Documentation

Nowadays, people always have to keep an account in various websites due to the development of multimedia. Thus, some people who have bad memories may have the trouble to memorize all the accounts and their corresponding passwords. Keeping a notebook is viable but not safe enough. Based on this situation, our group create a simple programming called ***Password Defender*** to help people record accounts and passwords safely. This documentation will give a brief introduction to the features of ***Password Defender*** and show how the important features are implemented.

outline

Overview

Password Defender is a simple programming which allows users to login using a master password, and store their own login names and passwords subsequently. Users are able to delete and add accounts, as well as their corresponding passwords. All the login names and passwords will be encrypted using RSA and stored locally. Hence, it's rather difficult for those one who are not familiar with programming to steal those contents (Although the RSA encryption in our programming is relatively simple and unsafe). Then the confidentiality of users can be protected.

Login window

At the first time of opening the login window, there is no master password by default. In other words, the user can login directly by just pressing the "login" button. The user will be asked to set his/her own master password as soon as possible for the safety issue. However, if the user doesn't add a new password, he/she can still access without an master password when the login window is reopened. If the user set one already, he/she are supposed to enter the right password to access next time. Users are able to modify their master password whenever they want.

List window

The added accounts will be arranged in the ascending dictionary order in the list window. All the added accounts will be presented below one of the 27 labels (A~Z, #) according to its initial character. In this window, users can click the account name to view or delete the account in the edit window. Users can click the ***setting button*** to change the master password (as described in the *login window part*), and ***New Account***

button to add a new account and its corresponding password.

Data structure

All the accounts added to ***Password Defender*** will be stored in an **AVL Tree**, using the account name as the key value. The feature that **AVL TREE** is balanced could reduce the searching time sufficiently when abundant accounts are added.

Search bar

Users can enter the prefix string of an account name to search for the record of that account. All the accounts, whose account names contain a prefix string of that entered string will be shown on ***search button*** when being clicked. The search function is based on the **AVL Tree** structure, so that the searching speed can be reasonable when there are massive accounts.

Add window

In this window, users are permitted to add new accounts. The entered passwords will be shown as dots initially for safety issue, and they will be shown as text after the reveal button is clicked.

Record manager

When closing the list window, which means that this application is closed, all the records stored in **AVL Tree** will be output into a vector of QString objects. Then, the password part will be encrypted with **RSA** into a long string composed of numbers. All the records will be preserved in a .txt file in this process. When opening this file again, it will be read and decrypted to build a new **AVL Tree**.

RSA crypto system

This feature is based on an **InfInt** class, which stores big integers and supports almost all operations. It uses a group of static keys and fast power algorithm. The encrypt and decrypt system are implemented as well.