CS 70 Discrete Mathematics and Probability Theory
Spring 2020 HW 5

1 RSA with Three Primes

Show how you can modify the RSA encryption method to work with three primes instead of two primes (i.e. $N = pqr$ where p,q,r are all prime), and prove the scheme you come up with works in the sense that $D(E(x)) \equiv x \pmod N$.

Solution:

$N = pqr$ where p,q,r are all prime. Then, let e be co-prime with $(p-1)(q-1)(r-1)$. Give the public key: (N,e) and calculate $d = e^{-1} \bmod (p-1)(q-1)(r-1)$. People who wish to send me a secret, x, send $y = x^e \bmod N$. I decrypt an incoming message, y, by calculating $y^d \bmod N$.

Does this work? We need to prove that $x^{ed} - x \equiv 0 \pmod N$ and thus $x^{ed} \equiv x \pmod N$. To prove that $x^{ed} - x \equiv 0 \pmod N$, we factor out the x to get $x \cdot (x^{ed-1} - 1) = x \cdot (x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ because $ed \equiv 1 \pmod{(p-1)(q-1)(r-1)}$. As a reminder, we are considering the number: $x \cdot (x^{k(p-1)(q-1)(r-1)} - 1)$.

We now argue that this number must be divisible by p, q, and r. Thus it is divisible by N and $x^{ed} - x \equiv 0 \pmod N$.
To prove that it is divisible by p:

- If x is divisible by p, then the entire thing is divisible by p.
- If x is not divisible by p, then that means we can use FLT on the inside to show that $(x^{p-1})^{k(q-1)(r-1)} - 1 \equiv 1 - 1 \equiv 0 \pmod p$. Thus it is divisible by p.

The same reasoning shows that it is divisible by q and r.

Alternatively, we could also have used a CRT-based argument. If we look at $y_p = y \bmod p$, then this is the coordinate corresponding to the prime p from the CRT perspective. We know that the coordinates don't interact with each other for addition, multiplication, or exponentiation of numbers. Since e is coprime with $(p-1)(q-1)(r-1)$, it is also coprime with just $p-1$ individually. So, we can compute d_p the multiplicative inverse mod $(p-1)$ of e. How? We just compute via the EGCD a,b so that $1 = ae + b(p-1)$. Here, we can set d_p to be any positive natural number that is congruent to a mod $p-1$. (Just in case the a that EGCD gives is negative.) Now, $x_p = y_p^{d_p} \bmod p$ since if $x_p \equiv 0 \pmod p$, then zero to any power is zero and we recover it. If x was not a multiple of p, then x_p is not zero and $y_p = x_p^e \bmod p$ by CRT coordinates, and thus $y_p^{d_p} \bmod p = (x_p^e)^{d_p} \bmod p = x_p^{ed_p} \bmod p = x_p^{1+k(p-1)} \bmod p = x_p x_p^{k(p-1)} \bmod p = x_p(x_p^{(p-1)})^k \bmod p = x_p$ where the last line follows from FLT since $x_p^{p-1} \bmod p = 1$. This tells us that we can recover x_p from y_p. We can do the same by analogous reasoning for x_q, x_r as well by constructing analogous d_q and d_r. Once we have x_p, x_q, x_r, we can use the CRT to reconstruct x since p,q,r are pairwise coprime (since they are prime) and hence $x = x_p v_p + x_q v_q + x_r v_r \mod pqr$ where $v_p = qr * (qr)^{-1}$ and the multiplicative inverse is being calculated mod p, and similarly for $v_q = pr * (pr)^{-1}$ and $v_r = pq * (pq)^{-1}$. The subscript of the v tells what the relevant multiplicative inverse is calculated relative to.

2 Breaking RSA

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).

Solution:

Let $a = (p-1)(q-1)$. If Eve knows $a = (p-1)(q-1) = pq - (p+q) + 1$, then she knows

$$N - q - p + 1 = a,$$

$$pq = N.$$

We can write q as $N - p - a + 1$ and substitute into the second equation:

$$p(N - p - a + 1) = N.$$

Then we get the following quadratic function for p:

$$p^2 + (a - N - 1)p + N = 0.$$

We can easily solve this equations and obtain p and q. This is equivalent to factoring N.

3 Quantum Factoring

We're pretty sure that classical computers can't break RSA (because it is hard to factor large numbers on them), but we know that quantum computers theoretically could. In this question, we will prove a fact that is a key part of Shor's Algorithm, a quantum algorithm for factoring large numbers quickly[1].

Let $N = pq$ where p,q are primes throughout this question.

(a) Prove that, for all $a \in \mathbb{N}$, there are only four possible values for $gcd(a,N)$.

(b) Using part (a), prove that, if $r^2 \equiv 1 \mod N$ and $r \not\equiv \pm 1 \pmod N$ (i.e. r is a "nontrivial square root of 1" mod N), then $gcd(r-1,N)$ is one of the prime factors of N.
Hint: $r^2 = 1 \mod N$ *can be rewritten as* $r^2 - 1 = 0 \mod N$ *or* $(r+1)(r-1) = 0 \mod N$.

Solution:

[1] Read more at `https://en.wikipedia.org/wiki/Shors_algorithm`.

(a) N only has four divisors: 1, p, q, and N. $gcd(a,N)$ is a divisor of N, and can thus only take one of those four values.

(b) Since we are restricted to four possible values, this is conducive to a proof by cases. We only have to show that $gcd(r-1,N)$ is not 1 or N; $gcd(r-1,N)$ can only take one of the previous four values, and, if it is not 1 or N, then it must be one of the prime factors.

Case 1: Proving $gcd(r-1,N) \neq 1$:

Assume for the sake of contradiction that $gcd(r-1,N) = 1$. By the extended GCD algorithm, $gcd(r-1,N) = a(r-1)+bN$. Since $bN \equiv 0 \pmod N$, then:

$$\begin{aligned} a(r-1) &\equiv 1 \pmod N \\ a(r^2-1) &\equiv r+1 \pmod N \end{aligned} \tag{1}$$

where the second line comes from multiplying both sides by $(r+1)$. We know the left side is 0 since $r^2-1 \equiv 0 \pmod N$, but this implies $0 \equiv r+1 \mod N$, or $r \equiv -1 \mod N$. Since we assumed that r is a nontrivial square root of 1, this is a contradiction.

Case 2: Proving $gcd(r-1,N) \neq N$:

If $gcd(r-1,N) = N$, then $N|r-1$ and therefore $r-1 \equiv 0 \pmod N$. Therefore $r = 1 \pmod N$. However, we assumed that r is a nontrivial square root of 1, so this is a contradiction.

Since $gcd(r-1,N) \neq 1$ and $gcd(r-1,N) \neq N$, $gcd(r-1,N)$ must be one of the prime factors of N.

4 Recursively Defined Polynomials

Let's define two polynomials $f_1(x) = x-2$, $f_2(x) = x^2+3$, and for each natural number $n \geq 2$, recursively define $f_n(x) = xf_{n-1}(x) - f_{n-2}(x)$.

(a) Compute $f_3(x)$ and $f_4(x)$.

(b) First, show that $f_n(x)$ has degree n. Based on this fact, what is the largest number of roots that $f_n(x)$ can have over $\mathbb{R}$? What is the smallest number of roots it can have?

(c) Prove that $f_n(2) = 0 \mod 7$ for every n.

Solution:

(a) Directly from the definition of $f_n(x)$ in the problem,

$$\begin{aligned} f_3(x) &= xf_2(x) - f_1(x) \\ &= x(x^2+3) - (x-2) \\ &= x^3 + 2x + 2 \end{aligned}$$

and

$$\begin{aligned} f_4(x) &= xf_3(x) - f_2(x) \\ &= x(x^3+2x+2) - (x^2+3) \\ &= x^4+x^2+2x-3. \end{aligned}$$

(b) Recall that the *degree* of a polynomial is the highest exponent that appears in it, and that a polynomial of degree d has at most d roots over any field. Over $\mathbb{R}$, a polynomial of even degree need not have any roots (consider $x^{2d}+a^2$), but a polynomial of odd degree must have at least one: if $f(x) = x^{2d+1} + f_{2d}x^{2d} + \cdots + f_0x_0$ then $f(x)$ goes to positive infinity as x gets very large and positive, to negative infinity as x gets very large and negative, and we can use the intermediate value theorem to prove that there is a root somewhere in between. So all we need to do is compute the degree of $f_n(x)$.

In our case, the degree of $f_1(x)$ is one, so it has exactly one root (a degree one polynomial over any field has exactly one root). The degree of $f_2(x)$ is two, so it could in principle have up to two roots, but because $f_2(x) = x^2+3$ and $3 > 0$, it has none. For $n > 1$, let's prove by (strong) induction that $f_n(x)$ has degree n, and that the coefficient of the x^n term is 1. We've already done the base case, so assume that $f_k(x)$ and $f_{k-1}(x)$ have degree k and $k-1$ respectively. Then by definition

$$\begin{aligned} f_{k+1}(x) &= xf_k(x) - f_{k-1}(x) \\ &= x\left(x^k + a_{k-1}x^{k-1} + \cdots + a_1x + a_0\right) - \left(x^{k-1} + b_{k-2}x^{k-2} + \cdots + b_1x + b_0\right) && \text{Ind. Assump.} \\ &= x^{k+1} + a_{k-1}x^k + (a_{k-2}-1)x^{k-1} + (a_{k-3}-b_{k-2})x^{k-2} + \cdots + (a_0-b_1)x - b_0, \end{aligned}$$

and the last line clearly has degree $k+1$.

Since we've proved that $f_n(x)$ is a degree-n polynomial, we know it can have at most n roots, and has at least zero if n is even, and at least one if n is odd.

(c) Let's prove this by strong induction as well. We'll need two base cases, since each $f_k(x)$ is defined in terms of the two previous polynomials in the sequence: these base cases are

$$\begin{aligned} f_0(2) &= 2-2 = 0 \\ f_1(2) &= 2^2+3 = 7 = 0 \pmod 7. \end{aligned}$$

For our inductive assumption, assume that $f_k(2) = 0 \pmod 7$ and $f_{k-1}(2) = 0 \pmod 7$. In other words, there exist $p, q \in \mathbb{Z}$ so that $f_k(2) = 7p$ and $f_{k+1}(2) = 7q$. Using the definition of f_{k+1}, we can write

$$f_{k+1}(2) = 2f_k(2) - f_{k-1}(2) = 2(7p) - (7q) = 7(2p-q) = 0 \pmod 7.$$

5 Equivalent Polynomials

This problem is about polynomials with coefficients in GF(q) for some prime $q \in \mathbb{N}$. We say that two such polynomials f and g are *equivalent* if $f(x) = g(x)$ for every $x \in \mathrm{GF}(q)$.

(a) Use Fermat's Little Theorem to find a polynomial equivalent to $f(x) = x^5$ over GF(5); then find one equivalent to $g(x) = 1 + 3x^{11} + 7x^{13}$ over GF(11).

(b) Prove that whenever $f(x)$ has degree $\geq q$, it is equivalent to some polynomial $\tilde{f}(x)$ with degree $< q$.

Solution:

(a) Fermat's Little Theorem says that for any nonzero integer a and any prime number q, $a^{q-1} \equiv 1 \mod q$. We're allowed to multiply through by a, so the theorem is equivalent to saying that $a^q \equiv a \mod q$; note that this is true even when $a = 0$, since in that case we just have $0^q \equiv 0 \mod q$. The problem asks for a polynomial $\tilde{f}(x)$, different from $f(x)$, with the property that $\tilde{f}(a) \equiv a^5 \mod 5$ for any integer a. Directly using the theorem, $\tilde{f}(x) = x$ will work. We can do something similar with $g(x) = 1 + 3x^{11} + 7x^{13}$ modulo 11: set $\tilde{g}(x) = 1 + 3x + 7x^3$.

(b) One proof uses Fermat's Little Theorem. As a warm-up, let $d \geq q$; we'll find a polynomial equivalent to x^d. For any integer, we know

$$\begin{aligned} a^d &= a^{d-q}a^q \\ &\equiv a^{d-q}a \mod q \\ &\equiv a^{d-q+1} \mod q. \end{aligned}$$

In other words x^d is equivalent to the polynomial $x^{d-(q-1)}$. If $d-(q-1) \geq q$, we can show in the same way that x^d is equivalent to $x^{d-2(q-1)}$. Since we subtract $q-1$ every time, the sequene $d, d-(q-1), d-2(q-1), \ldots$ must eventually be smaller than q. Now if $f(x)$ is any polynomial with degree $\geq q$, we can apply this same trick to every x^k that appears for which $k \geq q$.

Another proof uses Lagrange interpolation. Let $f(x)$ have degree $\geq q$. By Lagrange interpolation, there is a unique polynomial $\tilde{f}(x)$ of degree at most $q-1$ passing through the points $(0, f(0)), (1, f(1)), (2, f(2)), \ldots, (q-1, f(q-1))$, and we designed it exactly so that it would be equivalent to $f(x)$.

6 Repeated Roots

Let $p(x) = a_k x^k + \cdots + a_0$ be a polynomial in the variable x, where k is a positive integer and the coefficients $a_0, \ldots, a_k$ are from some field F (here, F can be $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, or GF(p) for some prime p). Let's define this polynomial's **derivative** to be the polynomial $p'(x) := k a_k x^{k-1} + \cdots + a_1 =$

$\sum_{j=1}^{k} ja_j x^{j-1}$. We say that α is a **repeated root of** p if $p(x)$ can be factored as $(x-\alpha)^2 q(x)$ for some polynomial q. Show that α is a repeated root of p if and only if $p(\alpha) = p'(\alpha) = 0$.

[Note: You may be familiar with the derivatives of polynomials from studying calculus, but we are not using any calculus here, because it does not really make sense to perform calculus on finite fields! Think of the polynomial's derivative as a formal definition, i.e., in this context, it has nothing to do with rate of change, etc. In particular, you should not use any calculus rules such as the product rule without proof.]

Solution:

Recall Claim 1 from Note 8, Page 4, which says that α is a root of p if and only if $p(x) = (x-\alpha)r(x)$ for some polynomial r.

Assume first that α is a repeated root of p; we'll prove that it is a root of p' as well. Applying Claim 1 twice gives us $p(x) = (x-\alpha)^2 q(x)$. Note that if we write $q(x) = \sum_{j=0}^{m} b_j x^j$,

$$\begin{aligned}
p(x) &= (x^2 - 2\alpha x + \alpha^2) \sum_{j=0}^{m} b_j x^j = \sum_{j=0}^{m} b_j (x^{j+2} - 2\alpha x^{j+1} + \alpha^2 x^j), \\
p'(x) &= \sum_{j=1}^{m} b_j \big((j+2)x^{j+1} - 2\alpha(j+1)x^j + \alpha^2 j x^{j-1}\big) + b_0(2x - 2\alpha), \\
p'(\alpha) &= \sum_{j=1}^{m} b_j \big((j+2)\alpha^{j+1} - 2(j+1)\alpha^{j+1} + j\alpha^{j+1}\big) + b_0(2\alpha - 2\alpha) = 0,
\end{aligned}$$

as we had hoped. Conversely, assume that $p(\alpha) = p'(\alpha) = 0$. We'll prove that α is a repeated root. Using Claim 1 again, we can write $p(x) = (x-\alpha)r(x)$, and it suffices to prove that α is a root of r, since (applying Claim 1 a final time) this will mean $r(x) = (x-\alpha)\tilde{q}(x)$, and thus $p(x) = (x-\alpha)^2 \tilde{q}(x)$. Writing $r(x) = \sum_{j=0}^{\ell} c_j x^j$, we have

$$\begin{aligned}
p(x) &= (x-\alpha) \sum_{j=0}^{\ell} c_j x^j = \sum_{j=0}^{\ell} c_j (x^{j+1} - \alpha x^j), \\
p'(x) &= \sum_{j=1}^{\ell} c_j \big((j+1)x^j - \alpha j x^{j-1}\big) + c_0, \\
p'(\alpha) &= \sum_{j=1}^{\ell} c_j \big((j+1)\alpha^j - j\alpha^j\big) + c_0 = \sum_{j=1}^{\ell} c_j \alpha^j + c_0 = r(\alpha) = 0,
\end{aligned}$$

and so α is a root of r.

7 The CRT and Lagrange Interpolation

Let $n_1, \ldots n_k$ be pairwise coprime, i.e. n_i and n_j are coprime for all $i \neq j$. The Chinese Remainder Theorem (CRT) tells us that there exist solutions to the following system of congruences:

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} && (1)\\ x &\equiv a_2 \pmod{n_2} && (2)\\ &\vdots && (\vdots)\\ x &\equiv a_k \pmod{n_k} && (k) \end{aligned}$$

and all solutions are equivalent $\pmod{n_1 n_2 \cdots n_k}$. For this problem, parts (a)-(c) will walk us through a proof of the Chinese Remainder Theorem. We will then use the CRT to revisit Lagrange interpolation.

(a) We start by proving the $k = 2$ case: Prove that we can always find an integer x_1 that solves (1) and (2) with $a_1 = 1, a_2 = 0$. Similarly, prove that we can always find an integer x_2 that solves (1) and (2) with $a_1 = 0, a_2 = 1$.

(b) Use part (a) to prove that we can always find at least one solution to (1) and (2) for any a_1, a_2. Furthermore, prove that all possible solutions are equivalent $\pmod{n_1 n_2}$.

(c) Now we can tackle the case of arbitrary k: Use part (b) to prove that there exists a solution x to (1)-(k) and that this solution is unique $\pmod{n_1 n_2 \cdots n_k}$.

(d) For two polynomials $p(x)$ and $q(x)$, mimic the definition of $a \bmod b$ for integers to define $p(x) \bmod q(x)$. Use your definition to find $p(x) \bmod (x-1)$.

(e) Define the polynomials $x - a$ and $x - b$ to be coprime if they have no common divisor of degree 1. Assuming that the CRT still holds when replacing x, a_i and n_i with polynomials (using the definition of coprime polynomials just given), show that the system of congruences

$$\begin{aligned} p(x) &\equiv y_1 \pmod{(x - x_1)} && (1')\\ p(x) &\equiv y_2 \pmod{(x - x_2)} && (2')\\ &\vdots && (\vdots)\\ p(x) &\equiv y_k \pmod{(x - x_k)} && (k') \end{aligned}$$

has a unique solution $\pmod{(x - x_1) \cdots (x - x_k)}$ whenever the x_i are pairwise distinct. What is the connection to Lagrange interpolation?

Hint: To show that a unique solution exists, you may use the fact that the CRT has a unique solution when certain properies are satisfied.

Solution:

(a) Since $\gcd(n_1, n_2) = 1$, there exist integers k_1, k_2 such that $1 = k_1 n_1 + k_2 n_2$. Setting $x_1 = k_2 n_2 = 1 - k_1 n_1$ and $x_2 = k_1 n_1 = 1 - k_2 n_2$ we obtain the two desired solutions.

(b) Using the x_1 and x_2 we found in Part (a), we show that $a_1x_1 + a_2x_2 \pmod{n_1n_2}$ is a solution to the desired equivalences:

$$\begin{aligned} a_1x_1 + a_2x_2 &\equiv a_1 \cdot 1 + a_2 \cdot 0 \equiv a_1 \pmod{n_1} \\ a_1x_1 + a_2x_2 &\equiv a_1 \cdot 0 + a_2 \cdot 1 \equiv a_2 \pmod{n_2}. \end{aligned}$$

Such result is also unique. Say that we have two difference solutions $x = c$ and $x = c'$, which both satisfy $x \equiv a_1 \pmod{n_1}$ and $x \equiv a_2 \pmod{n_2}$. This would give us $c \equiv c' \pmod{n_1}$ and $c \equiv c' \pmod{n_2}$, which suggests that $(c - c')$ is divisible by n_1 and n_2. Since n_1 and n_2 are coprime, $gcd(n_1, n_2) = 1$, $(c - c')$ is divisible by n_1n_2. Writing it in modular form gives us $c \equiv c' \pmod{n_1n_2}$. Therefore, all the result is unique with respect to $\pmod{n_1n_2}$

(c) We use induction on k. Part (b) handles the base case, $k = 2$. For the inductive hypothesis, assume for $k \leq l$, the system (1)-(k) has a unique solution $a \pmod{n_1 \cdots n_k}$. Now consider $k = l + 1$, so we add the equation $x \equiv a_{l+1} \pmod{n_{l+1}}$ to our system, resulting in

$$\begin{aligned} x &\equiv a \pmod{n_1 \cdots n_l} \\ x &\equiv a_{l+1} \pmod{n_{l+1}}. \end{aligned}$$

Since the n_i are pairwise coprime, $n_1n_2 \cdots n_l$ and n_{l+1} are coprime. Part (b) tells us that there exists a unique solution $a' \pmod{n_1 \cdots n_l n_{l+1}}$. We conclude that a' is the unique solution to (1)-$(l+1)$, when taken $\pmod{n_1n_2 \cdots n_l n_{l+1}}$.

(d) a mod b is defined as the remainder after division by b. But we know how to divide polynomials and compute remainders too! In particular, we know that we can write $p(x) = q'(x)q(x) + r(x)$ where $\deg r < \deg q$. So we define $p(x) \bmod q(x) = r(x)$.

To compute $p(x) \bmod (x - 1)$ then, we write $p(x) = (x - 1)q'(x) + r(x)$. We know that $\deg r < \deg(x - 1) = 1$ and so r must be a constant. Which constant is it? Plugging in $x = 1$ gives $p(1) = r(1)$ and so $r(x) = p(1)$ for all x.

(e) We only need to check that $q_i(x) = (x - x_i)$ and $q_j(x) = (x - x_j)$ are coprime whenever $x_i \neq x_j$; that is, that they don't share a common divisor of degree 1. If $d_i(x) = a_ix + b_i$ is a divisor of $q_i(x)$, then $q_i(x) = q'(x)(a_ix + b_i)$ for some polynomial $q'(x)$. But since $q_i(x)$ is of degree 1, $q'(x)$ must be of degree 0 and hence a constant, so $d_i(x)$ must be a constant multiple of $q_i(x)$. Similarly, any degree 1 divisor d_j of $q_j(x)$ must be a constant multiple of $q_j(x)$, and if $x_i \neq x_j$, then none of these multiples overlap, so $q_i(x)$ and $q_j(x)$ are coprime.

From our result in part (d), the congruences $(1')$-(k') assert that we are looking for a polynomial $p(x)$ such that $p(x_i) = y_i$. The CRT then establishes the existence of $p(x)$, and that it is unique modulo a degree k polynomial. That is, $p(x)$ is unique if its degree is at most $k - 1$. Lagrange interpolation finds $p(x)$.

8 GCD of Polynomials

Let $A(x)$ and $B(x)$ be polynomials (with coefficients in $\mathbb{R}$). We say that $\gcd(A(x), B(x)) = D(x)$ if $D(x)$ divides $A(x)$ and $B(x)$, and if every polynomial $C(x)$ that divides both $A(x)$ and $B(x)$ also divides $D(x)$. For example, $\gcd((x-1)(x+1),(x-1)(x+2)) = x-1$. Notice this is the exact same as the normal definition of GCD, just extended to polynomials.

Incidentally, $\gcd(A(x),B(x))$ is the highest degree polynomial that divides both $A(x)$ and $B(x)$. In the subproblems below, you may assume you already have a subroutine `divide`$(P(x),S(x))$ for dividing two polynomials, which returns a tuple $(Q(x),R(x))$ of the quotient and the remainder, respectively, of dividing $P(x)$ by $S(x)$.

(a) Write a recursive program to compute `gcd`$(A(x),B(x))$.

(b) Write a recursive program to compute `extended-gcd`$(A(x),B(x))$.

Solution:

(a) Specifically, we wish to find a gcd of two polynomials $A(x)$ and $B(x)$, assuming that $\deg A(x) \geq \deg B(x) > 0$. Here, $\deg A(x)$ denotes the degree of $A(x)$.

We can find two polynomials $Q_0(x)$ and $R_0(x)$ by polynomial long division which satisfy

$$A(x) = B(x)Q_0(x) + R_0(x), \qquad\qquad 0 \leq \deg R_0(x) < \deg B(x).$$

Notice that a polynomial $C(x)$ divides $A(x)$ and $B(x)$ if and only if it divides $B(x)$ and $R_0(x)$.

[*Proof*: Forward Direction: $C(x)$ divides $A(x)$ and $B(x)$, there exists $S(x)$ and $S'(x)$ s.t. $A(x) = C(x)S(x)$ and $B(x) = C(x)S'(x)$, so $R_0(x) = A(x) - B(x)Q_0(x) = C(x)(S(x) - S'(x)Q_0(x))$, therefore $C(x)$ divides $R_0(x)$ or $R_0(x) = 0$.

Backward Direction: $C(x)$ divides $B(x)$ and $R(x)$, there exists $S(x)$ and $S'(x)$ s.t. $B(x) = C(x)S(x)$ and $R(x) = C(x)S'(x)$, so $A(x) = B(x)Q_0(x) + R(x) = C(x)(S(x)Q_0(x) + S'(x))$, therefore $C(x)$ divides $A(x)$]

We deduce that

$$\gcd(A(x),B(x)) = \gcd(B(x),R(x))$$

and set $A_1(x) = B_1(x), B_1(x) = R_0(x)$; we then repeat to get new polynomials $Q_1(x)$, $R_1(x)$, $A_2(x)$, $B_2(x)$, and so on. The degrees of the polynomials keep getting smaller and will eventually reach a point at which $B_N(x) = 0$; and we will have found our GCD:

$$\gcd(A(x),B(x)) = \gcd(A_1(x),B_1(x)) = \cdots = \gcd(A_N(x),0) = A_N(x)$$

Here, we have the function that can perform the polynomial long division on $A(x)$ and $B(x)$ and return both the quotient $Q(x)$ and the remainder $R(x)$, i.e. [$Q(x)$, $R(x)$] = div($A(x)$,$B(x)$). The algorithm can be extended from the original integer-based GCD as follows:

```
function gcd(A(x), B(x)):
  if B(x) = 0:
    return A(x)
  else if deg A(x) < deg B(x):
    return gcd(B(x), A(x))
```

```
            else:
              (Q(x), R(x)) = div(A(x), B(x))
              return gcd(B(x), R(x))
```

(b) We will return a triple of polynomials $(d(x), g(x), h(x))$ such that $d(x) = \gcd(A(x), B(x))$ and $d(x) = g(x) \cdot A(x) + h(x) \cdot B(x)$.

```
function extended-gcd(A(x), B(x)):
  if B(x) = 0:
    return (A(x), 1, 0)
  else if deg A(x) < deg B(x):
    (d(x), g(x), h(x)) := extended-gcd(B(x), A(x))
    return (d(x), h(x), g(x))
  else:
    (Q(x), R(x)) = div(A(x), B(x))
    (d(x), g(x), h(x)) := extended-gcd(B(x), R(x))
    return (d(x), h(x), g(x) - Q(x) * h(x))
```

9 Trust No One

Gandalf has assembled a fellowship of eight people to transport the One Ring to the fires of Mount Doom: four hobbits, two men, one elf, and one dwarf. The ring has great power that may be of use to the fellowship during their long and dangerous journey. Unfortunately, the use of its immense power will eventually corrupt the user, so it must not be used except in the most dire of circumstances. To safeguard against this possibility, Gandalf wishes to keep the instructions a secret from members of the fellowship. The secret must only be revealed if enough members of the fellowship are present and agree to use it.

Requiring all eight members to agree is certainly a sufficient condition to know the instructions, but it seems excessive. However, we also know that the separate races (hobbits, men, elf, and dwarf) do not completely trust each other so instead we decide to require members from at least two races in order to use the ring. In particular, we will require a unanimous decision by all members of one race in addition to at least one member of a different race. That is, if only the four hobbits want to use the ring, then they alone should not have sufficient information to figure out the instructions. Same goes for the two men, the elf, and the dwarf.

More explicitly, some examples: only four hobbits agreeing to use the ring is not enough to know the instructions. Only two men agreeing is not enough. Only the elf agreeing is not enough. Only the dwarf agreeing is not enough. All four hobbits and a man agreeing is enough. Both men and a dwarf agreeing is enough. Both the elf and the dwarf agreeing is enough.

Gandalf has hired your services to help him come up with a secret sharing scheme that accomplishes this task, summarized by the following points:

- There is a party of four hobbits, two men, an elf, and a dwarf.
- There is a secret message that needs to be known if enough members of the party agree.
- The message must remain unknown to everyone (except Gandalf) if not enough members of the party agree.
- If only the members of one race agree, the message remains a secret.
- If all the members of one race agree plus at least one additional person, the message can be determined.

Solution:

There will be two parts to this secret: a unanimity secret U and a multi-race secret M. U ensures that at least all members of one races are in agreement while M ensures that members of at least two races are in agreement. We will discuss these two in order below. Once both U and M are recovered, they can then be combined to reveal the original secret: each will be a point of the degree-1 polynomial $R(x)$ whose y-intercept contains the secret of the ring.

The *unanimity secret* involves creating a separate secret for each race. We will require all members of that race to join forces in order to reveal the secret. For example, the hobbits will each have distinct points of a degree-3 polynomial and the men will each have distinct points of a degree-1 polynomial. When all members of a race come together, they will reveal U (encoded, for example, as the y-intercept of each of these polynomials). Note that the elf and the dwarf each know U already since they are the only members of their race.

The *multi-race secret* involves creating a degree-1 polynomial $P_m(x)$ and giving one point to all members of each race. For example, the hobbits may each get $P_m(1)$ while the elf gets $P_m(2)$ and the men each get $P_m(3)$. In this way if members of any two races are in agreement, they can reveal M (encoded, for example, as the y-intercept of $P_m(x)$).

Once U and M are each known, they can be *combined* to determine the final secret. U and M allow us to uniquely determine $R(x)$ and thus $R(0)$, the secret of the ring.

This scheme is an example of hierarchical secret sharing. Let's work out a specific example.

Example: Suppose the secret is $s = 4$, $M = 3$, and $U = 2$. From now on, we can work in GF(5) since $s < 5$ and $n < 5$ (n is the number of people who have pieces of the secret).

First we need to create a degree-1 polynomial $R(x)$ such that $R(0) = s = 4$, $R(1) = M = 3$, and $R(2) = U = 2$. By inspection, $R(x) = 4x + 4$ has these properties (e.g. $R(1) = 4 \cdot 1 + 4 = 8 \equiv 3$).

Now we can create the multi-race secret M. We choose degree-1 polynomial $P_m(x) = x + 3$ and tell each hobbit $P_m(1) = 4$, the elf $P_m(2) = 5 \equiv 0$, each of the men $P_m(3) = 6 \equiv 1$, and the dwarf $P_m(4) = 7 \equiv 2$. Now any two members of distinct races can determine $P_m(x)$ and thus $P_m(0)$ by interpolating their two values.

When creating the unanimity secret U, we first note that each of the dwarf and the elf will be told U directly since they are the only members of their respective races. On the other hand, the men will each have a point on the degree-1 polynomial $P_{men}(x)$. Suppose $P_{men}(x) = 2x + 2$. Then the first man receives $P_{men}(1) = 4$ and the second receives $P_{men}(2) = 4 + 2 = 6 \equiv 1$. When they interpolate using these values, they will discover the original polynomial and therefore $P_{men}(0) = U = 2$. The hobbits will have a similar secret but with a degree-3 polynomial (e.g. $P_{hobbit}(x) = 4x^3 + x^2 + 2$).

Now suppose that two men and one hobbit come together. The two men work together to determine U as described above. Together the three of them also know $P_m(3) = 6$ and $P_m(1) = 4$, from which they can find $P_m(x)$ and thus $P_m(0) = M = 3$. Now that they have U and M, they can interpolate to find $R(x)$ and thus $R(0) = s = 4$.

10 Make Your Own Question

You must make your own question on this week's material and solve it.

11 Homework Process and Study Group

You must describe your homework process and study group in order to receive credit for this question.