BSPro - A First Bachelor Semester Project in BiCS-land

Saturday 23rd January, 2021 - 15:44

Issam Jomaa
University of Luxembourg
Email: issam.jomaa.001.student@uni.lu

Răzvan Roşie
University of Luxembourg
Email: razvan.rosie@uni.lu

Abstract

This paper presents a summary of 1000 words approx, in English for the first Bachelor Semester Project made by Issam Jomaa under the direction of his tutor Răzvan Roşie. for more information the reader is invited to read the main report.

1. Introduction and Objectives

The project aim is to create a program that creates and solves time Lock Puzzles at the same time based on the RSA scheme. While the program is solving this Time lock puzzle it should record data related to the time taken. This program will be written in both C++ and Python. Another program should be made to plot the data and render good graphs, from which we can draw some assumptions and conclusions about the performance of our main program. This second program is written in Python. Throughout this project, our aim is first to get a solid understanding of the world of cryptography and the different Cryptography methods that exist. Then a second objective would be to get a deep understanding of the RSA scheme so that we can tweak it and base our program on it. To write our program and since this a first semester Bachelor project we also will need to learn and improve our coding skills in both c++ and Python. At the end of the project using our second program, we should be able to draw some conclusions about the performance of our program in general but also when it comes to the difference between the one written in Python and the one in C++.

2. Background

The domain of this project is Cryptography and it revolves especially around one of the 3 main types of cryptography: **Public key Cryptography** also called **Asymmetric Cryptography**. It is a way to send data securely by encrypting it. This system uses a pair of keys that have different functions.

- A public key which is known by both the sender and a receiver.
- A private key that should only be known by the receiver.

How this works is that the Sender will use the public key known by both the users to encrypt the data through a mathematical algorithm. Once the data is encrypted it can only be deciphered using the private key. This means that a regular user needs only to protect his private key that he doesn't need to send over through the internet meaning a less likely chance for it to be hacked. Time lock puzzles are essentially based on Asymmetric Cryptography, the one we will focus on in this project is the RSA scheme. A time lock puzzle is supposed to take a message and cipher it in a way that it is only possible to decipher after a specific time that the creator of the puzzle will setup. In this way, Time Lock puzzles can have multiple Use in real life especially in the Domains of finances and on the internet globally.

Finally, we will talk about the languages we will use to write our programs. "Python is a high-level, interpreted, interactive and object-oriented scripting language" [2]. Compared to other programming languages, Python stands out as being highly readable thanks to it using more English words compared to other programming languages that would use often more punctuation as syntax. It is also considered in general an all-purpose language. One important part is that it features dynamically allocated memory.

C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming. C++ is regarded as a middle-level language, as it comprises a combination of both high-level and low-level language features. [1] C++ doesn't allocate memory dynamically.

3. Requirements,Design and production

Our first requirement globally is something pretty basic which is to write a professional program, which means our code should be clean, as simple as possible, and well commented. This assure that we are doing professional work with good industry standards. Another requirement would be that our program should be modulable so that our program is easy to modify/maintain/debug and also so that we can reuse these functions in future works if necessary. Our main program should be able to generate prime numbers and the keys for our time lock puzzle fast, for c++ since we won't use specialized libraries, we will have to create all the functions needed, especially the one that deals with randomness and prime numbers. First, since we need to use big variables for our program we will use the GNU library focusing on the mpz_class.The randomness will also be achieved using this library. The generation of primes will be done using the Miller-Rabin primality test while adding a few options to assure its liability. This function will be called $generateprime()$ and will use other functions we created called $fonctionrandom()$ and $isPrime()$ which is the Primality test. Once the key is created we will have a function called rsw that will solve this puzzle. While solving the puzzle, it should be able to record the data and save it into a file in a good way. Our second program should be able to create graphs out of our data file. The Python program will have the same pattern but we will use functions taken from Libraries to generate random prime numbers... Our second program should be able to use the data file and plot graphs that will help us analyze the performance of the main program.

4. Conclusion

After running our program for a long time and obtaining data files from both main programs, we create graphs using the second program. From this graph, we have found that the c++ program was faster than the Python program by a big margin when it comes to solving the time lock puzzle. We have deduced the fact that this could be the result of the way data is stored in these programs and how Python dynamical storage might be the reason the program is taking more time. We also observed that the parameter t had an exponential influence on the time taken to solve the puzzle as the key-size became bigger and bigger.

References

[1] *Cpp_overview*. URL: https://www.tutorialspoint.com/cplusplus/cpp_overview.htm (visited on 12/11/2020).

[2] *Python_overview*. URL: https://www.tutorialspoint.com/python3/python_overview.htm (visited on 12/11/2020).