|  |
| --- |
| Name:  Ratnayake K Basura N B |
| Student Reference Number:  10541973 |



|  |  |  |
| --- | --- | --- |
| Module Code: **SOFT336SL** | Module Name: **Cross Platform Application Development in C++** | |
| Coursework Title: Assignment - User Documentation | | |
| Deadline Date: **23/12/2015** | | Member of staff responsible for coursework:  Dr. Davide Marocco |
| |  |  | | --- | --- | | Programme:  **4872/3** |  | |  |  | | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.  ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***  Signed on behalf of the group: | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***  Signed: | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |



**Degree:** BSc (Honours) Software Engineering

**Stage:** 3

**Batch:** 14.2

**THE**

**FORE RUNNERS**

Believe everything is possible

**INDEX NUMBER FULL NAME**

10541973 Basura Ratnayake

Rahasa 1.0 – Customizable Encryption

User Documentation

Table of Contents

[Description of the Software 1](#_Toc470293270)

[Scope/ functionality of the Software 1](#_Toc470293271)

[Desired Improvements 2](#_Toc470293272)

[Supported platforms 2](#_Toc470293273)

[Installation 3](#_Toc470293274)

# Description of the Software

The purpose of this software is to encrypt file(s) specified by the user. The program creates a pair of prime numbers in the range of 100000 to 900000 as encryption and decryption keys, once a file is encrypted the user can only view it using the program and the password specified by the user. Software architecture is developed entirely based upon OOP concepts, the program is designed for both Windows and Linux based OS but only tested on Kali Linux. The GUI design is kept simple, yet provides a very friendly experience to the user

The main purpose of implementing such a scenario was to gain first-hand knowledge in using cross-platform implementation and file encryption using C++, and to identify the unique characteristics provided by the language and Qt to make maximum use of this style of implementation.

# Scope/ functionality of the Software

1. **Encrypt files with a password.**

User can select a file and encrypt it using a password, without the password the file will not get decrypted.

1. **Limit number of views of a file.**

User can specify a maximum amount of views for a file and if exceed the file will get corrupted and never be able to view again.

1. **Impose a valid duration for the file.**

User can specify a duration the file can be view and if the duration is passed then the file gets corrupted upon open.

1. **Decrypt encrypted file.**

By providing the password that was used in encryption the user can decrypt the file.

1. **View decrypted file.**

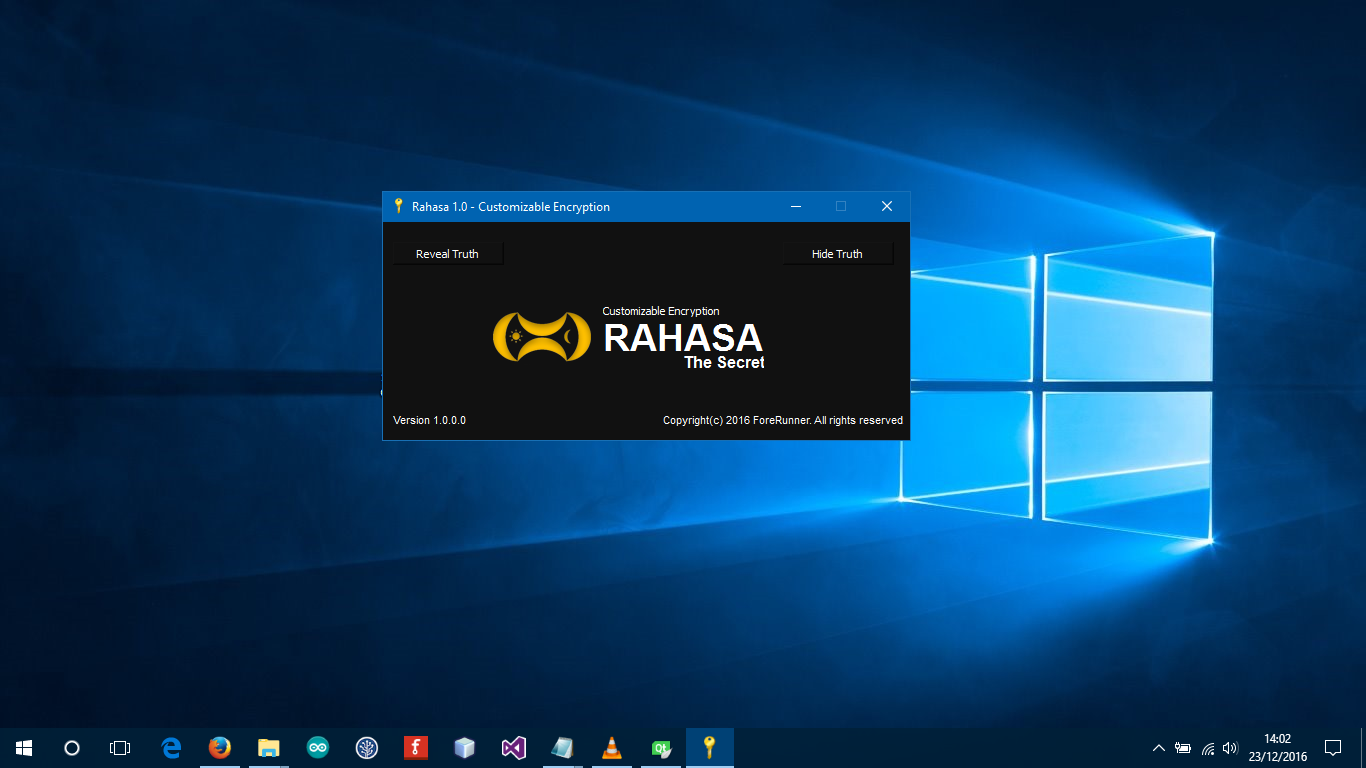
After successful decryption the file can be viewed using the software but only through the software.

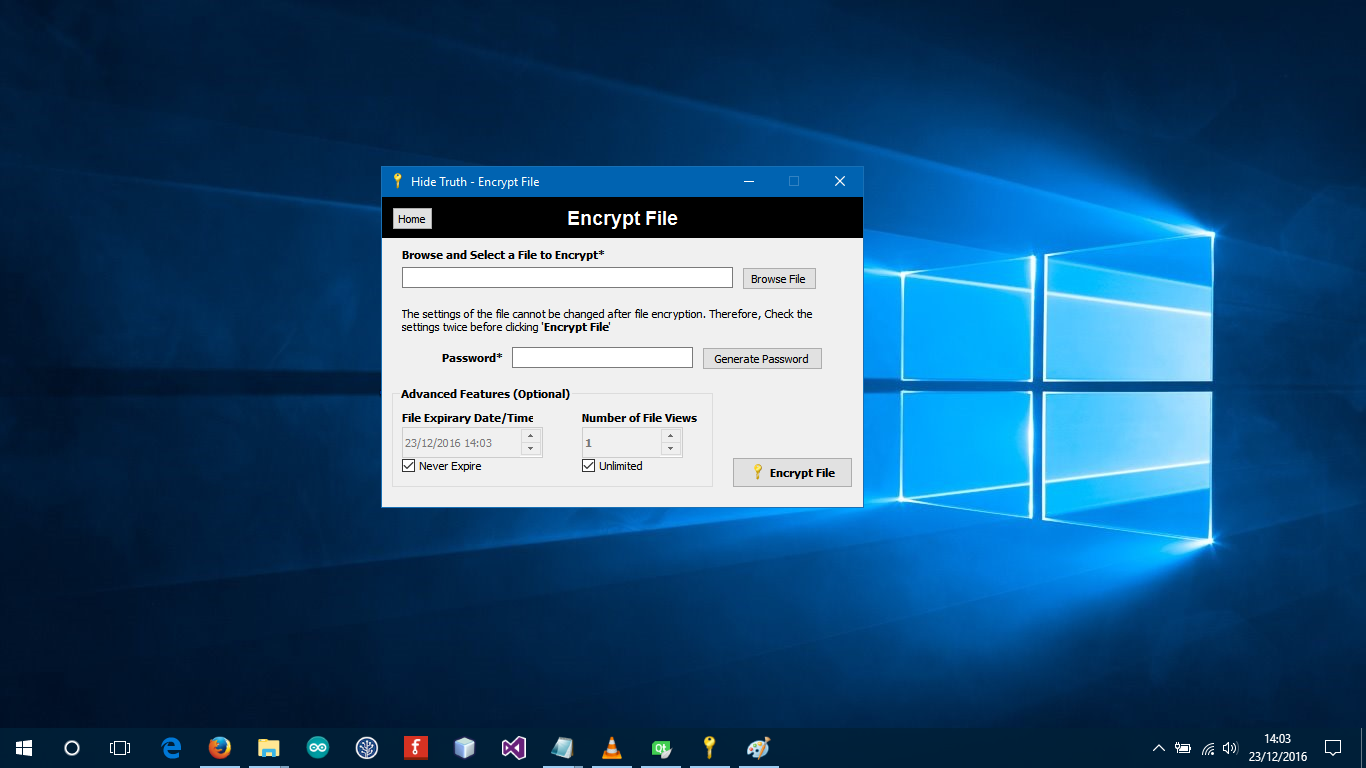
# Desired Improvements

* **RSA Encryption,** would have greatly increased the encryption level of the files and would prove to be nearly impenetrable.
* **Hide in File,** approach is to hide the encrypted file inside a normal image, so anyone who view the file will see the image not the encrypted file. In simpler terms its *‘Hiding in Plain Sight ‘.*
* **Encrypt Any File,** provides a way to encrypt files of any format and any size.
* **Compress File,** reduce the size of the file to enable faster upload to email and preserve memory capacity.
* **Email the File,** after a file is encrypted the user can email it.

# Supported platforms

The application has been tested and run on both Windows 10 64-bit and on Ubuntu. Not tested on any Mac OS but expects to run without any problems in Mac OS X as well.

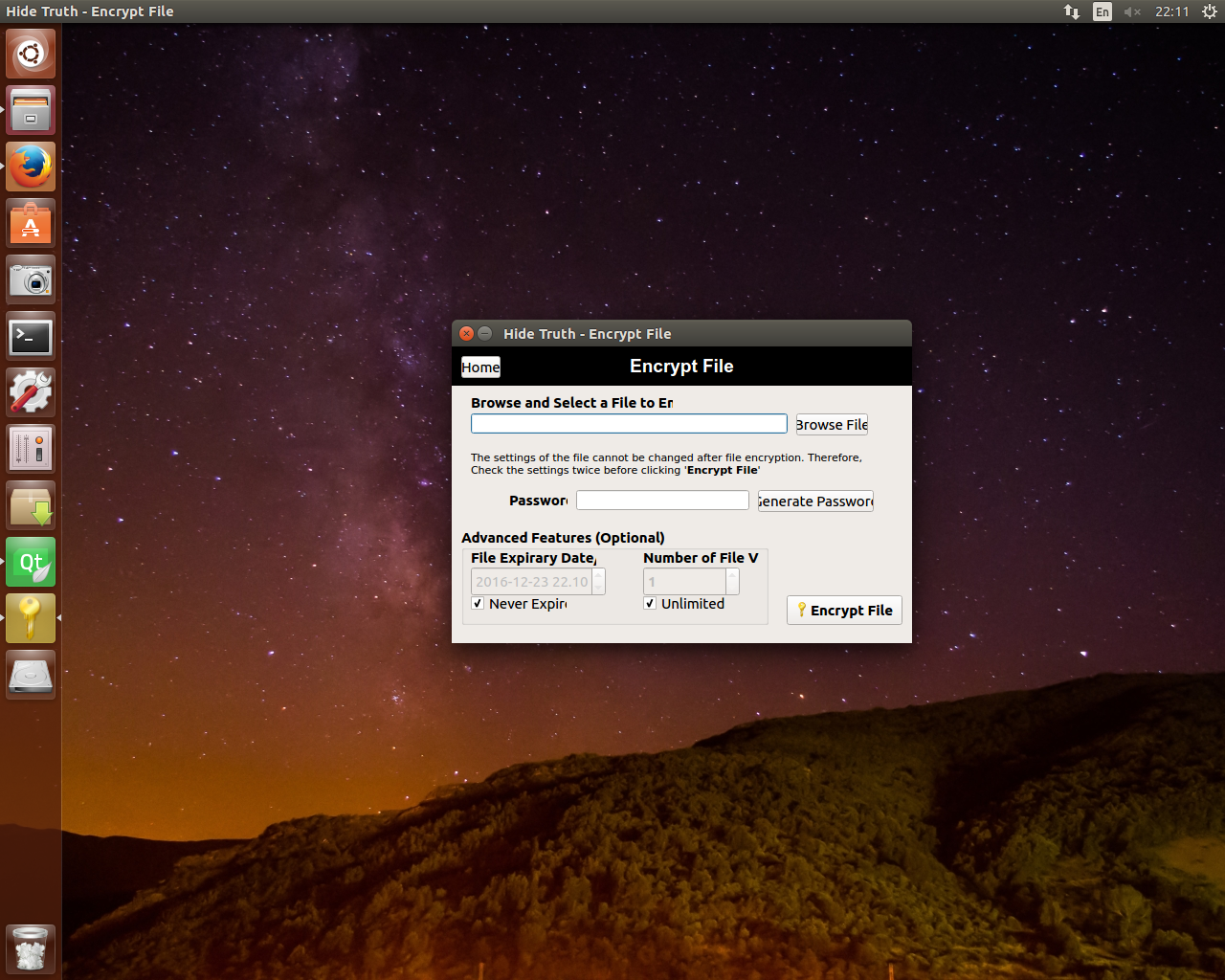




Screenshot: Windows 10 (64-bit)



Screenshot: Ubuntu (64-bit)



Screenshot: Ubuntu (64 bit)

# Installation

The application can simply be built and run using Qt creator by loading the .pro file to it, and compiling using the application.