Matthew Milone
Internet Security and RSA Encryption

1. Modular arithmetic is a system of arithmetic in which numbers “wrap around” when we reach a number known as the modulus. We start by performing the operation, then dividing the answer by a modulo and taking the remainder. It might be helpful to visualize a clock. If it is 8 o’clock and we want to add 7 hours, we get:

$$8 + 7 \equiv 15 \ (mod\ 12) \equiv 3$$

Calculate the following:

a) $24x4 \ (mod\ 5)$

b) $50 + 8 \ (mod\ 4)$

c) $200 - 43 \ (mod\ 50)$

2. The totient function $\varphi(n)$, also called Euler's totient function, is defined as the number of positive integers $\leq n$ that are relatively prime to (i.e., do not contain any factor in common with) n, where 1 is counted as being relatively prime to all numbers (Wolfram MathWorld).

$\varphi(n)$ is multiplicative when n is the product of two relatively prime numbers a and b. This means that $\varphi(n) = \varphi(a * b) = \varphi(a) * \varphi(b)$. Let’s test to see if this is true.

a) Find $\varphi(3)$.

b) Find $\varphi(5)$.

c) Find $\varphi(15)$.

d) Does $\varphi(3 * 5) = \varphi(3) * \varphi(5)$?

3. We want to send a message *m*. To do this:

- We will choose two prime numbers, called *p* and *q*.
- These will be multiplied together to get a modulus, *n=p*q*.
- We can use *p* and *q* to calculate *φ(n)*:

$$\varphi(n) = \varphi(p * q) = \varphi(p) * \varphi(q) = (p-1)(q-1)$$

- A public key *e* is chosen; it should be a prime number on the interval [3,*φ(n)*) such that *e* and *φ(n)* share no common factors.
- A private key *d* is chosen such that *e*d≡1 (mod φ(n))*
- *m* is encrypted as ciphertext *c* using the following formula:

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

- *c* is decrypted by the message recipient as follows:

$$m \equiv c^d \ (mod\ n)$$

Encrypting the message:

We have a message $m = 42$.

We choose two prime numbers, $p = 5$ and $q = 11$.

Find $n = p * q$:

Find $\varphi(n) = (p-1)(q-1)$:

Let's choose a public encryption key $e = 3$.

Find a private decryption key d such that $e * d \equiv 1 \ (mod\ \varphi(n))$:

Encode the message using the formula $c \equiv m^e \ (mod\ n)$:

Decrypting the message:

Now decode the ciphertext *c* using the formula $m = c^d \ (mod\ n)$:

Did you get your original message? Do the encryption and decryption processes work?