Exploratory Group Problem: RSA Encryption

K. Stange

November 5, 2008

This problem counts as THREE exploratory problems. It requires working in a group of at least two people, and at most four. You should submit ONE set of answers for the entire group. In this problem you will encode and decode using a cryptographic system called RSA using modular arithmetic. Divide your group into two teams (consisting of one or two people each). The instructions are for two people (or teams of people), here called Alice and Bob. You are allowed the use of a modest calculator.

1. Preparatory Phase (everyone works together):

   (a) Make up a table of the Simple Cipher which turns letters into numbers and vice versa according the pattern:

   $$A = 11, B = 12, C = 13, \ldots, W = 33, X = 34, Y = 35, Z = 36.$$

   So for example, 'HELP' is the number 18152226. This cipher isn't safe, because it is so simple. We call '18152226' the 'numbertext'. Cipher and decipher a few simple words and test each other to make sure you can use the cipher correctly.

   (b) How many invertible elements are there in $\mathbb{Z}_N$ where $N = pq$ is a product of two primes? Explain why. The answer to this question is $\phi = N - p - q + 1$. If you didn't get that, then go back and think about it again. Explain why this implies

   $$x^{\phi} \equiv 1 \mod N \text{ for all } x$$

   by considering the size of the group of multiplicatively invertible elements of $\mathbb{Z}_N$, and the size of the subgroup generated by x.

(c) Calculate $12398732^{3140000100}$ modulo $323 = 19 \times 17$ using only a pocket or equivalent basic onscreen operating-system calculator for help (your calculator should do multiplications, additions and subtractions, and maybe keep a running result and you shouldn't use any more advanced functions). To do this, you will need to develop some methods of fast modular arithmetic, since your calculator can't do it all at once (I hope). You can also parallelize the work and farm it out to everyone in the group. Tricks to use:

   i. One easy trick is this: when reducing modulo N, remember that $N \times 100$ and $N \times 10$ are easy-to-use multiples of N. For example, $51700 \equiv 1100 \equiv 88 \mod 506$ since 50600 and 1012 are easy-to-spot multiples of 506. Sometimes it helps to keep a list of easy-to-spot multiples. The basic paradigm is to do the calculation a piece at a time, always reducing modulo the modulus between each step.

   ii. First calculate $\phi = \phi(323)$ as in part b), and then reduce the exponent modulo ϕ. By part b), this will not change the answer.

   iii. However, even after the last part, the exponent is still fairly large, call it E. Calculate successive squares of the base $B = 12398732$ modulo 323, writing them in a list: $B^0 \equiv 1, B^1 \equiv 54, B^2 \equiv 9, \ldots$ modulo 323. Use the binary expansion of E to write B^E as a product of elements of your list. Calculate the product, reducing at each step.

   By the way, the answer is $12398732^{3140000100} \equiv 115 \mod 323$. Show your work.

2. Key Generation (Bob does this):

   (a) Bob: Pick two **secret** primes, $40 < p_b, q_b < 100$. Don't tell Alice what they are – these are secret. Make sure they are actually prime or this won't work! You can find lists of primes on the internet if you aren't sure.

   (b) Bob: Calculate $N_b = p_b q_b$.

   (c) Bob: Calculate **secret** $\phi_b = N_b - p_b - q_b + 1$.

   (d) Bob: Choose a k_b satisfying $10 < k_b < 100$ which is relatively prime to ϕ_b. You can probably find one by picking a random number. Do Euclid's algorithm to check its gcd with ϕ_b is actually 1. From your work with Euclid's algorith, you will be able to find the multiplicative inverse of k_b modulo ϕ_b. Call this g_b and keep it a **secret**.

(e) The numbers N_b and k_b are your **public key**. Either write your public key on a sign around your neck, or publish it on your facebook page. Anyone can see it!

(f) The numbers ϕ_b and g_b are your **secret key**. You can forget what p_b and q_b are.

3. Coding Phase (Alice does this):

   (a) Alice: Choose a short **secret** message and calculate its **secret** numbertext.

   (b) Alice: Break up your secret numbertext into blocks of length three and pad the last block so it is also of length three (for example 'HELP' is 181 522 260).

   (c) Alice: Now, for each block x, do the following. Consider x as a three-digit number. Compute $x^{k_b} \mod N_b$. Note that you can't use ϕ_b since you don't know what it is, but Bob was kind enough to choose a fairly small k_b so it won't matter.

   (d) Alice: Now, write the list of results (a list of numbers), in order, and call it 'Message for Bob.' Write it on his facebook wall (or deliver it in some other public way). Even though everyone can see the message, only Bob will be able to decode it.

4. Decoding Phase (Bob does this):

   (a) Bob: For each of the numbers y in the list that Alice has given you, calculate y^{g_b} modulo N_b (you can use ϕ_b since you know it). Use the fast modular arithmetic methods. You should get a series of numbers as a result. Run them all together in a line and read each pair of digits as a letter according to the Simple Cipher. You have read Alice's message!

5. Now repeat #2-4 with the teams switched, so Bob sends a message to Alice.

6. Analysis Phase (Everyone works together)

   (a) Show that the message Bob gets back is in fact the message Alice coded up.

   (b) Can you think of vulnerabilities in this method? For example, if Eve is an evesdropper who reads Bob's facebook page, she will know Bob's public key and Alice's coded up message. Can she decode the message? Why can't she use Bob's method of decoding? How might she attack this system and figure out the secret message? What is the 'hard problem' protecting the security of this system?