Implementacja publicznego systemu kryptograficznego w oparciu o algorytm RSA.

Kacper Matecki 145430

1. **Założenia**
   Program został napisany w języku Python, który nie posiada ograniczeń co do wielkości typu int, przez co wielkość obliczanych liczb jest ograniczana przez pamięć komputera. Największym ograniczeniem w przypadku tego języka jest czas, ponieważ nawet dla małych wartości liczb pierwszych - w poniższym przykładzie 1861 i 1481 - czas trwania deszyfrowania wynosił kilka minut.

2. **Opis metod użytych do wyznaczania e i d**
   Do wyznaczenia **e** sprawdzane są liczby z zakresu 2 - **phi**. Jeżeli wartość jest liczbą pierwszą oraz jej największy wspólny dzielnik z **phi** wynosi 1, to jest ona wybierana.

```
for i in range(2, phi):
    if isPrime(i):
        if (math.gcd(i,phi) == 1):
            e = i
            break
```

   Do wyznaczenia **d** wykorzystywane jest wcześniej obliczone **e**. Iterując od 1, wyznaczana jest wartość 1 + iteracja*__phi__. Jeśli ta wartość po wykonaniu modulo **e** daje 0 to obliczenia są przerywane, a następnie jest ona przepisywana do **d.** Ostatnim krokiem jest podzielenie **d** przez **e** oraz zamiana na liczbę całkowitoliczbową.

```
d = 1
i = 0
while(True):
    x = 1 + i*phi
    i += 1
    if x % e == 0:
        d = int(x/e)
        break
```

3. **Opis realizacji zadania**
   - Dane wejściowe:

```
original_message.txt
1    jakas wiadomosc zawierajaca 50 znakow aaaaaaaaaaa
```

   - Wartości parametrów **p**, **q**, **e**, **n** oraz **d**

```
p - 1861
q - 1481
```

```
e - 7
n - 2756141
d - 2359543
```

- Dane zaszyfrowane (fragment)

```
encrypted.txt
1   738460 1836324 1269034 1836324 1444041 1684662 2336603
```

- Dane odszyfrowane

```
decrypted.txt
1   jakas wiadomosc zawierajaca 50 znakow aaaaaaaaaaa
```

4. **Odpowiedzi na pytania**
   - **Jakie elementy algorytmu są trudne w realizacji?**
     Najbardziej złożonym obliczeniowo elementem algorytmu jest deszyfrowanie wiadomości. Jest to spowodowane koniecznością podniesienia wartości zaszyfrowanej do potęgi **d**, która może osiągać bardzo duże wartości. Dodatkowym krokiem jest przeprowadzenie modulo z **n**.
   - **Co stanowi o bezpieczeństwie i jakości tego algorytmu szyfrowania?**
     Bezpieczeństwo algorytmu RSA leży w trudności faktoryzacji dużych liczb. Mnożenie olbrzymich liczb nie jest procesem złożonym obliczeniowo, ale znajdowanie dzielników tych liczb - tak.

5. **Wnioski**
   - Tworzenie kluczy publicznych i prywatnych oraz szyfrowanie wiadomości nie zajmuje dużo czasu.
   - Nawet dla niewielkich liczb pierwszych pierwszych (około 10000) deszyfrowanie wiadomości zajmuje ogromną ilość czasu.