RSA

----- Motivation -----

The Rivet-Shamir-Adleman (RSA) Cryptosystem is an asymmetric cryptosystem that is used today by modern computers to encrypt and decrypt messages. Asymmetric means that there are two different keys. It is also classified as a public key cryptosystem since one of the keys can be given to anyone. RSA is based on the principle that it is easy to multiply large numbers, but factoring large numbers is very difficult.

RSA is used today in modern computers. IIt is used in a range of web browsers, email, VPNs, chat and other communication channels and it is considered the most secure way of encryption. So, it is a very valuable cryptosystem to learn for many computer scientists!

----- Encrypt -----

Before we begin the encryption method, there are a few things we must set up.

Step 1: pick two -large- primes

The bigger the prime, the more secure the system is. For better visualization and understanding, I will use two smaller primes: p = 11 and q = 3

If you want to try larger numbers, I would suggest checking out this website: https://bigprimes.org

Easy enough! Next step!

Step 2: find N(modulus) and Euler's Phi

N is the product of the two primes we picked earlier, it's simply: N = p*q

With my example, N = 3*11 = 33

As for Euler's Phi function, it's: ϕ = (p-1)(q-1). (attach popup 1)

With the nubes above, it is: (11-1)(3-1)=20

Step 3: Find e, the *public exponent*

We must choose an integer e such that: 1< e < ϕ , and: gcd(e,ϕ) = 1, (If you can't think of one, we can find a random e for you)!

We must find an example such that gcd(e,20) = 1. 3 fits this criteria! So, e = 3.

Step 4: Send out the public key!

Public key = (N,e) which in our example is (33,3)

Now the encryption begins!

Let's suppose Alice wants to send a message to Bob. What does she have to do?

1. Pick a message m to encrypt. In our example, we will use m = 7
2. Obtain Bob's public key (N,e) to compute the ciphertext: c ≡ m^e mod N.
   Which is, $7^3 \ mod \ 33$ ⇔ 343 mod 33 ⇔ 13 mod 33
   So, our ciphertext is 13.
3. Send the Cipher text to Bob!

Encryption is done! Click decryption to see how to decrypt the message!

----- Decrypt -----

Recall, We already know p, q, and the public key (N,e).

There is one simple step Bob has to do before starting the encryption:

Generate private key

For this step, Bob has to compute the secret exponent d such that: 1<d<ϕ, and ed ≡ 1 mod ϕ (attach popup 2)

In our example, we must find a value such that 3d≡1 mod 20

Using simple methods learned in discrete math, we are looking for a d such that 20 divides 3d-1. Since our numbers are very small we can test and see that 7 works. d = 7.

So the private key is: which in our example is (33,7)

Now the decryption begins!

Bob only has to do one thing:

Compute m' ≡ $c^d mod\ N$. m' will equal the plaintext m!

In our example we will have

Alice sent Bob the ciphertext c = 13.

Bob will compute the following:

m' ≡ $c^d mod\ N$ ⇔ $13^7 mod\ 33$ ⇔ 7

Note: We don't have to compute the full value of 13^7 . we can factor and make use of the fact that:

A = bc mod n ⇔ (b mod n)(c mod n) mod n

Done! :)

**********popup 1**********

The original RSA used the Euler totient function $\phi(n)$ = (p-1)(q-1). Recent standards of RSA use the *Charmichael function* λ(n) = lcm(p-1,q-1).

λ(n) is smaller than $\phi(n)$ and divides it. The value of d' computed by d' = e^{-1} mod λ(n) is usually different from from that derived by d = e^{-1} mod $\phi(n)$, but the end result is the same.

Both values (d and d'), will decrypt a message m^e mod n, and both will have the same signature value s = m^d and n = $m^{d'}$ mod n. To compute λ(n), we can use this relation: $\lambda(n) = \frac{(p-1)(q-1)}{gcd(p-1, q-1)}$

**********popup 2**********

To compute d, we can use the Extended Euclidean Algorithm to find the modular inverse, that is to calculate: d = e^{-1} mod ϕ.

The modular inverse f is defined as the integer value such that ed = 1 mod ϕ. It only exists if e and ϕ have no common factors (which we make sure of in step 3 of our encryption).

****