Discrete Structures: Worksheet 1

Today we're going over the basic theory behind RSA encryption, which is one of the oldest and most widely used encryption schemes. First, we'll look at technique to 'scramble' and 'unscramble' some integers and then notice that because numbers can encode whatever data you like, this scrambling and unscrambling can be used to encrypt data. The 'scrambling' will be easy and anyone with a public key can do it. The unscrambling requires knowledge of how to factor a particular number.

SCRAMBLE STAGE USUALLY CALLED THE *encryption* STAGE):

Take two primes p and q (in practice, these are very large primes). Set $n = pq$ and pick some e that is relatively prime to $(p \quad 1)(q \quad 1)$. The pair $(n \ e)$ is the *encryption key*. You can give this to anyone and they can encrypt data as follows:

- Take any integer $0 \leq m < n$. The encrypted value of m is

$$m^e \mod n$$

(1) Encrypt the following numbers with the key (43 59 13) using the modular exponentiation technique **from last week (note: we're using small primes for the examples here).**

- 1819
- 1415

UNSCRAMBLE STAGE USUALLY CALLED THE *decryption st ge*)

So, anyone can encrypt. To decrypt though, it seems like we need special knowledge of what the primes p and q are so that we can find an inverse of$e$ mod $(p \quad 1)(q \quad 1)$. Let's suppose that we know an inverse of e, call it d. We will prove here that $(m^e) \equiv m \mod n$, that is, **exponentiation by d unscrambles exponentiation by e.**

- We know that $ed \equiv 1 \mod (p \quad 1)(q \quad 1)$. This means that $ed = 1 + k(p \quad 1)(q \quad 1)$ for some integer k.

(2) Use the above bullet point to prove that

$$(m^e) \equiv m \mod n$$

by analyzing three cases and applying Fermat's little theorem along with the Chinese remainder theorem:

- Assume that m is not divisible by p or q
- Assume that m is equal to 0, or
- Assume that m is divisible by exactly one of$p$ or q, but not both.

Why are these three cases adequate?

(3) Find an inverse of 13 modulo (42)(58) and check that the decryption works for the two numbers you encrypted earlier.

(4) Decrypt 0981 and 0461 (the leading zeros are cosmetic).

1. ENCODING DATA

There are many ways to encode data as a sequence of integers. A simple way to encode text as a string of integers is to find the greatest number of consecutive 25's that is less than the product of our two primes. In our concrete example, we can only fit two consecutive 25's, which means that we can only encrypt two letters at a time.

(1) Encode the word $MATH$ as two integers between 0 and $43 \cdot 59$. Encrypt your message using the RSA scheme.

(2) What words have we been encoding from our earlier examples?