

Encrypt & Decrypt Text App

Secure Programming exercise work

Aliisa Nissinen

Secure Programming course

Tampere University

1. Introduction	1
2. Program design, implementation and operation	1
2.1 Dependencies	2
2.2 Interface Design and UML Diagram	2
2.3 User Stories	4
2.4 User Manual	5
3. Testing	6
4. Program security	6
5. Improvement ideas	7

1. Introduction

This exercise work was implemented in Tampere University's Secure Programming course. The purpose of the exercise work was to learn more about programming security and to implement something related to the topic.

I implemented a computer program that encrypts and decrypts messages, that can then be sent to other people. The program can be used once it has been downloaded to the user's desktop, so only people with the program can write or read secret messages. Program is used with login credentials, so no one else can use the program without the permission of the desktop owner.

This program is related to the topic because it uses user login and cryptography. Also, this program is designed with help of user stories, that are also related to the secure programming life cycle.

Throughout the program life cycle, the focus was on secure programming. At the end of this document, the program is evaluated with different testing programs and programming guides.

2. Program design, implementation and operation

This program is made to encrypt and decrypt messages. All cryptography operations are done in the same window. The program can be used with pre-acquired credentials.

The making of the program started with a raw interface design and inventing different user stories. These enabled the actual programming to begin.

2.1 Dependencies

C++ programming environment, QML, Crypto++ library


2.2 Interface Design and UML Diagram

The raw user interfaces that were designed before the actual programming are shown below.

Log In to Encrypt & Decrypt Program

User name:

Password:

Encrypt & Decrypt Program User name 

Info box

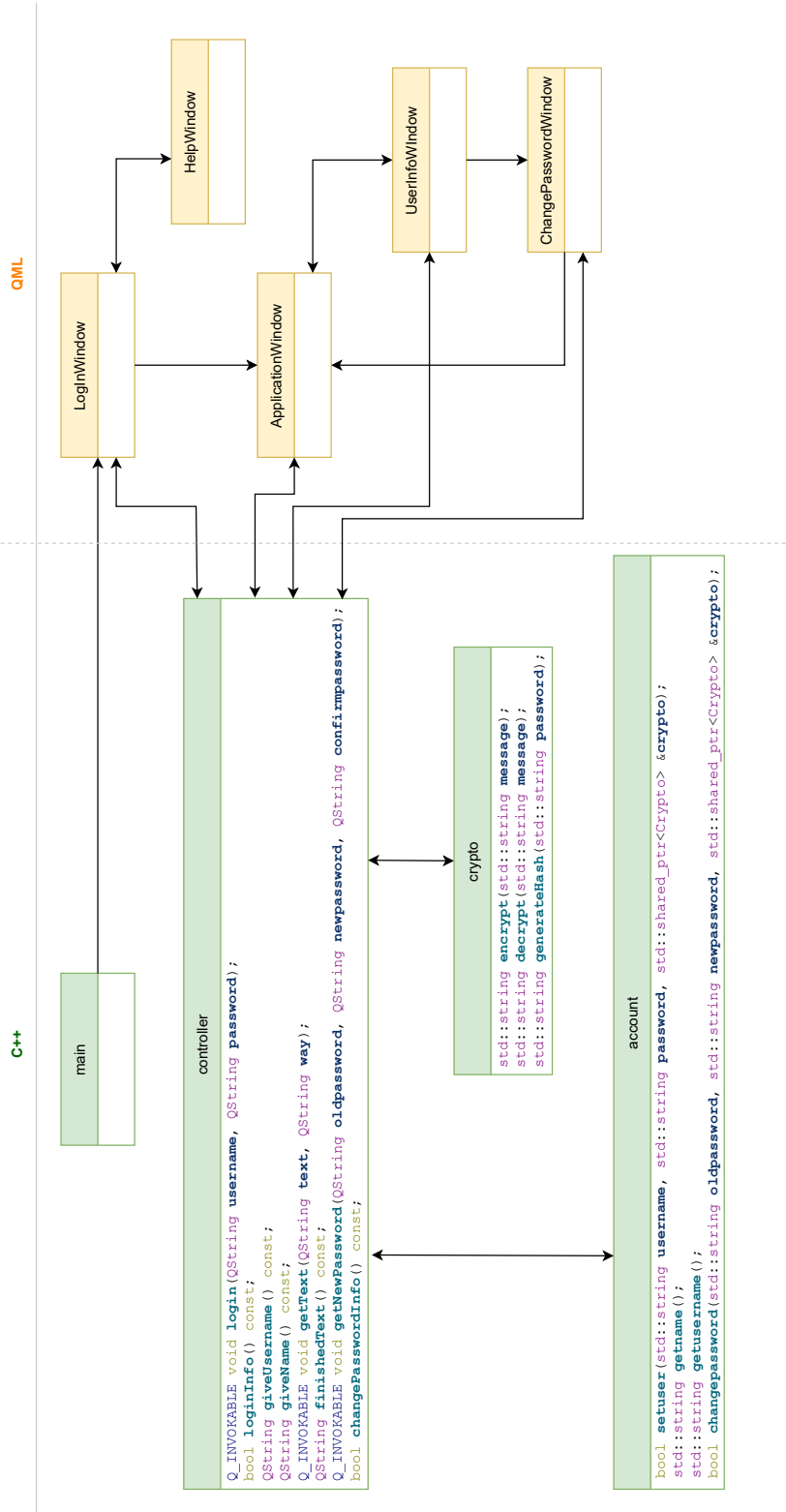
Enter the text to encrypt:

Encrypted text:

Enter the text to decrypt:

Decrypted text:

Below is the UML diagram of the finished program. Later, a new parameter (std::string userkey) was added to the encrypt and decrypt functions of the crypto class, and private function checkPassword was added to the account class.



2.3 User Stories

	Who	What	Why	Done
1	User	Wants to login.	To use the program.	X
2	User	Wants to have safe login.	So no one can steal login credentials.	X
3	User	Wants to change password.	To keep the password secure.	X
4	User	Wants to change username.	To have new username.	
5	User	Want to change name.	To have new name.	
6	User	Wants to encrypt a message.	So that it can be sent encrypted.	X
7	User	Wants to decrypt a message.	So that it can be read.	X
8	Software	Wants to protect the data.	So no one can stole the login informations.	X
9	Software	Wants users to have secure passwords.	So no one can stole the login informations.	X
10	User	Wants to get forgotten login information.	To be able to use the program despite forgetting.	
11	Software	Wants that sessions are timed.	So that no one can use the program if it was forgotten open on the desktop.	X
12	User	Wants to logout.	So session forgets credentials and close the program.	X
13	Software	Wants strong encryption.	So no one can break the encryption.	X
14	Software	Wants reliable decryption.	So that the user can read the message.	X
15	User	Wants to contact the software owner.	To have help with problems in the program.	X
16	Software	Wants that user cannot write commands to the text window.	To prevent the program from being misused in a malicious way.	
17	Software	Wants that all used libraries and platforms are secure.	So that those don't bring vulnerabilities into the program.	
18	Software	Wants that user cannot change texts in the program window.	So that the program is original.	X

2.4 User Manual

First, user has to download the necessary dependencies and run the program in the right environment. After that user needs to log in with a username and password. When user has successfully logged in, he/she can use the program in the one window that is opened after.

User's encryption and decryption key need to be added under the infobox. The Key needs to be 16 bytes long (16 string characters), so it works correctly. How the secret key is sent to other parties is not taken into account by the program.

To encrypt a message, the text needs to be added to the left box of the encryption section, and then the encrypted text will appear on the right side.

To decrypt a secret message, the text need to be added to the left of the decrypt section, and the decrypted text will appear on the right side.

User can also use copy/paste methods from the keyboard. From the username section, user can see the user information and change the password. Password needs to be strong, so it needs to be 8 characters long, have a lower and uppercase alphabet, digit, and special character.

User can see information about the progress of the application from the application output (QDebug messages).

Program can be tested with username *user1* and password *Password123-*.

3. Testing

- QML profiler OK
- Valgrind Memory Analyzer OK
- CppCheck OK
- Clang-Tidy and Clazy OK
- Manual testing OK

4. Program security

Critical points for security are login, saving credentials in the txt file, encryption, decryption, and password change.

The encryption and decryption of messages are implemented with the Crypto++ library and user's own key. This library is a free open source library for C++ and can be loaded from the internet (www.cryptopp.com). This program is using the newest version of the library, Crypto++ 8.6.0., that is released on 24.9.2021. Crypto++ is widely used and has many cryptographic schemes, so it can be assumed to be safe.

User credentials are saved to the credentials.txt file and passwords are saved hashed, so no one can recognize them. The hash method is also from the Crypto++ library. The hashing algorithm is SHA512. Password is validated to be strong enough in the checkPassword function, that is in the account class.

Always when the user writes the password to the application window, the letters have been replaced by dot marks, so outsiders can't see it.

Application has been done according to the SEI CERT C++ Coding Standard rules. Attention has been paid to preventing memory leaks and buffer overflows, and good programming practices.

The program is to be updated to stay as secure as possible, when a new version of Crypto++ has been released, it will be updated to the program.

5. Improvement ideas

- User can change the username
- User can change the name
- The user can somehow recover the forgotten credentials
- Some characters are forbidden in text fields, so those are ignored
- The security of the libraries needs to be checked
- It is not necessary to use classes in the code
- Check if the hashing with SHA512 is the best option
- Save credentials.txt file with password
- Implement registration to the application safely
- Unit testing
- Some kind of login monitoring