Staff Solutions to In-Class Problems Week 6, Wed.

Problem 1.
Let’s try out RSA! There is a complete description of the algorithm in the text box. You’ll probably need extra paper. Check your work carefully!

(a) 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 hundreds of 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 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!

Problem 2. (a) Just as RSA would be trivial to crack knowing the factorization into two primes of $n$ in the public key, explain why RSA would also be trivial to crack knowing $\phi(n)$.

**STAFF NOTE:** Hint: The answer is so obvious you may wonder if you misunderstood the question.

**Solution.** If you knew $\phi(pq) = (p-1)(q-1)$ you could find the private key $d$ the same way the Receiver does using the Pulverizer to find the inverse mod $(p-1)(q-1)$ of the public key $e$.

(b) Show that if you knew $n$, $\phi(n)$, and that $n$ was the product of two primes, then you could easily factor $n$.

*Hint:* Suppose $n = pq$, replace $q$ by $n/p$ in the expression for $\phi(n)$, and solve for $p$. 

Creative Commons 2013, Eric Lehman, F Tom Leighton, Albert R Meyer.
Solution.

\[ \phi(n) = (p-1)(q-1) = n - p - \frac{n}{p} + 1, \quad \text{so} \]
\[ p\phi(n) = pn - p^2 - n + p \]
\[ 0 = p^2 + (\phi(n) - n - 1)p + n. \]

Now we can solve for \( p \) using the formula for the roots of a quadratic. \( \blacksquare \)

Problem 3.
A critical fact about RSA is, of course, that decrypting an encrypted message always gives back the original message, \( m \). Namely, if \( n = pq \) where \( p \) and \( q \) are distinct primes, \( m \in [0, pq) \), and

\[ d \cdot e \equiv 1 \pmod{(p-1)(q-1)}, \]

then

\[ \hat{m}^d := (m^e)^d = m \pmod{\mathbb{Z}_n}. \] \hspace{1cm} (1)

We’ll now prove this.

(a) Explain why (1) follows very simply from Euler’s theorem when \( m \) is relatively prime to \( n \).

Solution. By definition of \( d \) and \( e \), we have that

\[ de = 1 + k(p-1)(q-1) = 1 + k\phi(n) \]

for some integer \( k \). So

\[
\begin{align*}
(m^e)^d &= m^{de} = m^{1+k\phi(n)} \\
&= m \left( m^{\phi(n)} \right)^k \\
&= m \cdot 1^k \\
&= m \pmod{\mathbb{Z}_n},
\end{align*}
\]

(Euler’s Theorem for \( m \in \mathbb{Z}_n^* \)) \( \blacksquare \)

All the rest of this problem is about removing the restriction that \( m \) be relatively prime to \( n \). That is, we aim to prove that equation (1) holds for all \( m \in [0, n) \).

It is important to realize that, even if it was theoretically necessary, there would be no practical reason to worry about—or to bother to check for—this relative primality condition before sending a message \( m \) using RSA. That’s because the whole RSA enterprise is predicated on the difficulty of factoring. If an \( m \) ever came up that wasn’t relatively prime to \( n \), then we could factor \( n \) by computing \( \gcd(m, n) \). So believing in the security of RSA implies believing that the probability of a message \( m \) turning up that was not relatively prime to \( n \) is negligible.

But’s let’s be pure, impractical mathematicians and rid of this technically unnecessary relative primality side condition, even if it is harmless. One gain for doing this is that statements about RSA will be simpler without the side condition. More important, the proof below illustrates a useful general method of proving things about a number \( n \) by proving them separately for the prime factors of \( n \).

(b) Prove that if \( p \) is prime and \( a \equiv 1 \pmod{p-1} \), then

\[ m^a = m \pmod{\mathbb{Z}_p}. \] \hspace{1cm} (2)
**Solution.** If \( p \mid m \), then equation (2) holds since both sides equal 0 in \( \mathbb{Z}_p \). On the other hand, if \( p \) does not divide \( m \), then (2) holds by part (a).

(c) Give an elementary proof\(^1\) that if \( a \equiv b \pmod{p_i} \) for distinct primes \( p_i \), then \( a \equiv b \) modulo their product.

**Solution.** If \( a \equiv b \pmod{p_i} \), then by definition, \( p_i \mid (a - b) \). By the Unique Factorization Theorem 8.4.1, the product of the \( p_i \)'s must therefore also divide \( a - b \), which means that \( a \equiv b \) modulo their product.

(d) Note that (1) is a special case of

**Claim.** If \( n \) is a product of distinct primes and \( a \equiv 1 \pmod{\phi(n)} \), then

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

Use the previous parts to Prove the Claim.

**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 \( a \) is also 1 plus a multiple of \( p_i - 1 \), namely,

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

Hence, by part (b),

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

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

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

which proves the Claim.\(\)

---

\(^1\)There is no need to appeal to the Chinese Remainder Theorem.