Math 126 Number Theory

Prof. D. Joyce, Clark University

15 Mar 2006

Due Friday. Asmt. 10. From page 106: 2, 4, 5, 9; and from page 108 Misc. exercises: 8, 15.

Quiz Monday.

Last time. The group of totatives modulo n. The order of a totative a, denoted $\mathrm{ord}_n(a)$. Primitive roots. A couple of theorems including these two important ones:

Corollary. The order of a totative a divides $\phi(n)$.

Theorem. For a prime p, if d divides $p-1$, then there are exactly $\phi(d)$ totatives whose order is d. In particular, there are exactly $\phi(p-1)$ primitive roots of p.

Today. Public-key cryptography, in particular, the mathematics behind the RSA algorithm.

The most used cryptography system is the RSA algorithm proposed by Rivest, Shamir, and Adleman in their article "On Digital Signatures and Public Key Cryptosystems," *Communications of the ACM* 21 (1978): 120–126. We'll look at the number theory behind the algorithm today. We only need what we've already studied, the most important parts being (1) the Euclidean algorithm, and (2) Euler's theorem:

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

for a relatively prime to n.

This is a public-key system in which the key needed to encode messages is made public, but the key needed to decode messages is kept private. It works because the private key cannot be determined from the public key. (At least it can't be determined easily.)

The RSA algorithm is based on exponentiation modulo n. The encoding and decoding algorthims are actually functions $\mathbf{Z}_n \to \mathbf{Z}_n$. That means that the message has to start out as an element a in $\mathbf{Z}_n$, that is, a number $0 \le a < n$. A real message is actually a string of characters, so a preliminary coding is needed to convert that into a string of numbers modulo n.

This RSA system is public-key which means the algorithm for coding is made public, but the inverse algorithm for decoding is kept private. You would think that if you knew one function, it wouldn't be hard to find the inverse function. But there are a number of these *trap-door* functions that don't seem to be easily inverted.

Here are the steps in creating the two keys for the encoding and decoding functions for the RSA system.

1. Select two large prime numbers p and q, and let $n = pq$. Then $\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$.

2. Select a number e relatively prime to $\phi(n)$. (This number e can be pretty small, typically $e = 3$.)

3. Compute $d \equiv e^{-1} \pmod{\phi(n)}$, that is, solve the congruence $ex \equiv 1 \pmod{\phi(n)}$, and call the solution d. Solving that linear congruence will involve the Euclidean algorithm.

4. The encoding algorithm is the function $\mathbf{Z}_n \to \mathbf{Z}_n$ which converts the original message a into the coded message $a^e \pmod{n}$. Make public this encoding algorithm, that is, make public n and e.

5. The decoding algorithm is the function $\mathbf{Z}_n \to \mathbf{Z}_n$ which converts an coded message b back into the original message $b^d \pmod n$. Keep private this decoding algorthm, that is, don't tell anyone d, and don't tell anyone p or q either, because then d could be determined.

Why is the decoding algorithm actually inverse to the encoding algorithm? Let's check that starting with a message a, then encoding it, then decoding it, returns the original message a. That is, we need to verify the congruence

$$(a^e)^d \equiv a \pmod n.$$

We'll do that in two cases. First, when a is relatively prime to n (which is the case for nearly all a). In that case, we can use Euler's theorem modulo n. We'll also use the fact that $ed \equiv 1 \pmod{\phi(n)}$, that is, that $ed = 1 + c\,\phi(n)$ for some number c. Then, modulo n we have

$$\begin{aligned}(a^e)^d &\equiv a^{ed} \\ &\equiv a^{1+c\,\phi(n)} \\ &\equiv a\,(a^{\phi(n)})^c \\ &\equiv a\,1^c \equiv a \pmod n\end{aligned}$$

That takes care of the case that $(a, n) = 1$.

Suppose now that a is not relatively prime to n. If $a = 0$, then clearly, $(a^e)^d \equiv a \pmod n$. Otherwise a is divisible by exactly one of the two primes p and q. Let's say p divides it. Of course, modulo p we have $(a^e)^d \equiv 0 \equiv a \pmod p$. Now, since a is relatively prime to q, we can apply Euler's theorem modulo q. Then we have

$$\begin{aligned}(a^e)^d &\equiv a^{ed} \\ &\equiv a^{1+c\,\phi(n)} \\ &\equiv a^{1+c(p-1)(q-1)} \\ &\equiv a\,(a^{q-1})^{c(p-1)} \\ &\equiv a\,1^{c(p-1)} \equiv a \pmod q\end{aligned}$$

Now, since $(a^e)^d \equiv a$ both modulo p and modulo q, therefore the congruence holds modulo their product $n = pq$.

That finishes the proof that the decoding function actually is inverse to the encoding function.