Solutions to In-Class Problems Week 8, Fri.

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 —or Euler’s Theorem).

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!

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 \( m \), where \( 0 \leq m < n \), to produce \( m' \) using the public key:

\[
m' = \text{rem}(m^e, n).
\]

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

\[
m = \text{rem}((m')^d, n).
\]

Problem 2.
A critical fact about RSA is, of course, that decrypting an encrypted message always gives back the original message! That is, that \( \text{rem}((m^e)^d, pq) = m \). This will follow from something slightly more general:

Lemma 2.1. Let \( n \) be a product of distinct primes and \( a \equiv 1 \pmod{\phi(n)} \) for some nonnegative integer, \( a \). Then

\[
m^a \equiv m \pmod{n}.
\]

(a) Explain why Lemma 2.1 implies that \( k^5 \) and \( k^5 \) have the same last digit. For example:

\[
2^5 = 32 \quad 79^5 = 3077056392
\]

Hint: What is \( \phi(10) \)?

Solution. Two nonnegative integers have the same last digit iff they are \( \equiv \pmod{10} \). Now \( \phi(10) = \phi(2)\phi(5) = 4 \) and \( 5 \equiv 1 \pmod{4} \), so by Lemma 2.1,

\[
k^5 \equiv k \pmod{10}.
\]

(b) Explain why Lemma 2.1 implies that the original message, \( m \), equals \( \text{rem}((m^e)^d, pq) \).

Solution. To apply Lemma 2.1 to RSA, note that the first condition of the Lemma is that \( n \) be a product of primes. In RSA, \( n = pq \) so this condition holds.
For \( n = pq \), the Euler function equations (see the Appendix) imply that \( \phi(n) = (p - 1)(q - 1) \). So when \( d \) and \( e \) are chosen according to RSA, \( de \equiv 1 \pmod{\phi(n)} \). So \( a := de \) satisfies the second condition of the Lemma.

Now, from equation (1) with \( n = pq \) and \( a = de \), we have

\[
(m^e)^d = m^{de} \equiv m \pmod{pq}.
\]

Hence,

\[
\text{rem}((m^e)^d, pq) = \text{rem}(m, pq),
\]

but \( \text{rem}(m, pq) = m \), since \( 0 \leq m < pq \).

\( \Box \)

(c) Prove that if \( p \) is prime, then

\[
m^a \equiv m \pmod{p}
\]

for all nonnegative integers \( a \equiv 1 \pmod{p - 1} \).

Solution. If \( p \mid m \), then equation (2) holds since both sides of the congruence are \( \equiv 0 \pmod{p} \). So assume \( p \) does not divide \( m \). Now if \( a \equiv 1 \pmod{p - 1} \), then \( a = 1 + (p - 1)k \) for some \( k \), so

\[
m^a = m^{1 + (p - 1)k} = m \cdot (m^{p-1})^k \\
\equiv m \cdot 1^k \pmod{p} \quad \text{(by Fermat’s Little Thm.)} \\
\equiv m \pmod{p}.
\]

\( \Box \)

(d) Prove that if \( n \) is a product of distinct primes, and \( a \equiv b \pmod{p} \) for all prime factors, \( p_i \), of \( n \), then \( a \equiv b \pmod{n} \).

Solution. By definition of congruence, \( a \equiv b \pmod{k} \) iff \( k \mid (a - b) \). So if \( a \equiv b \pmod{p} \) for each prime factor, \( p_i \), of \( n \), then \( p \mid (a - b) \) for each prime factor, \( p_i \), and hence, so does their product (by the Unique Factorization Theorem). That is, \( n \mid (a - b) \), which means \( a \equiv b \pmod{n} \).

\( \Box \)

(e) Combine the previous parts to complete the proof of Lemma 2.1.

Solution. Suppose \( n \) is a product of distinct primes, \( p_1 p_2 \cdots p_k \). Then from the formulas for the Euler function, \( \phi \), we have

\[
\phi(n) = (p_1 - 1)(p_2 - 1) \cdots (p_k - 1).
\]

Now suppose \( a \equiv 1 \pmod{\phi(n)} \), that is, \( a \) is 1 plus a multiple of \( \phi(n) \), so it is also 1 plus a multiple of \( p_i - 1 \). That is,

\[
a \equiv 1 \pmod{p_i - 1}.
\]

Hence, by part (c),

\[
m^a \equiv m \pmod{p_i}
\]

for all \( m \). Since this holds for all factors, \( p_i \), of \( n \), we conclude from part (d) that

\[
m^a \equiv m \pmod{n},
\]

which proves Lemma 2.1.
Appendix

Inverses, Fermat, Euler

Lemma (Inverses mod $n$). If $k$ and $n$ are relatively prime, then there is integer $k'$ called the modulo $n$ inverse of $k$, such that

$$k \cdot k' \equiv 1 \pmod{n}.$$

Remark: If $\gcd(k, n) = 1$, then $sk + tn = 1$ for some $s, t$, so we can choose $k' ::= s$ in the previous Lemma. So given $k$ and $n$, an inverse $k'$ can be found efficiently using the Pulverizer.

Theorem (Fermat’s (Little) Theorem). If $p$ is prime and $k$ is not a multiple of $p$, then

$$k^{p-1} \equiv 1 \pmod{p}$$

Definition. The value of Euler’s totient function, $\phi(n)$, is defined to be the number of positive integers less than $n$ that are relatively prime to $n$.

Lemma (Euler Totient Function Equations).

$$\phi(p^k) = p^k - p^{k-1} \quad \text{for prime } p, \text{ and } k > 0,$$

$$\phi(mn) = \phi(m) \cdot \phi(n) \quad \text{when } \gcd(m, n) = 1.$$

Theorem (Euler’s Theorem). If $k$ and $n$ are relatively prime, then

$$k^{\phi(n)} \equiv 1 \pmod{n}$$

Corollary. If $k$ and $n$ are relatively prime, then $k^{\phi(n)-1}$ is an inverse modulo $n$ of $k$.

Remark: Using fast exponentiation to compute $k^{\phi(n)-1}$ is another efficient way to compute an inverse modulo $n$ of $k$.

The Pulverizer

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

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

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

\[
\begin{align*}
\gcd(259, 70) & = \gcd(70, 49) & \text{since } \text{rem}(259, 70) = 49 \\
& = \gcd(49, 21) & \text{since } \text{rem}(70, 49) = 21 \\
& = \gcd(21, 7) & \text{since } \text{rem}(49, 21) = 7 \\
& = \gcd(7, 0) & \text{since } \text{rem}(21, 7) = 0 \\
& = 7. \\
\end{align*}
\]

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:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>rem(x, y)</th>
<th>x − q · y</th>
</tr>
</thead>
<tbody>
<tr>
<td>259</td>
<td>70</td>
<td>49</td>
<td>259 − 3 · 70</td>
</tr>
<tr>
<td>70</td>
<td>49</td>
<td>21</td>
<td>70 − 1 · 49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 70 − 1 · (259 − 3 · 70)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= −1 · 259 + 4 · 70</td>
</tr>
<tr>
<td>49</td>
<td>21</td>
<td>7</td>
<td>49 − 2 · 21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= (259 − 3 · 70) − 2 · (−1 · 259 + 4 · 70)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 3 · 259 − 11 · 70</td>
</tr>
<tr>
<td>21</td>
<td>7</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

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 $\text{rem}(x, y)$, which can be written in the form $x − q · y$. (Remember that the Division Algorithm says $x = q · y + r$, where $r$ is the remainder. We get $r = x − q · y$ 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.