Notes for Recitation 7

1 RSA

In 1977, Ronald Rivest, Adi Shamir, and Leonard Adleman proposed a highly secure cryptosystem (called RSA) based on number theory. Despite decades of attack, no significant weakness has been found. (Well, none that you and me would know…) Moreover, RSA has a major advantage over traditional codes: the sender and receiver of an encrypted message need not meet beforehand to agree on a secret key. Rather, the receiver has both a secret key, which she guards closely, and a public key, which she distributes as widely as possible. To send her a message, one encrypts using her widely-distributed public key. Then she decrypts the message using her closely-held private key. The use of such a public key cryptography system allows you and Amazon, for example, to engage in a secure transaction without meeting up beforehand in a dark alley to exchange a key.

\begin{center}
\textbf{RSA Public-Key Encryption}
\end{center}

\textbf{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 \textit{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 \textit{secret key} is the pair \( (d, n) \). This should be kept hidden!

\textbf{Encoding} The sender encrypts message \( m \) to produce \( m' \) using the public key:

\[ m' = m^e \mod n. \]

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

\[ m = (m')^d \mod n. \]
2 Let’s try it out!

You’ll probably need extra paper. Check your work carefully!

• As a team, go through the beforehand steps.
  – Choose primes $p$ and $q$ to be relatively small, say in the range 10-20. 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.

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

• 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 suck!
  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.

• Decrypt the message sent to you and verify that you received what the other team sent!

• 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.
3 But does it really work?

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 \( k \).

- This congruence 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:

\[
m^{1+k(p-1)(q-1)} \equiv m \cdot (m^{p-1})^{k(q-1)} \pmod{p}
\equiv m \cdot 1^{k(q-1)} \pmod{p}
\equiv m \pmod{p}
\]

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

- 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:

\[ p \mid (m - m^{ed}), \]
\[ q \mid (m - m^{ed}). \]

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

\[ m \equiv m^{ed} \pmod{pq}. \]