Lesson Nineteen

Math 6080 (for the Masters Teaching Program), Summer 2020

19. RSA. Messages coded with the use of a cipher can be pretty easy to decode. An RSA-encrypted message, however, can be virtually impossible to decrypt without a "private key". The message is encrypted with a publicly available "key" but the private key is known only to the person decrypting the message.

Private Key. Two large prime numbers p and q.

Public Key. The product $n = pq$ and an additional (large) r satisfying

$$\gcd(r, \phi(n)) = 1$$

(r is chosen privately along with p and q, but it is shared along with n).

We first see how to encrypt and decrypt (large) **integers** x and y.

How to encrypt x. Raise x to the rth power modulo n.

$$y = (x**r)\%n$$

How to decrypt y. In private, pull p and q from the vault and solve:

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

Armed with this and the enhanced Euclid algorithm, solve:

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

Then assuming that x is relatively prime to n (which is a statistical certainty), raising y to the ath power recovers x:

$$(y**a)\%n = x$$

This is because:

$$(y**a) = (x**r)**a = x**(ra)$$

and

$$x = x**1 = x**(ra + b\phi(n)) = (x**(ra)) * (x**(b\phi(n)))$$

and

$$x**(b\phi(n)) = (x**\phi(n))**b = 1**b = 1$$

by Euler's Theorem!

Exercise. (a) Write the encryption Python code. Prompt for the public numbers n and r, and the 'message' number x. Encrypt the number as:

$$y = (x**r)\%n$$

and return it to the user.

(b) Write the decryption Python code. Prompt the user for the private primes p and q and the public number r. Use the enhanced Euclid algorithm to solve:

$$ar + b(p-1)(q-1) = 1$$

Prompt the user for the "secret message" number y and print the decoded number

$$x = (y**a)\%n$$

(Tricky point. You need to handle the possibility that a may be a negative number.)

Our next job is to encode a message. We will do this with punctuation and all. The ascii values of every character in a message sent in English vary from:

$$\text{ord}('(\text{space})') = 32 \text{ to } \text{ord}('\sim') = 126$$

so we could add 100 to each of these numbers to get three-digit numbers from:

$$132 \text{ to } 226$$

Chop a given string into groups of three (adding one or two spaces if necessary) and via the ord function, rewrite these as nine digit numbers. For example:

Happy Birthday

is turned into the nine digit numbers.

172197212 212221132 166205214 216204200 197221132

Then you encrypt those nine digit numbers with your public key and back at CIA headquarters these nine digit numbers are recovered with the private key, and turned back into the message.

Exercise. Write Python code to implement this! Then find two ten digit primes p and q and an additional nine digit prime r. Give the class the twenty (or so) digit number $n = pq$ along with r and have them pass encrypted messages to you, which you then decrypt to everyone's amazement.