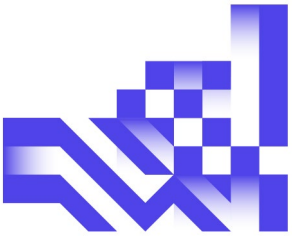
Hogeschool van Amsterdam

Discrete Mathematics

Practical
Assignment 3

Table of Contents



1 Assignment

In this assignment, you will apply your knowledge of encryption and decoding messages to develop one program that is able to:

- encode a message (variable m) given a product of two primes (variable n)
- decode a given message (variable c) with the public key (variables n and e).

More information about the requirements can be found in the section “Delivery and Grading”.

2 Environment set up

2.1 Installation

You will write your program in Java version 8, since this is the language you are used to working during your study. For this assignment, you need to make one jar file named ***pa3.jar*** which can be executed on a machine which does not have a full JDK installation.

2.2 Usage

Your program should open a window with two sections to select which function will be executed: encryption or decryption.

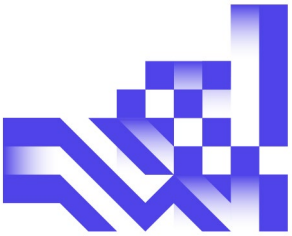
2.2.1 Encryption

The first task is to find p and q based on the given n. They are necessary for the encoding process because you need to calculate a suitable e. Then, a given message can be encrypted.

Step 1: Calculating p and q
The user should be able to input the value n and click on a button called “Step 1” to find p and q.
The program should print the following elements on the screen:
p is <value of p found>
q is <value of q found>
Amount of time busy finding p and q: <amount of time in milliseconds>

Step 2: Calculating e
The user should be able to generate a suitable e by clicking a button called “Step 2”.
The program should print the following on the screen:
e is <value of e found>

Step 3: Encrypting the message
The user should be able to input the value m and click on a button called “Step 3” to encrypt the message. Then, the program should encrypt the message and print it on the screen as follows:
Message after encryption is: <c>



2.2.2 Decryption

The first task is to find d based on the given n and e. It is necessary for the decryption process. Then, a given message c can be decrypted.

Step 1: Calculating d
The user should be able to input the value n and e click on a button called “Step 1” to find d. The program should print the following elements on the screen:
d is <value of d found>

Step 2: Decrypting the message
The user should be able to input the value c and click on a button called “Step 2” to decrypt the message. Then, the program should decrypt the message and print it on the screen as follows:
Message after decryption is: <m>
The decoded message should be text.

3 Resources

Text Book: Discrete Mathematics with Applications, International Edition
Chapter 8.4

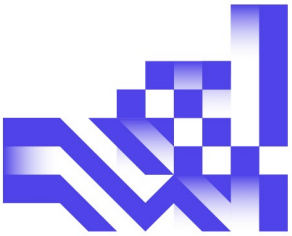
4 Delivery and grading

4.1 Delivery

You hand in a zip file that contains:

- The source code of your program
- A .jar file called pa3.jar for the program
- The pdf file, named manual.pdf. It contains information about your program and instructions for testing

The name of the zip file should have the following form: 03_IntialName1InitialName2.zip where Name1 and Name2 are the surnames of the students who hand in the practical assignment. Rules?



4.3 Grading

See below the grading rubric for this assignment.

Passing grade (5.5) Minimal requirements	+1 Organisation	+3.5 Extra
The executable pa3.jar starts correctly. Encoding works correctly (step 1, 2 and 3 as described in the assignment). Decoding works correctly (step 1 and 2 as described in the assignment). Encoding: It is possible to enter different message m and p and q for encoding. Decoding: it is possible to enter different cypher and e. Step by step description of the generated results while encrypting the given message in the manual (value of given m, n) and de encoded message. A step-by-step process, describing the generated results to decrypt the given message in the manual (value of given cypher and e).	Zip file that contains the source code and pa3.jar. The manual is named manual.pdf. There is a clear distinction between the encoding and decoding process in the user interface.	A prediction with rationale on how much time it would cost to find a larger p and q (extra section in the manual) Description of the Big(O) for this algorithm based on different (increasing) p and q's, including graphs (extra section in the manual) The user interface is able to validate the input of e Option to generate several e's. The program is prepared to deal with big numbers (>32bits)