Danmarks tekniske universitet

01410 Cryptology

Cryptology 2

Noah Grænge Surel
s215783

Christopher Zwinge
s204459

Tobias Collin
s183713

16. Marts 2022

Exercise 1.1

```java
public static ArrayList<Long> trialDivision(Long N){
    ArrayList<Long> a = new ArrayList<Long>();
    long f = 2; //Smallest possible factor
    while(true){
        if(N%f == 0){
            a.add(f);
            N /= f;
            a.add(N);
            break;
        }else{
            f+=1;
        }
    }
    return a;
}
```

To do trial division we check every number from 2 and up. If a number doesn't divide N without a remainder, we check the next number. If the number does divide N without a remainder we add it to a list and divide N by the number we found to find the number that the first is multiplied by to get N. This gives us the following values for p and q when we use the function with N

$$p = 3209$$
$$q = 1338413$$

Exercise 1.2

```java
public static long egcd(long a,long b) {

    long rP = a; long r= b;
    long sP = 1L; long s = 0L;
    long tP = 0L; long t = 1L;

    while(r!= 0){
        long q = Math.floorDiv(rP,r);

        long rTemp = rP;
        rP = r;
        r= rTemp -(q*r);

        long sTemp = sP;
        sP = s;
        s = sTemp -(q*s);

        long tTemp = tP;
```

```
19              tP = t;
20              t = tTemp -(q*t);
21
22          }
23
24          System.out.println("1= (" + r+ "," + sP + "," + tP + ")");
25          long sInverted = (sP+b)%b;
26          System.out.println("d =" +sInverted);
27          return sInverted;
28      }
```

We followed the algorithm in the slide to make the implementation. When trying it with the encryption e and (p-1)*(q-1) we got a negative number for d and we therefore added the calculation to find the inverse modulo of the number. Which gave us the following value for d 2321638369.

$$d = 2321638369$$

Exercise 1.3

We used the decryption value to find the plaintext with the following equation.

$$c^d \ mod \ N = m$$
$$17^{2321638369} \ mod \ 4294967317 = 1372559486$$

Which gave us the plaintext found above and we then re-encrypted it with the following equation.

$$m^e \ mod \ N = c$$
$$1372559486^{2^{16}+1} \ mod \ 4294967317 = 17$$

Exercise 2

If the message is multiplied with another message which gets encrypted, it can be seen that this is the same as 2 already encrypted messages being multiplied together. Which can be described with the following equation.

$$enc(m_1 \cdot m_2) \ \ mod \ \ N = c_1 \cdot c_2 \ \ mod \ \ N$$

However the message will change if the combined message exceeds N, therefore the result must be less than N.

With this information we can see that any message that the attacker can encrypt and multiply with the orginal ciphermessage, will when decrypted return the original message multiplied with the chosen message.

$$\frac{m \cdot m'}{m'} = m$$

As can be seen from this equation we can get back the original message through this simple interaction and then from there we have the original message m.

This can then be made into an algorithm A.
First we receive the ciphertext c.
We make a message m' and encrypt that to c'.
We take c $\cdot$ c' and send that back to the challenger to get the combined message mm'.
From there we take mm' and divide by m' to get the original message.
We then return the correct b