Blake Johnson
9663980

Homework 3

Task 1:

a) This calculation involves calculating Euler's totient function. This is the process of counting up the number of relative primes that are less than the target number. The relative primes can be determined by taking the greatest common denominator of the target number and the number you are checking, and if the greatest common denominator of the two is 1, then you know it is a relative prime. Below is the pseudocode for my function that calculates Φ(targetNumber). I calculated Φ(2214119) to be 2211144 based on this pseudocode.

```
def eulersTotient(target):
    count = 0
    for i in range(target):
        if(gcd(target, i+1) == 1):
            count = count + 1
    return count
```

b) $d= e^{-1} \bmod \Phi(2214119)$
$d= (5)^{-1} \bmod 2211144$
$5d = 1 \bmod 2211144$
$d = 2211145/5$
$d = 442229 + 2211144*k$ <- the k signifies the other multiples of Φ(N) that would make a valid modulo

The formula for d given N, e is $d= e^{-1} \bmod \Phi(N)$ which means that there is an acceptable d value that when multiplied by e and divided by Φ(N) has a remainder of 1.

c) This is a good way to randomize the ciphertext because with a random message, the encryption will be deterministic of that random value, making it random as well. However if the receiver is not aware of what the value of r is (if it is chosen at random how could they be), then there could be some confusion in determining m from (m*r), making it not a good method.

Task 2:

a) The attacker would have to trick the certification authority into revealing one of its keys, or trick many people into thinking that the attacker was a certificate authority and could create valid signatures on public key, domain pairs. Aside from that, the only thing an attacker could do would be to not forward network traffic which would result in a lack of communication between the server and the client, but would not break the encryption scheme.

b) This is not really a problem because it is difficult to be a trusted certificate authority, and so attackers usually do not end up with signing abilities. Aside from attacking the certificate, there is not much a man in the middle attack under these circumstances can do besides dropping traffic, which is not really a problem for the server because they are not exposing any information to an attacker, and it is something that is out of their control.

Task 3:

a) The first vulnerability is that the MD5 function is inherently insecure because it is prone to collisions.
b) The second vulnerability is that the password can only be 16 bytes long, meaning the passwords aren't that long, and they can be more easily brute forced than if they had the possibility of being longer.
c) The third vulnerability is converting the message to uppercase. This decreases the number of possible different messages, which also increases the probability of collision.
d) There is no salt to this, meaning the passwords of other users are vulnerable if one becomes discovered, and other users have the same password. This function does not take into account salts. The best way to get all the passwords would be to brute force just one and if the hashes of the other passwords are the same then you have the passwords to all of them.