Bob wants to send a message to Alice. His message is "42" which is my age and the answer to Life, the Universe and Everything.

Alice chooses two "large" primes : 7 and 13, which means:





Alice has to choose  so that . She chooses:





these numbers work because  is divisible by 72

She makes public:





She keeps private:









To encrypt his message , he computes:

   35

let 

he sends the message Y in public channels to Alice

Alice receives the message Y and computes:

42

so Alice has received Bob’s message

Observations and Questions:

\* choosing  was easy b/c they were tiny, but choosing  was much harder. On my first few tries,  ended up being the same as  which probably isn’t good OR I got , which I assume is also not a good thing.

\* Since  is public, how does she know the message is from Bob? Should he include a signature at the beginning or end of every message? probably a good idea.

\* When Alice is decrypting, the numbers got very large and my calculator can’t do modular math, so I had to find an online program that would do it for me. If folks in the real world are dealing with 200 digit numbers and they need every decimal place to be accurate, how do they accomplish this quickly?

\* After playing with this for a bit, I realized that I couldn’t send a message whose ‘value’ was larger than , so I tried all of this again with larger primes.

I started by googling a list of all primes up to 10,000 and picked two of them:  and . This gave me:

 and



Finding  then became much harder, so I created an Excel spreadsheet to help me by looking for divisors of . I came up with two choices for  relatively quickly and settled on:

 and



I then used this set of values to encrypt the message MATH (13,012,008: M=13, A=01 etc), which I sent to Alice, who decrypted it just fine. Pretty cool.

I used the modular arithmetic calculator on [www.ptrow.com](http://www.ptrow.com) to carry out the calculations.

A final observation is that if Eve intercepts my encrypted message (which, by the way is ), would she have a fairly hard time decrypting it because she doesn’t know the decryptor exponent ? She does know the modulus, so I guess she could try different values of  with the public modulus . A speedy computer could probably handle this quickly, but I can now see that if the original primes are 200 digits or so long, how a supercomputer doing trillions of calculations a second would have a hard time. Also, I suppose you could add an additional layer of security by first encrypting the message with a Ceasar (or some other) cipher before sending?