CS 70 Discrete Mathematics and Probability Theory
Fall 2015 Rao HW 6

Due Wednesday Oct 7 at 10PM

1. **(Breaking RSA)**

   (a) Eve is not convinced she needs to factor $N = pq$ in order to break RSA. She argues: "All I need to know is $(p-1)(q-1)$... then I can find d as the inverse of e mod $(p-1)(q-1)$. This should be easier than factoring N". Prove Eve wrong, by showing that if she knows $(p-1)(q-1)$, she can easily factor N (thus showing finding $(p-1)(q-1$) is at least as hard as factoring N).. Assume Eve has a friend Wolfram, who can easily return the roots of polynomials over $\mathbb{R}$ (this is, in fact, easy).

   Answer: Let $a = (p-1)(q-1)$. If Eve knows $a = (p-1)(q-1) = pq-(p+q)+1$, then she knows $p+q = pq-a+1$ (note that $pq = N$ is known too). In fact, p and q are the two roots of polynomial $f(x) = x^2-(p+q)x+pq$ because $x^2-(p+q)x+pq = (x-p)(x-q)$. Since she knows $p+q$ and pq, she can give the polynomial $f(x)$ to Wolfram to find the two roots of $f(x)$, which are exactly p and q.

   (b) When working with RSA, it is not uncommon to use $e = 3$ in the public key. Suppose that Alice has sent Bob, Carol, and Dorothy the same message indicating the time she is having her birthday party. Eve, who is not invited, wants to decrypt the message and show up to the party. Bob, Carol, and Dorothy have public keys $(N_1, e_1), (N_2, e_2), (N_3, e_3)$ respectively, where $e_1 = e_2 = e_3 = 3$. Furthermore assume that N_1, N_2, N_3 are all different. Alice has chosen a number $0 \leq x < \min\{N_1, N_2, N_3\}$ which indicates the time her party starts and has encoded it via the three public keys and sent it to her three friends. Eve has been able to obtain the three encoded messages. Prove that Eve can figure out x. First solve the problem when two of N_1, N_2, N_3 have a common factor. Then solve it when no two of them have a common factor. Again, assume Eve is friends with Wolfram as above.

   Hint: The concept behind this problem is the Chinese Remainder Theorem: Suppose $n_1, \ldots, n_k$ are positive integers, that are pairwise co-prime. Then, for any given sequence of integers $a_1, \ldots, a_k$, there exists an integer x solving the following system of simultaneous congruences:

$$\begin{cases} x \equiv a_1 (\bmod n_1) \\ x \equiv a_2 (\bmod n_2) \\ \ldots \\ x \equiv a_k (\bmod n_k) \end{cases}$$

   Furthermore, all solutions x of the system are congruent modulo the product, $N = n_1 \ldots n_k$. Hence: $x \equiv y (\bmod n_i) for 1 \leq i \leq k \Leftrightarrow x \equiv y (\bmod N)$

   Answer: Eve first tests the GCD of all pairs of N_1, N_2, N_3. Let $d_1 = gcd(N_1, N_2)$, $d_2 = gcd(N_2, N_3)$ and $d_3 = gcd(N_1, N_3)$. Then there are two cases:

   case 1 If one of the d_1, d_2 and d_3 is greater than 1, it must be one of the prime factors p of the two N_i's. The other prime factor q can be recovered by $q = \frac{N_i}{p}$. Therefore we can factorize one of the N_i's and once we do that RSA is broken.

case 2 If $d_1 = d_2 = d_3 = 1$, it means all pairs of the N_i's are coprime. Let the three encoded messages be y_1, y_2, y_3. Since the messages are encoded by RSA with public keys $(N_1, 3)$, $(N_2, 3)$ and $(N_3, 3)$, we have:

$$x^3 \equiv y_1 \bmod N_1$$
$$x^3 \equiv y_2 \bmod N_2$$
$$x^3 \equiv y_3 \bmod N_3$$

Since all pairs of N_1, N_2, N_3 are coprime, By using the Chinese Remainder Theorem, we can solve the above system of congruence equations. Let the solution be

$$x^3 \equiv x_0 \bmod N_1N_2N_3$$

with $0 \le x_0 < N_1N_2N_3$. Since $x < N_1, N_2, N_3$, $x^3 < N_1N_2N_3$ and thus $x^3 = x_0$. We can take the cube root of x_0 and recover the original message $x = x_0^{1/3}$. In this problem, the trick is that we were able to convert a problem of finding cube-roots mod a prime (which is hard) into finding cube-roots in the integers (which is easy).

2. **(Polynomial Interpolations)**

(a) Consider the set of four points $\{(0,1), (1,-2), (3,4), (4,0)\}$, construct the unique degree-3 polynomial (over the reals) that passes through these four points by writing down and solving a system of linear equations.

Answer: Suppose the unique degree 3 polynomial passing through the four given points is

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

The coefficients of $p(x)$ satisfy the following linear equations:

$$p(0) = 1 \Rightarrow a_0 = 1 \quad (1)$$
$$p(1) = -2 \Rightarrow a_0 + a_1 + a_2 + a_3 = -2 \quad (2)$$
$$p(3) = 4 \Rightarrow a_0 + 3a_1 + 9a_2 + 27a_3 = 4 \quad (3)$$
$$p(4) = 0 \Rightarrow a_0 + 4a_1 + 16a_2 + 64a_3 = 0 \quad (4)$$

Substituting $a_0 = 1$ and subtracting (i) 3 times equation (2) from equation (3) and (ii) 4 times equation (2) from equation (4), we obtain the following simultaneous equations:

$$6a_2 + 24a_3 = 12$$
$$12a_2 + 60a_3 = 11$$

Solving for a_2 and a_3, we obtain $a_2 = 76/12$ and $a_3 = -13/12$. Substituting in equation (2) we obtain $a_1 = -99/12$. Hence

$$p(x) = (12 - 99x + 76x^2 - 13x^3)/12$$

is the unique degree 3 polynomial passing through the given points.

(b) Use Lagrange interpolation to find a polynomial $p(x)$ of degree at most 2 that passes through the points $(1,2)$, $(2,3)$, and $(3,5)$, working in $GF(7)$. In other words, we want $p(x)$ to satisfy $p(1) \equiv 2 \pmod 7$, $p(2) \equiv 3 \pmod 7$, and $p(3) \equiv 5 \pmod 7$. Show your work clearly and use the same notations as in Lecture Note 8.

Answer: First we would find the three $\Delta_i(x)$ polynomials.

$$\begin{aligned}
\Delta_1(x) &\equiv \frac{(x-2)(x-3)}{(1-2)(1-3)} \equiv \frac{(x-2)(x-3)}{2} \equiv 4(x-2)(x-3) \equiv 4x^2+x+3 \pmod 7\\
\Delta_2(x) &\equiv \frac{(x-1)(x-3)}{(2-1)(2-3)} \equiv \frac{(x-1)(x-3)}{-1} \equiv -(x-1)(x-3) \equiv 6x^2+4x+4 \pmod 7\\
\Delta_3(x) &\equiv \frac{(x-1)(x-2)}{(3-1)(3-2)} \equiv \frac{(x-1)(x-2)}{2} \equiv 4(x-1)(x-2) \equiv 4x^2+2x+1 \pmod 7
\end{aligned}$$

Next we compute $p(x)$:

$$\begin{aligned}
p(x) &\equiv 2\Delta_1(x)+3\Delta_2(x)+5\Delta_3(x) \pmod 7\\
&\equiv 2(4x^2+x+3)+3(6x^2+4x+4)+5(4x^2+2x+1) \pmod 7\\
&\equiv x^2+2x+6+4x^2+5x+5+6x^2+3x+5 \pmod 7\\
&\equiv 4x^2+3x+2 \pmod 7
\end{aligned}$$

3. **(Proofs about polynomials)** In this problem, you will give two different proofs of the following theorem: For every prime p, every polynomial over $GF(p)$, even polynomials with degree $\geq p$, is equivalent to a polynomial of degree at most $p-1$. (Two polynomials f, g over $GF(p)$ are said to be equivalent iff $f(x) = g(x)$ for all $x \in GF(p)$.)

(a) Show how the theorem follows from Fermat's Little Theorem. **Answer:** From Fermat's Little Theorem, we know $\forall x \not\equiv 0$, $x^{p-1} \equiv 1 \pmod p$.
Multiplying both sides by x, and noting that $0^p \equiv 0 \pmod p$, we can see that

$$\forall x,\ x^p \equiv x \pmod p$$

Therefore, any x^k, where $k \geq p$ will be equivalent to x^n, where $n \in \{0,1,\ldots,p-1\}$, and will have a degree at most $p-1$.

(b) Now prove the theorem using what you know about Lagrange interpolation. **Answer:** Since a polynomial f of degree d is described completely by $d+1$ points, we cannot specify a polynomial of degree $\geq p$ because $f(p) = f(0)$ (and so forth for other values larger than p) over $GF(p)$. Thus, we can only specify at most p unique points. Therefore, every polynomial is equivalent to a polynomial of degree at most $p-1$.

4. **(Poker mathematics)** A *pseudo-random number generator* is a way of generating a large quantity of random-looking numbers, if all we have is a little bit of randomness (known as the *seed*). One simple scheme is the *linear congruential generator*, where we pick some modulus m, some constants a, b, and a seed x_0, and then generate the sequence of outputs $x_1, x_2, x_3, x_4 \ldots$ according to the following equation:

$$x_{t+1} = (ax_t + b) \bmod m$$

(Notice that $0 \leq x_t < m$ holds for every t.)

You've discovered that a popular web site uses a linear congruential generator to generate poker hands for its players. For instance, it uses x_0 to pseudo-randomly pick the first card to go into your hand, x_1 to pseudo-randomly pick the second card to go into your hand, and so on. For extra security, the poker site has kept the parameters a and b secret, but you do know that the modulus is $m = 2^{31} - 1$ (which is prime).

Suppose that you can observe the values x_0, x_1, x_2, x_3, and x_4 from the information available to you, and that the values $x_5, \dots, x_9$ will be used to pseudo-randomly pick the cards for the next person's hand. Describe how to efficiently predict the values $x_5, \dots, x_9$, given the values known to you.

Answer: **Answer 1:** We know

$$\begin{aligned} x_1 &\equiv ax_0 + b \pmod{m} \\ x_2 &\equiv ax_1 + b \pmod{m} \end{aligned}$$

Because we know x_0, x_1, and x_2, this is a system of two equations with two unknowns (namely, a and b). So we can solve for a and b. More explicitly, by subtracting the first equation from the second, we get

$$x_2 - x_1 \equiv a(x_1 - x_0) \pmod{m}$$

If $x_0 \equiv x_1 \pmod{m}$ then by induction on n we see $x_n \equiv x_0 \pmod{m}$ for all n which allows us to immediately calculate x_5, x_6, x_7, x_8 and x_9. So suppose $x_0 \not\equiv x_1 \pmod{m}$. Then $x_1 - x_0$ is invertible modulo m (because m is prime, therefore $\gcd(x_1 - x_0, m) = 1$), and we see

$$a \equiv (x_2 - x_1)(x_1 - x_0)^{-1} \pmod{m}$$

Once we know a, we can plug in the known value of a into the first equation and solve for b:

$$b \equiv x_1 - ax_0 \equiv x_1 - x_0(x_2 - x_1)(x_1 - x_0)^{-1} \pmod{m}$$

Since we know a modulo m and b modulo m, we can compute x_5, x_6, x_7, x_8 and x_9.

Answer 2: Alternatively, we could start by solving for b. Multiplying the first equation by x_1, multiplying the second equation by x_0, and subtracting gives $x_1^2 - x_2x_0 \equiv b(x_1 - x_0) \pmod{m}$, and then

$$b \equiv (x_1^2 - x_0x_2)(x_1 - x_0)^{-1} \pmod{m}$$

Now we can plug in the known value of b into the first equation and solve for a, and continue as before.

5. **(How many errors?)** Suppose that the message we want to send consists of 10 numbers. We find a polynomial of degree 9 which goes through all these points and evaluate it on 25 points. How many erasure errors can we recover from?

Answer: For k erasure errors, we need to send k additional packets for a total of $n + k$. Here, $n = 10$ and $n + k = 25$, so we can recover from 15 erasure errors.

For k general errors, we need $2k$ additional packets. We have 15 additional packets, so we may recover from $\lfloor 15/2 \rfloor = 7$ general errors.

6. **(Why work with primes?)** In class, you learned about erasure codes and error correcting codes, and prime numbers played a central role in both kinds of codes – since all calculations were supposed to be done modulo a *prime number*. In this problem, we will see why this is a crucial requirement, and explore what happens if this requirement is relaxed in a naive manner.

   For this problem, assume that Alice wants to send n packets to Bob, across an "erasure channel" (Check detailed definition below). Let us say all calculations are done modulo $N = 12$ (note that this is *not* a prime number).

   Erasure Channel: let us say Alice sends $n+1$ packets to Bob, and Bob receives at least n of these packets intact. That is, the channel can erase at most 1 packet, and if it does so, Bob gets to know which packet was erased (although he does not know the contents of the erased packet).

   (a) Suppose $n = 1$. That is, Alice wants to send only 1 packet to Bob (plus one redundant packet to compensate for erasure). Would the scheme discussed in class work with $N = 12$? What are all the possible 2-packet lists that Alice could transmit? In each case, would Bob be able to recover Alice's message in spite of a possible erasure? Would Alice or Bob face any problems because they are doing their calculations modulo 12?

   Answer: If Alice wants to send only 1 packet to Bob, then her "message" is just a single integer m, drawn from the set $\{0,\ 1,\ 2,\ \ldots\ N-1\}$. Let's call this set $\texttt{range}(N)$. Then, following the scheme discussed in class, Alice will now try to find a polynomial $P(x)$ of degree 0 that evaluates to m at $x = 1$. This polynomial, of course, has to be the constant polynomial $P(x) = m$. Now Alice will transmit both $P(1)$ and $P(2)$ to Bob (to compensate for possible erasure). So her transmission will be of the form (m,m),i.e. a symbol drawn from $\texttt{range}(N)$ that is repeated twice. Clearly, even if one packet is erased, Bob can retrieve Alice's message from the other. So N being composite has no adverse effect when $n = 1$.

   (b) Now suppose $n = 2$. That is, Alice now wants to send 2 packets to Bob (plus one redundant packet to compensate for erasure). Now, would there be any problems because $N = 12$?

   Answer: If Alice wants to send 2 packets to Bob, then her "message" consists of a pair of integers (m_1, m_2), each drawn from $\texttt{range}(N)$. Now Alice will try to find a polynomial $P(x)$ of degree at most 1 that evaluates to m_1 at $x = 1$ and m_2 at $x = 2$. Let's say this polynomial is $P(x) = ax + b$ (the highest power of x is 1 because the degree of the polynomial is at most 1). Then we have:

$$\begin{aligned} a + b &= m_1 \pmod{N}, \text{ and} \\ 2a + b &= m_2 \pmod{N} \end{aligned}$$

   The above equations always have a solution, and what is more, this solution is unique and is given by $a = (m_2 - m_1) \bmod N$, and $b = (2m_1 - m_2) \bmod N$, regardless of whether N is prime or composite. So there is one, and only one, polynomial $P(x)$ that Alice can construct, before she transmits $(P(1), P(2), P(3))$.

   However, that does *not* mean that this is the only polynomial that Bob will come up with. To see why, consider this simple example. Let us say Alice wants to send the message $(3,5)$. Her polynomial, therefore, will be $P(x) = 2x + 1$. So far, this is unique and well-defined. The integers Alice transmits will then be $(3,5,7)$. Now, what if the channel erases the second packet, which leaves Bob with $(3,\square,7)$. Bob dutifully attempts to find a polynomial $Q(x) = cx + d$ such that $Q(1) = 3$ and $Q(3) = 7$. This gives him the following 2 equations:

$$\begin{aligned} c+d &= 3 \pmod{12}, \text{ and} \\ 3c+d &= 7 \pmod{12} \end{aligned}$$

The problem is that the above equations don't have a unique solution. Indeed, there are 2 solutions that work, $(c,d) = (2,1)$ and $(c,d) = (8,7)$. Thus, as far as Bob can tell, Alice's polynomial might have been either $2x+1$ or $8x+7$. This means Alice's message could have been either $(3,5)$ or $(3,11)$; Bob can narrow it down to these two possibilities, but beyond that, he has no clue as to which of these is the actual message that Alice had intended.

Note: if Bob had tried Lagrange interpolation to find $Q(x)$, that would not have given him a solution either. Indeed, applying Lagrange's method, the "solution" obtained by Bob would have been:

$$\begin{aligned} Q(x) &= 3 \cdot \frac{x-3}{1-3} + 7 \cdot \frac{x-1}{3-1} \\ &= 3 \cdot \frac{x-3}{10} + 7 \cdot \frac{x-1}{2} \\ &= 3 \cdot (x-3) \cdot 10^{-1} + 7 \cdot (x-1) \cdot 2^{-1} \end{aligned}$$

But neither 10 nor 2 has an inverse in mod 12 arithmetic, so the above formula becomes meaningless.

The root cause of the above problems is that N is composite. If N had been prime, Bob's system of equations above would have had a unique solution. Alternatively, every integer in `range`(N) would have had an inverse mod N, and therefore Lagrange interpolation would have succeeded. So N being composite has serious adverse effects when $n = 2$.

(c) Now let $n = 3$ (3 packets plus one additional packet to compensate for erasure). Assume that Alice wants to encode messages into "systematic" codewords (with the first few evaluations of the polynomial being the message itself). Prove that Alice can no longer send arbitrary messages of her liking to Bob, by showing that it would be impossible for Alice to send the message $(11,6,2)$. Find 2 other examples of messages that Alice cannot send to Bob.

Answer: In the previous ($n = 2$) case, at least the polynomial $P(x)$ that Alice would need was unique and well-defined. All the problems that arose were at Bob's end, so to speak. Now, with $n = 3$, that is no longer the case (now there can be problems at Alice's end as well). Let's say Alice wants to send the message $(11,6,2)$. Thus, she has to find a polynomial $P(x) = ax^2+bx+c$, such that

$$\begin{aligned} a+b+c &= 11 \pmod{12}, \\ 4a+2b+c &= 6 \pmod{12}, \text{ and} \\ 9a+3b+c &= 2 \pmod{12} \end{aligned}$$

Eliminating c from the first two equations, and then from the last two equations, we obtain:

$$\begin{aligned} 3a+b &= 7 \pmod{12}, \text{ and} \\ 5a+b &= 8 \pmod{12} \end{aligned}$$

Subtracting the first equation above from the second, we obtain:

$$2a = 1 \pmod{12}$$

However, the equation above is impossible. No matter what the value assigned to a, $2a$ never leaves remainder 1 when divided by 12. This is because 2 has no multiplicative inverse in mod 12 arithmetic.

Thus, it is impossible for Alice to send the message $(11,6,2)$.

Note: As in part (b) above, if Alice had tried Lagrangian interpolation, that would have failed as well, giving rise to meaningless multiplicative inverses, just like those that came up in part (b) above.

Two other messages that Alice cannot possibly send are $(0,0,1)$ and $(0,1,1)$.