Math 222: A quick introduction to RSA public key encryption scheme

The RSA public key encryption scheme is widely used on the internet to encode network traffic. Mainly items like credit card numbers and account numbers. RSA stand for *R*ivest, *S*hamir and *A*dleman; the 3 people who discovered it.

Let's encode "GO ND". First we will turn the letters into numbers. We will use 01 for "a", 02 for "b", etc and 27 for " " (a blank space). So "GO ND" becomes "0715271404". Now we will break this up into *words* of length $r = 2$ to encode. So this becomes 07 15 27 14 04. (When r is small the scheme can be broken so in actual use r is chosen to be much larger than 2.)

Now we choose two prime numbers, say $p = 31$ and $q = 37$ and let $m = pq = 1147$. m is called the *modulus* and is known to the public, while the primes p and q are not. Let w be a word. Since the largest word (in our case a two digit number from 01 to 27) is 27, we have $p, q > w$. This is the important fact and we must always choose r, p and q such this occurs. Note that this implies w has a multiplicative inverse in $\mathbf{Z}_m$.

We now choose a number e called the *encoding factor* which is also known to the public. We'll see how to choose this below. Then we encode the word w as the number w^e modulo m. For example, take $e = 11$. Then "07" is encoded as the number $07^{11} = 1977326743 \equiv_{1147} 826$. Doing this for each word, you can check that "GO ND" is encoded as 826 759 492 970 872. *Note:* There are some nice tricks to make this encoding go very quickly.

Now how do we decode? It is not as simple as taking the eth root of w^e in $\mathbf{Z}_m$ since there may be e different eth roots of w^e and more than one of them may be a valid possibility for w. (Remember that in our situation we need $0 \leq w \leq 27$.) Indeed, the larger the value of r the more likely it is that a random eth root of w^e will be valid. So, how do we figure out which is the "correct" value of w? Thankfully, there is a definition and theorem to the rescue.

Definition: Let $\phi(m)$ be the number of positive integers x less than m such that x and m are relatively prime.

In our situation, since $m = pq$, it is easy to compute $\phi(m)$. For, what numbers are *not* relatively prime to $m = pq$? Precisely those numbers which are either a multiple of p or a multiple of q. That is, those numbers of the form kp for $0 \leq k \leq q-1$ or of the form kq for $0 \leq k \leq p-1$. Now, there are q numbers of the first type and p numbers of the second type. However, $0 = 0p = 0q$ was counted twice, so there are $p + q - 1$ numbers less than m and not relatively prime to m. Hence, $\phi(m) = m - (p + q - 1) = pq - p - q + 1 = (p-1)(q-1)$. For example, we get $\phi(1147) = (30)(36) = 1080$.

Notice that if p is prime, then $\phi(p) = p - 1$ and recall Fermat's theorem says that if a is non-zero in $\mathbb{Z}_p$, then $a^p \equiv_p 1$. It turns out that there is a generalization of this involving the Euler number:

Euler's Theorem: If w has a multiplicative inverse in $\mathbf{Z}_m$ then $w^{\phi(m)} \equiv_m 1$.

Now lets look back at our choice of $e = 11$. e was chosen so that e and $\phi(m)$ would be relatively prime. Indeed, 11 is relatively prime to $30 \times 36 = 1080$. Hence e has a multiplicative inverse d modulo $\phi(m)$. The number d is called the *decoding factor* and is not known to the public. In our case, using Euclid's algorithm, one can figure out that $e = 11$ has a multiplicative inverse modulo 1080 of $d = 491$. In general, we can then decode w^e by simply computing $(w^e)^d$ since

$$(w^e)^d \equiv_m w^{ed} \equiv_m w^{l\phi(m)+1} \equiv_m (w^{\phi(m)})^l w \equiv_m w,$$

for some l.

Here, if we compute 826^{491}, we see that $826^{491} \equiv_{1147} 7$. It works! Again, there are some tricks that can be used to compute this number but we did not use any of them we just used Mathematica.

Summarizing, let's see how to decode something. It's "easy": (1) Factor m to find p and q; (2) Compute $\phi(m) = (p-1)(q-1)$; (3) find d (Euclid's algorthim is just one way; a clever use of Euler's Theorem is another), and (4) compute $(w^e)^d$ modulo m. Now, the hard part of this can be step 1. It would take at least several months even with the latest advances in factoring for all the computers at Notre Dame to factor m if p and q are chosen to have 70 or more digits. On the other hand, if you properly implement this encoding and decoding scheme and use it to transfer data between computers (there are some small pitfalls to avoid) using primes of 154 digits, you will have a program which you cannot legally export. By the way, it is possible to quickly find primes of this size.

Homework: Given that $m = 1189$ and $e = 13$, find d. Decode the coded symbol 419. *Hint:* use Mathematica. One can find $\phi(m)$ in Mathematica with the command "EulerPhi[m]". The notebook from the above example is on the web.