Summer High School 2009
Aaron Bertram

7. RSA Encoding and Decoding. We now have the tools to analyze (and implement) a public-key encryption scheme known as RSA. It is a scheme whereby the tools to encode a message are public, so that anyone can create a secret message, but the tools to decode the secret messages are kept secret. Its effectiveness depends upon the following:

- Large numbers can't be factored in a reasonable amount of time.
- The Euler phi function of a large number is impossible to compute in a reasonable amount of time without knowing its factorization, BUT computing it is a cinch if you know the factorization.
- In contrast, taking very large *powers* of one very large number modulo another very large number can be done relatively quickly.

Private (Top Secret): Two large prime numbers p and q.

Public (Open to All): The product $n = pq$ of the two primes and an (also large) additional number m that is relatively prime to $\phi(n)$. Also, the cipher and block size (see below).

Privately: From the private information, it is easy to see that:

$$\phi(n) = (p-1)(q-1)$$

but this is impossible to figure out from the public information!

How to Encode: The cipher is a method for replacing each letter (and space) of our message with a number.

Let's agree on the following simple cipher:

$$A = 11, B = 12, \ldots, Z = 36, < \text{space} >= 99$$

so that, for example, "Happy Birthday" becomes:

18 11 26 26 35 99 12 19 28 30 18 14 11 35

(This is easy for codebreakers to crack. What you do next is diabolical.)

We agree on a block size of numbers to encode. Typically the block size give us numbers just smaller than the number n that we are given in the public key. These numbers may have 80 or more digits, so that strings of 40 letters can be encoded at once. For this simple example, we'll choose block sizes of 3 numbers (to keep things manageable!), so that we are using RSA to encode the string of three-digit numbers:

181 126 263 599 121 928 301 814 113 500

Suppose that the meat of the public key consists of:

$$n = 1147 \quad \text{and} \quad m = 517$$

(take my word for it, for now, that m is relatively prime to $\phi(n)$). Then we encode each of the strings *abc* of three digits by computing:

$$(***)^m \bmod n$$

and listing it as another string of digits. This can be done quickly using the Binary and successive squaring techniques that we've just learned, although even in this simple example we already need a computer!

$$(181)^{517} \equiv 367 \bmod 1147$$
$$(126)^{517} \equiv 686 \bmod 1147$$
$$(263)^{517} \equiv 891 \bmod 1147$$
$$(599)^{517} \equiv 144 \bmod 1147$$
$$(121)^{517} \equiv 417 \bmod 1147$$
$$(928)^{517} \equiv 585 \bmod 1147$$
$$(301)^{517} \equiv 827 \bmod 1147$$
$$(814)^{517} \equiv 777 \bmod 1147$$
$$(113)^{517} \equiv 607 \bmod 1147$$
$$(500)^{517} \equiv 264 \bmod 1147$$

(Full disclosure: I picked these numbers small enough so my application can compute all the powers without binary and successive squaring.) So we have encoded our message!

367 686 891 144 417 585 827 777 607 264

How to Decode: We start with the (private) $\phi(n)$ and solve:

$$am + b\phi(n) = 1$$

Now suppose k is any natural number, and consider:

$$k^1 = k^{am+b\phi(n)} = k^{am}k^{b\phi(n)} = (k^m)^a(k^{\phi(n)})^b \bmod n$$

and if k and n are relatively prime, then Euler's formula tells us that $k^{\phi(n)} \equiv 1 \bmod n$, so:

$$(k^m)^a \equiv k \bmod n$$

This means that if $k = (***)$ is the number we want to recover, then

$$(***)^m \text{ is the encoded number, and } ((***)^m)^a$$

returns us back to $k \bmod n$. In other words, all we have to do to decode the message is to raise the encoded numbers by the (secret) power a. But to emphasize, the number a is only known privately, since it requires us to know $\phi(n)$, which in turn requires p and q.

In our example, the top secret p and q are 31 and 37, so:

$$\phi(n) = 1080,\ m = 517 \text{ and } a = 493,\ b = -236 \text{ from Euclid}$$

and you are invited to go ahead and check that:

$$(367)^{493} \equiv 181 \bmod 1147$$
$$(686)^{493} \equiv 126 \bmod 1147$$
$$\text{etc.}$$

For Thought: What do you do if $a < 0$?

A Tiny Fudge: We assumed $\text{GCD}(k, n) = 1$ to decode the message. Disaster would ensue if they failed to be relatively prime because then $\text{GCD}(k, n) = p$ or q, and the code would be broken! Fortunately, the chances are infinitesimal for a number generated essentially at random from the message to share a common factor with n.

Recap:

Public Knowledge: Numbers n and m with $\text{GCD}(m, \phi(n)) = 1$.

How to Encode: Use a cipher and blocks to convert the message to a string of numbers $(***)$ of approximately the same size as n. Then replace each string of numbers with $(***)^m$ mod n.

Private Knowledge: The prime factorization $n = pq$, from which $\phi(n) = (p-1)(q-1)$ and $am + b\phi(n) = 1$ are easily computed.

How to Decode: Replace each encoded $(***)$ with $(***)^a$ to recover the original! Then use the blocks and cipher to recover the message.

What could go wrong? Some string $(***)$ might not be relatively prime to n. But the chances of this happening are infinitesimal.