FRANCESCA GRIMALDI

**Problem 1**

Factorize $n = 275621053$. You can assume that $n = pq$, where $p - q$ is relatively small. Show your calculation steps.

p-q is relatively small → p and q are close together and Fermat's factorization method can be applied

① $a = \lceil \sqrt{n} \rceil = 16602$
$b = \sqrt{a^2 - n} = 73,15052973$

② $a = a+1 = 16603$
$b = \sqrt{a^2 - n} = 196,3568181$

③ $a = a+1 = 16604$
$b = 267,8861699$

④ $a = 16605$
$b = 323,9938271$

⑤ $a = 16606$
$b = 371,7297405$

⑥ $a = 16607$
$b = 414$

$n = (a-b)(a+b) = (16607 - 414)(16607 + 414) =$
$= 275621053$

$p = 16607 - 414 = 16193$
$q = 16607 + 414 = 17021$

The same result is obtained with the program in the fermat.py file.

**Problem 2**

a) Alice wants to set up her RSA encryption with private key $(n, d)$ with $n = pq$, using two primes $p$ and $q$, and private key $d = 3$. She chooses $p = 1283$, but wonders which of the following choices for $q$ she should use (NB! They are all prime numbers):

$$1307,\ 1879,\ 2003,\ 2027$$

Explain why she should use $q = 2027$ for the system to work and to be most secure. For the weak choices of $q$, name an effektive attack to factorize $n$ (of course, these numbers are far too small to be secure, so consider the security in relative terms.)

b) Find the corresponding public key $e$ using the extended Euclidean algorithm. Write a program to do the calculation.

c) Encrypt the message 111 using repeated squaring. Implement the algorithm yourself.

**a)** The system is the most secure using $q = 2027$ because p and q should not be close together, and this value is the furthest from p.
Effective attacks to factorize n for the weak choices of q are Fermat's factorization method and Pollard (p-1).

**b)** we want to find e as the multiplicative inverse to d modulo $(p-1)(q-1)$

$$de \equiv 1 \pmod{(p-1)(q-1)} \rightarrow de \equiv 1 \bmod 2597332$$

From the code in euclid-extended.py file, $e = 1731555$
Verification: $ed = 1731555 \cdot 3 = 5194665 = 2 \cdot 2597332 + 1$
$\equiv 1 \pmod{2597332}$ ✓

**c)** message: $111_2 = 7_{10}$
encryption function: $C \equiv M^e \pmod n$, in our case

$$7^{1731555} \bmod 2600641$$

($1731555$: we apply repeated squaring so that we reduce the number of multiplications; $2600641$: $n = pq$)

Using the functions in rsa_repeated_squaring.py file to find the powers of 2 that add up to 1731555 and then compute the exponentiation of 7 (mod n), the resulting encrypted message is 2358372.
In the program, I also verified this solution by evaluating the decryption function $M \equiv C^d \pmod n$, which gives 7 as result.

**Problem 3**

a) Let $n = 1829$ and $B = 5$. Find a prime factor of $n$ by using Pollard $(p-1)$ attack.

b) Let $n = 18779$. Using Pollard $(p-1)$, how small $B$ can be used for the attack to be successful (Use knowledge of the factorizations of $n$.) You do not need to find the facotirzation.

**a)** $B = 5$

$A = 2^{5!} \pmod{1829}$

↳ starting from a generic "a" > 1

$A = 2^{5\cdot4\cdot3\cdot2} \pmod{1829} \equiv 311$

↓

$2^2 \bmod 1829$

$4^3 \bmod 1829$

$64^4 \bmod 1829 \equiv 1628$

$1628^5 \bmod 1829 \equiv 311$

$\gcd(311-1, 1829) = 31$

using Euclidean algorithm

$1829 = 5 \cdot 310 + 279$

$310 = 1 \cdot 279 + 31$

$279 = 9 \cdot 31 + 0$

So 31 is a prime factor of 1829 (same result is obtained using pollard.py program).

**b)** $n = 89 \cdot 211$

we prime factorize $89-1$ and $211-1$

$88 = 2^3 \cdot 11$

$210 = 2 \cdot 3 \cdot 5 \cdot 7$

The best choice is $B = 7$ as it is a relatively small number and allows to find the prime number 211 with $a = 2$, therefore doing less calculations.

**Problem 4**

a) Show that encryption in RSA has the following property:

$$e_K(x_1)e_K(x_2) \mod n = e_K(x_1x_2) \mod n$$

b) Show how RSA is vulnerable to **chosen cipher text attack**: For ciphertext $y$, then Eva can choose some $r \not\equiv 1 \mod n$, and construct $y' = y \cdot r^e$. If she then knows the decryption $x' = d_K(y')$, show how she can calculate $x = d_K(y)$. (Hint: She can also calculate $r^{-1} \mod n$)

**a)** $e_K(x_1) \cdot e_K(x_2) \mod n = x_1^e \cdot x_2^e \mod n$

$e_K(x_1 x_2) \mod n = (x_1 x_2)^e \mod n = x_1^e \cdot x_2^e \mod n$

↑ LAW OF POWER OF A PRODUCT $(ab)^m = a^m \cdot b^m$

Therefore $e_K(x_1) \cdot e_K(x_2) \mod n = e_K(x_1 x_2) \mod n$

**b)** $x' = d_K(y') = (y')^d \mod n = (y \cdot r^e)^d \mod n =$

$= y^d \cdot r^{ed} \mod n \equiv y^d \cdot r \mod n$

↑ for the CHINESE REMAINDER THEOREM

Knowing $r^{-1} \mod n$, Eva can obtain $y^d$

$y^d \cdot r \cdot r^{-1} \equiv y^d \mod n$

↖ what we were looking for

$x = d_K(y) = y^d \mod n = d_K(y') \cdot r^{-1} \mod n$

**Problem 5**

Alice and Bob want to have an common key using Diffie-Hellmann key exchange. They agree on using the prime 101, and base $n = 3$. Alice choosed her secret $a = 33$, and Bob chooses $b = 65$.

a) Write a program that prints out all the powers $3^i$ for $i = 1, ..., 100$. Do the same for $5^i$. What is a major difference between these two sequences?

b) Find their common key.

**a)** Looking at the powers of 3 and 5 it is possible to notice that those of 5 grow faster (being 5 a bigger number), and are also more easily recognizable, as all the numbers end with the digit 5.

(program: powers-table.py)

**b)** Alice sends Bob $n^a \bmod p = 3^{33} \bmod 101 = a_1 = 61$
Bob sends Alice $n^b \bmod p = 3^{65} \bmod 101 = b_1 = 62$

Alice calculates $K = b_1^a \bmod p = 32$
Bob calculates $K = a_1^b \bmod p = 32$

$K = 32$ is the common key

These results were obtained using the program in diffie-hellman.py and computing the exponentiations with repeated squaring.