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?

2 Polynomials and (mod p)

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

2. Is the polynomial $3x^3 + 5x^2 + 4x + 2$ perfectly divisible by $x - 3$ (mod 7)?

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

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

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

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

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 $Z(i, j) = 0 \pmod p$?

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 $p = 0.9$?

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

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?

   (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. $P(x) = R(x)$?

   (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?

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?

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?

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?

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

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

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!)

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