these look like both $Z(x_i, n)$ and $Z(n, y_i)$. There are exactly d^2 of these, so subtract these from $2pd$ to get the answer.

   At the end, we get at most $2pd - d^2$ places where the polynomial can be zero.

3 Polynomial Applications

1. Assume we send n packets to Alice, and we know that $p = 20\%$ of any packets we sent are lost. How many packets should we send under standard error correcting schemes to ensure Alice can recover our message? What happens if $p = 0.9$?

   Solution: We must consider what happens *after* we send the packets! If we only send 20% extra packets, we will lose 20% of the original packets we send AND 20% of the 20% additional packets we send, meaning we receive $\frac{24}{25}n$ uncorrupted packets in total, which isn't enough. So in the end, we want $(n + k)(1 - \alpha) \geq n$ to ensure the number of uncorrupted packets received is at least n. By doing some algebraic manipulation we find that

   $$k \geq n\frac{\alpha}{1 - \alpha}$$

   .

2. Given the error polynomial from Berlekamp-Welch algorithm, $x^2 + 3x + 2 \pmod{11}$, for what 'x' values are the points corrupted?

   Solution: Factor the polynomial as $(x + 1)(x + 2)$. The polynomial is 0 exactly at the points that there are errors, which would be $x = -1, x = -2$, but since we are working mod 11 these are $x = 10, x = 9$.

3. We'll prove how Berlekamp-Welch can work, and maybe provide some reason for why it needs $n + 2k$ points. Alice encoded her n-length message in P, then sent $n + 2k$ packets $[P(1) = x_1], [P(2) = x_2], [P(n + 2k) = x_{n+2k}]$ over to us, of which at most k have been corrupted.

   (a) What degree polynomial P did Alice use to send us the message?

   Solution: The polynomial Alice sent to us encoded n messages, so it has degree $n - 1$.

   (b) Let's call R the polynomial made up of the $n+2k$ packets we received $[R(1) = r_1], [R(2) = r_2], [R(n+2k) = r_{n+2k}]$. With at most k corruptions, for how many points among $1, \ldots, n+2k$ must P and R agree on (i.e. $P(x) = R(x)$)?

   Solution: They must agree on at least $n + k$ points. This is because if there are most k points that are different, there must be at least $(n + 2k) - k = n + k$ points that are the same.

   (c) Now assume we have two polynomials of degree part a) that both agree with P on part b) number of points. How many points must these two polynomials agree on? What can we conclude from this?

   Solution: Let these two polynomials be R_1 and R_2. You can think of this situation as R_1 and R_2 being two possible messages we can receive from Alice that have been corrupted differently. If R_1 only differs from P on k points and P differs from R_2 on k points, this means that R_1 and R_2 differ from *each other* on at most $2k$ points. Then they must share n points! Because a degree $n - 1$ polynomial is completely determined by those n points, there can only be one degree $n - 1$ polynomial that agrees with at least $n + k$ points of R_1 and at least $n + k$ points of R_2.

4. The standard polynomial secret sharing is being used and we are working mod 5. Three shares are required to determine the secret, encoded as P(0). We have the following shares: P(1) = 2, P(2) = 0, P(3) = 2. What is the secret?

Solution: If we require 3 shares, that makes the polynomial degree 2. Setup a system of equations for $P(x) = p_0 + p_1 x + p_2 x^2$ by plugging in all the values:

$$p_0 + p_1 + p_2 = 2$$
$$p_0 + 2p_1 + 4p_2 = 0$$
$$p_0 + 3p_1 + 9p_2 = 2$$

We end up with $p_0 = 3$, $p_1 = 2$, $p_2 = 2$, so $P(x) = 3 + 2x + 2x^2$, so $P(0) = 3$ and the secret is 3.

5. How might we split a secret up among n people that requires two numbers? The last problem encoded the secret as $P(0)$, but what if one number is not enough to describe our secret?

   Solution: This problem is written a little badly, but the idea is if we have a secret that is actually a pair (a, b), just design a degree $n-1$ polynomial such that $P(0) = a$, $P(1) = b$, by some method of interpolation. If enough people get together to reconstruct the polynomial, then they can recover both a and b by evaluating the polynomial at both points.

4 Un(coun,compu)tability

1. Is the powerset of $\mathbb{N}$ countable (the set of all subsets of $\mathbb{N}$)? How would you prove this?

   Solution: See the next worksheet

2. Are the integers $\mathbb{Z}$ countable? How about pairs of integers where one of the pair must be zero?

   Solution: See the next worksheet

3. Is a countable union of countable subsets countable? This means $\bigcup_i U_i$ where $i \in \mathbb{N}$.

   Solution: See the next worksheet

4. Consider the following program:

```
def is_mod_2(P):
    if (P implements the mod 2 function):
        return True
    else:
        return False
```

   Show it cannot exist as a program.

   (*Hint*: Assume it exists, and show that it solves the halting problem. Because a program to solve the halting problem doesn't exist, neither can this one!)

   Solution: See the next worksheet

5. Show that there exist numbers in $\mathbb{R}$ that cannot be computed. (Wow!!!)

   Solution: See the next worksheet