Assignment 2: Implementing and Breaking RSA!

Project Objectives:
1. Apply security concepts you study in the course.
2. Enhance student's understanding of encryption algorithms.
3. Have experience with code breaking.
4. Implement the RSA algorithm, one of asymmetric key encryption algorithms.

Requirements:

1. Implement the RSA algorithm (encryption and decryption).
2. Implement a simple program sending and receiving a text message, the sender should encrypt the message using the receiver's public key, then the receiver will decrypt it using his own private key and display it. Your program should support any text message containing alphabetical and numeric characters, not only numbers (all ascii characters).
3. Try to use different key lengths for RSA (in terms of the size of n), and calculate efficiency in terms of encryption time using RSA for each key length, plot a graph of RSA encryption time vs. Key length.
4. Implement brute force (mathematical attack) on RSA algorithm using different values for n,plot
a graph of Time to break the private key (in seconds) versus value of n. Discuss the results you Obtain.
5. Your code should be as robust as possible, as I will try to break it one way or another.

Important Notes
• Implement all the required algorithms (RSA, and breaking the code) and all the tasks by yourself.
• The sender and receiver should be two separate modules (2 programs), Communication can be done using files, or network (socket programming).
• Use any programming language of your choice such as: Python, Java, C++, C#,..etc.

Deliverables:

- source files
- Sample input files test cases that were used for testing.
- 2 graphs (one is encryption time against different key lengths, the other is brute force times against different key lengths)
- analysis results and your conclusions from the Chosen Cyphertext Attack.
- All projects will be tested by another set of data.

Project Due Date: 14 / 8, next monday, discussions will be via online meets
Assignment can be done in Pairs, it is better this way. Both students will be discussed.

Chat Requirement

Inputs:
One text file with 2 numbers separated by a new line, the first number is p, the second is q

Outputs:
The sent message should be displayed at the receiver WITH the encrypted message.

يعني اشوف الرسالة المتشفرة و الرسالة الأصلية هناك في الريسيفر

The receiver begins first to calculate his public and private keys, then by any means, sends the public key to the sender.
“By any means” can be either writing to a text file or sending an initialization message carrying the public key (it is okay if everyone saw the public key)

Graphs Requirement

Graph 1:
A plot of the taken encryption time against different key lengths n.
Remember n = p * q
It is okay to try big key lengths, don’t worry it will not take hours, maybe minutes.

Graph 2:
A plot of the taken time to brute force different key lengths.
It is expected that the time will increase massively with larger n. So you might consider using a small range of lengths, for example n = 3, 4, 5, 6, 7,....,64, depending on your machine, you can increase the max key length as you wish as long as the final plot is informative.

Length of n should either be in number of bits, or number of digits, you can decide.

You should expect that the brute forcing times of greater lengths to take hours, as a threshold, this plot should take on average 4 hours, leave it overnight after making sure everything works as intended by using small numbers first.

Disclaimer note:
This assignment was adopted from the cyber security course project you will attend during your second semester.

A bonus point will be given to any of the following:

- A nice GUI
- Two way chat (with 2 txt files to compute 2keys)

- Using hashcat or john the ripper or hydra (any encryption breaking tool) to brute force the key, hashcat/others should be called inside your code, not doing it manually.