

Exercise 1

$de \equiv 1 \pmod{\phi(n)}$. Since we know the factorization of n (101 and 113) $\phi(n)$ is just $(p-1)(q-1)$, or 11200. With $\phi(n)$ known to find d we just need to find $e^{-1} \pmod{\phi(n)}$, which is 3. Now that we know d , we can find the plaintext with $c^d \pmod{n}$, which is 1415.

Exercise 3

First Alice and Bob find a and b relatively prime to $p-1$ where p is some large prime. They also find the multiplicative inverse of a and $b \pmod{p-1}$ because they'll need that later.

$$p = 71$$

$$a = 17$$

$$a' = 33$$

$$b = 27$$

$$b' = 13$$

Next Alice computes $c \equiv m^a \pmod{p}$ and sends c to Bob. We'll set m (the message) to 21.

$$m = 21$$

$$c \equiv m^a \pmod{p}$$

$$c \equiv 21^{17} \pmod{71} = 13$$

Next, Bob computes $d \equiv c^b \pmod{p}$ and sends d to Alice.

$$d \equiv c^b \pmod{p}$$

$$d \equiv 13^{27} \pmod{71} = 59$$

Next, Alice computes $e \equiv d^{a'} \pmod{p}$ and sends e to Bob.

$$e \equiv d^{a'} \pmod{p}$$

$$e \equiv 59^{33} \pmod{71} = 35$$

Finally, when Bob gets e , he just has to calculate $f \equiv e^{b'} \pmod{p}$ to get the original message.

$$f \equiv e^{b'} \pmod{p}$$

$$f \equiv 35^{13} \pmod{71} = 21$$

This works because of Eulers Theorem. After all the operations above the result is $f = m^{aba'b'}$. Looking at just the aa' part first we know that $aa' = 1 + (p-1)k$ for some integer k . This is because $aa' \equiv 1 \pmod{p-1}$. This means:

$$\begin{aligned} x^{aa'} &= x^1 x^{(p-1)k} \\ &= x^1 (x^{p-1})^k \\ x^{aa'} &\equiv x1^k \pmod{p} && \text{By Eulers Theorem} \\ &\equiv x \pmod{p} \end{aligned}$$

We can do the same process for b and b' to show that they cancel eachother out too.

Exercise 5

Encrypting a message twice with two different exponents (e_1 and e_2), but the same modulus (n) **does not** increase security. If Eve has the ability to find one of the decryption exponents, which can only happen if she manages to factor n , finding the other is trivial.

Exercise 7

The exponent 1 shouldn't be used in RSA because finding the multiplicative inverse of 1 mod anything is simple, it is just 1. You don't even need to factor n to find d when the exponent is 1. In addition, "encrypting" with an exponent of one will just get you the original message.

The exponent 2 also shouldn't be used. $\phi(n)$ is always even if p and q are odd primes (if either of them were 2 factoring n would be trivial, so they must be odd) because $\phi(n)$ is $(p-1)(q-1)$ and multiplying two even numbers always results in an even number. This means $\gcd(2, \phi(n))$ can't ever be 1, which violates the rules for choosing an exponent e . If we were to use 2 anyway we'd later find that $m^2 \pmod{n}$ wouldn't be unique for every possible m and that we couldn't calculate $de \equiv 1 \pmod{n}$.

Exercise 10

If Alice and Bob both use the same modulus to encrypt the same message, but use different exponents (that are relatively prime), Eve will be able to retrieve the original plaintext if she knows c_a , c_b , e_a , and e_b .

To find the original plaintext Eve first runs e_a and e_b though the Extended Euclidian Algorithm to verify that they are indeed relatively prime and find two integers, s and t such that $se_a + te_b = 1$. With s and t known the plaintext can be found with $c_a^{-s} c_a^{-1} c_b^t \pmod{n}$ (assuming s is negative, you just do the opposite if t was negative).

This works because $e_a s + e_b t = 1$ which cancels out the exponentiation done when the message was encrypted.

Proof (the negative exponents are ignored for the proof, you just need to factor out the -1 (and find the multiplicative inverse) to do the exponentiation on real numbers. We assume e_a and e_b are relatively prime and $e_a s + e_b t = 1$.)

$$\begin{aligned}
m &\equiv c_a^s c_b^t \pmod n \\
&\equiv (m^{e_a})^s (m^{e_b})^t \pmod n \\
&\equiv m^{e_a s} m^{e_b t} \pmod n \\
&\equiv m^{e_a s + e_b t} \pmod n \\
&\equiv m^1 \pmod n \\
m &\equiv m
\end{aligned}$$

Substitute for c_a and c_b
Put the exponents together
Combine both ms
By the Extended Euclidian Algorithm

Example

$$\begin{aligned}
p &= 7 \\
q &= 11 \\
n &= 77 \\
e_a &= 23 \\
e_b &= 37 \\
m &= 4 \\
c_a &\equiv m_a^e \pmod n \\
c_a &\equiv 4^{23} \pmod{77} = 9 \\
c_b &\equiv m_b^e \pmod n \\
c_b &\equiv 4^{37} \pmod{77} = 60 \\
s &= -8 \\
t &= 5 \\
m' &\equiv c_a^s c_b^t \pmod n \\
m' &\equiv 9^{-8} \times 60^5 \pmod{77} \\
m' &\equiv 9^8 \times 9^{-1} \times 60^5 \pmod{77} \\
m' &\equiv 9^8 \times 60 \times 60^5 \pmod{77} \\
m' &= 4 \\
m' &= m
\end{aligned}$$

Found with the extended euclidian algorithm
Found with the extended euclidian algorithm

Exercise 11

With Alice's encryption method there is a very limited range to the plaintext (1-26). Eve can use this to her advantage. She knows each piece of the ciphertext maps to a single character in the range 1-26. There are probably very few exponents that will decrypt each part of the ciphertext to a plaintext value in the range of 1-26. To find the message all Eve has to do is run through every possible d value and find which ones result in possible messages.

The following Haskell expression computes the possible d values:

```
filter (\d -> and
        $ map (\x->x>=1&&x<=26)
        [powMod 8881 c d | c <- [4461,794,2015,2015,3603]]
    ) [1,3..8879]
```

This calculation revealed 1003 and 5349 as possible exponents. They both decrypt the ciphertext to the plaintext "hello"

Question 7

Given the congruencies:

$$x \equiv 12 \pmod{25}$$

$$x \equiv 9 \pmod{26}$$

$$x \equiv 23 \pmod{27}$$

the Chinese Remainder Theorem can be used to find x . x is **14387**.

The following Haskell function was used to compute this.

```
crt :: Integral a => [(a,a)] -> a
crt congruencies = x `mod` mprod
  where
    (a,m) = unzip congruencies
    mprod = product m
    z = map (div mprod) m
    y = zipWith multInvMod m z
    x = sum $ zipWith3 (\b c d -> b*c*d) a y z
```