Solutions to In-Class Problems Week 7, Wed.

Problem 1. Let's try out RSA! There is a complete description of the algorithm at the bottom of the page. You'll probably need extra paper. *Check your work carefully!*

(a) As a team, go through the **beforehand** steps.

- Choose primes p and q to be relatively small, say in the range 10-40. In practice, p and q might contain several hundred digits, but small numbers are easier to handle with pencil and paper.
- Try $e = 3, 5, 7, \ldots$ until you find something that works. Use Euclid's algorithm to compute the gcd.
- Find d using the Pulverizer (see appendix for a reminder on how the Pulverizer works).

When you're done, put your public key on the board. This lets another team send you a message.

(b) Now send an encrypted message to another team using their public key. Select your message m from the codebook below:

- 2 = Greetings and salutations!
- 3 = Yo, wassup?
- 4 = You guys are slow!
- 5 = All your base are belong to us.
- 6 = Someone on *our* team thinks someone on *your* team is kinda cute.
- 7 = You *are* the weakest link. Goodbye.

(c) Decrypt the message sent to you and verify that you received what the other team sent!

(d) Explain how you could read messages encrypted with RSA if you could quickly factor large numbers.

Solution. Suppose you see a public key (e, n). If you can factor n to obtain p and q, then you can compute d using the Pulverizer. This gives you the secret key (d, n), and so you can decode messages as well as the intended recipient. ■

RSA Public Key Encryption

Beforehand The receiver creates a public key and a secret key as follows.

1. Generate two distinct primes, p and q.
2. Let $n = pq$.
3. Select an integer e such that $\gcd(e, (p-1)(q-1)) = 1$.
   The *public key* is the pair (e, n). This should be distributed widely.
4. Compute d such that $de \equiv 1 \pmod{(p-1)(q-1)}$.
   The *secret key* is the pair (d, n). This should be kept hidden!

Encoding The sender encrypts message m to produce m' using the public key:

$$m' = m^e \text{ rem } n.$$

Decoding The receiver decrypts message m' back to message m using the secret key:

$$m = (m')^d \text{ rem } n.$$

Problem 2. A critical question is whether decrypting an encrypted message always gives back the original message! Mathematically, this amounts to asking whether:

$$m^{de} \equiv m \pmod{pq}.$$

Note that the procedure ensures that $de = 1 + k(p-1)(q-1)$ for some integer k.

(a) Use Euler's Theorem to prove that $m^{de} \equiv m \pmod{pq}$ for all messages m relatively prime to pq. (Euler's Theorem says that if k is relatively prime to n then $k^{\phi(n)} \equiv 1 \pmod{n}$.) In practice, is m likely to be relatively prime to pq or not?

Solution.

$$\begin{aligned} m^{de} &\equiv m^{1+k\phi(pq)} \pmod{pq} \\ &\equiv m \cdot (m^{\phi(pq)})^k \pmod{pq} \\ &\equiv m \cdot 1^k \pmod{pq} \end{aligned}$$

The first step uses the fact that $\phi(pq) = (p-1)(q-1)$, the second uses exponent laws, and third uses Euler's Theorem. If p and q are hundred-digit primes, m is very likely to be relatively prime to both p and q. ■

(b) This congruence actually holds for all messages m. First, use Fermat's theorem to prove that $m \equiv m^{de} \pmod{p}$ for all m. (Fermat's Theorem says that $a^{p-1} \equiv 1 \pmod{p}$ if p is a prime that does not divide a.)

Solution. If m is a multiple of p, then the claim holds because both sides are congruent to 0 mod p. Otherwise, suppose that m is not a multiple of p. Then:

$$\begin{aligned} m^{1+k(p-1)(q-1)} &\equiv m \cdot (m^{p-1})^{k(q-1)} \pmod{p} \\ &\equiv m \cdot 1^{k(q-1)} \pmod{p} \\ &\equiv m \pmod{p} \end{aligned}$$

The second step uses Fermat's theorem, which says that $m^{p-1} \equiv 1 \pmod{p}$ provided m is not a multiple of p. ■

(c) By the same argument, you can equally well show that $m \equiv m^{ed} \pmod{q}$. Show that these two facts together imply that $m \equiv m^{ed} \pmod{pq}$ for all m.

Solution. We know that:

$$\begin{aligned} p &\mid (m - m^{ed}), \\ q &\mid (m - m^{ed}). \end{aligned}$$

Thus, both p and q appear in the prime factorization of $m - m^{ed}$. Therefore, $pq \mid (m - m^{ed})$, and so:

$$m \equiv m^{ed} \pmod{pq}.$$

■

1 Appendix: The Pulverizer

Euclid's algorithm for finding the GCD of two numbers relies on repeated application of the equation:

$$\gcd(a, b) = \gcd(b, a \text{ rem } b)$$

For example, we can compute the GCD of 259 and 70 as follows:

$$\begin{aligned} \gcd(259, 70) &= \gcd(70, 49) && \text{since } 259 \text{ rem } 70 = 49 \\ &= \gcd(49, 21) && \text{since } 70 \text{ rem } 49 = 21 \\ &= \gcd(21, 7) && \text{since } 49 \text{ rem } 21 = 7 \\ &= \gcd(7, 0) && \text{since } 21 \text{ rem } 7 = 0 \\ &= 7. \end{aligned}$$

The Pulverizer goes through the same steps, but requires some extra bookkeeping along the way: as we compute $\gcd(a, b)$, we keep track of how to write each of the remainders (49, 21, and 7, in the example) as a linear combination of a and b (this is worthwhile, because our objective is to write the last nonzero remainder, which is the GCD, as such a linear combination). For our example, here is this extra bookkeeping:

x	y	$(x \text{ rem } y)$	$=$	$x - q \cdot y$
259	70	49	$=$	$259 - 3 \cdot 70$
70	49	21	$=$	$70 - 1 \cdot 49$
			$=$	$70 - 1 \cdot (259 - 3 \cdot 70)$
			$=$	$-1 \cdot 259 + 4 \cdot 70$
49	21	7	$=$	$49 - 2 \cdot 21$
			$=$	$(259 - 3 \cdot 70) - 2 \cdot (-1 \cdot 259 + 4 \cdot 70)$
			$=$	$\boxed{3 \cdot 259 - 11 \cdot 70}$
21	7	0		

We began by initializing two variables, $x = a$ and $y = b$. In the first two columns above, we carried out Euclid's algorithm. At each step, we computed x rem y, which can be written in the form $x - q \cdot y$. (Remember that the Division Algorithm says $x = q \cdot y + r$, where r is the remainder. We get $r = x - q \cdot y$ by rearranging terms.) Then we replaced x and y in this equation with equivalent linear combinations of a and b, which we already had computed. After simplifying, we were left with a linear combination of a and b that was equal to the remainder as desired. The final solution is boxed.