CSCE-110 Fall 2019: Term Project
The Safe Box

Overview

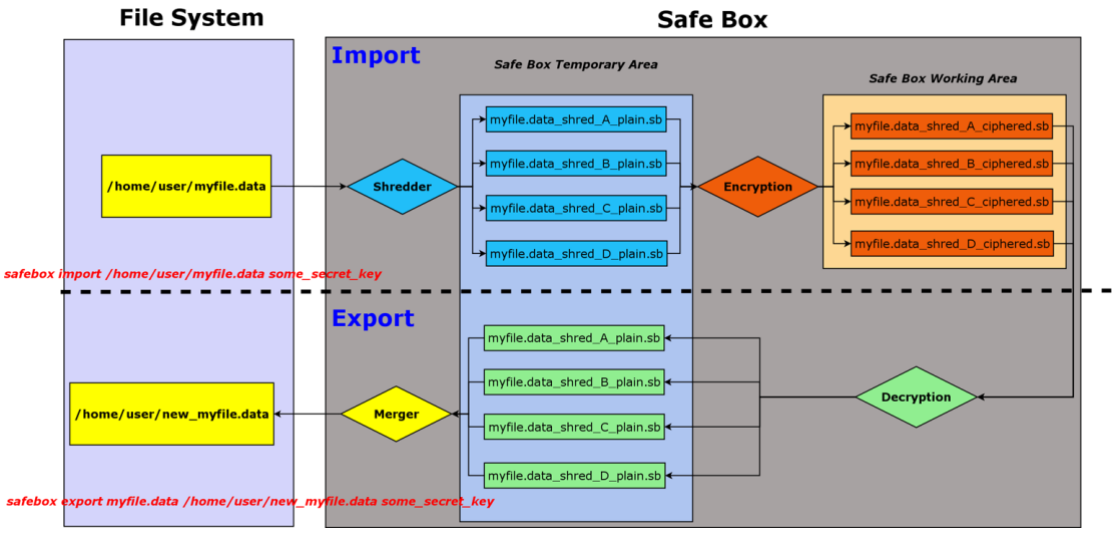
The **Safe Box** is a virtual storage box that aims mainly at securing the content of files stored in it making it unaccessible to unauthorized access. Typically, a computer has a local hard drive, and based on the operating system installed on it a compatible file system is installed on the local drives to allow storing files on them; in the simplest form, each local drive has its own file system. Without a filesystem being installed on the local drive, you cannot store data as files, rather data can be stored only as disk sectors. The filesystem is responsible for organizing your data as files and directories. The disk is organized by the file system as disk blocks, each is one or more disk sector, and data is then stored in disk blocks and some filesystem data structure need to be in place to relate a file to its corresponding disk blocks. If someone have access to your computer, or even can disconnect your hard drives, she/he can have access to all your files and can view them if she/he can identify the filesystem installed, which is a fairly trivial to guess and identify.

What we want to do is to allow the user of a computer to be able to store some of her/his files in a format that cannot be read by anyone except through a secret key (passcode) that is only known by the computer owner. So even having access to the disk and the computer, the content of the files cannot be retrieved. This can be done at the file system level, as there are some file systems that support encryption, but it will need to be done at the operating system level, and most probably will encrypt the whole file system, and all the encryption/decryption will take place by the underlying file system without the user even notice that. The encryption passcode is usually stored encrypted using the user account password. Of course, this can entail some performance overhead, but still the common way to secure data on a file system. On the other hand, it can be done at the user level by having a program that can encrypt the content of selected files by the user, and store them as normal files in a specific place. In that case, even if an intruder have access to those files, the content cannot be made readable except when having the passcode that the files were encrypted with, and of course the program that was used to encrypt the target files.

The Safe Box software, that you will be designing and implementing in your project, is a user level file encryption tool that aims at converting files into an unreadable format through encryption. The Safe Box uses a dedicated directory, we will call it the working directory, on your filesystem to store encrypted files. The way Safe Box works is through two main commands,

import and export. The import command takes a file that is stored in your file system in a readable format, shred it into fragments, encrypt the fragments, and finally store them distributively over many files in the working area. The export will perform exactly the opposite, upon invoking export on a specific file that exists in the working area, Safe Box will locate all the file fragments, decrypt them into its original representation to make the target file readable, and store it back on the file system at a designated location provided by the user. Of course, both operations require that the user provide a valid encryption/decryption key.

The import/export commands, are the main fundamental building block of the Safe Box. But practically, the SB acts as a file shelf that can store more than one encrypted file, and is responsible to manage all the files in the shelf to be able to retrieve any stored file using its name. To be able to implement the Safe Box shelf, a data structure should be in place to allow searching the shelf for files by name. The data structure should be written on disk, and it is equally important to encrypt its content while it is stored on disk. Other secondary operations can be performed on the Safe Box, the shelf, other than import and export; e.g delete a file, list names of all files in the shelf, etc.



This is a term-project that you will work on it in teams of maximum 3 members. The project will be divided up into 4 main phases, where each phase will be responsible for a working part of the Safe Box software. You are required to conduct a UML design for all your work and apply necessary updates and modifications across different phases to keep your design up to date. You are required to implement all the features and functionalities of the Safe Box that will be detailed in each phase handout. You will be required to demo your work by the end of each phase.

You are required apply all the elements of the object model studied and explained in class, and try to utilize all the C++11/14 features possible, such as STL containers. You are also expected to utilize concepts like inheritance,

templates, virtual methods, and polymorphism; **basically a functional approach will not be considered at all**. You should also be able to utilize different technologies such as **multi-threading**, **container-based approach**, **dynamic shared objects (FDO)**, and you will definitely be working with the **crypto++** library to perform the needed encryption and decryption operations.

Crypto Systems Extra Lab Lectures

VERY IMPORTANT: the professor will conduct 2 volunteer lab lectures which will be announced to explain some fundamental security concepts. It is not mandatory to attend these lectures but you are very much encouraged to attend them as they will deepen your understanding and broaden your comprehension of the crypto++ library which you are going to use. Moreover, these 2 lectures will give you a smooth start in the topic of security and crypto systems that will pave the way for you and make it more smooth when you take a more specialized course in security. **We will still take attendance to know who attended and who did not, but this will not count in your attendance formula.**

Grading

This project is worth 20% of the overall course grade. The project will be graded on a 100% grade scale, and then will be scaled down to the 20% its worth. The project contains 4 phases. Each mandatory phase is worth 25% of the project. Each phase will be graded out of 100% and then scaled down to the 25% it is worth of the project grade. The total grade of the project will be calculated out of 100% and then scaled down to 20% (divided by 5), which will be your final grade.

The project will be evaluated based on the correct execution of your implementation of the Safe Box, the code correctness and neatness, design decision aspects, and how much you have utilized the concepts of abstraction, encapsulation, modularity, and hierarchy. It is very important to highlight that although in-line documentation is worth 10% of each phase, yet missing or unclear documentation might result in more grade deduction if the grader (TA) cannot verify the correctness of your code and logic, and the provided documentation was not sufficient to clarify such matters.

Important: Each group should provide a contribution document that documents and details the work and effort done by each group member in each phase. At the end of each phase, each group will need to present the work through a planned scheduled demo with the TA, and the TA will ask individual questions to the group members to make sure that the workload was equally distributed among them. **Consequently, it is very important to highlight that different individual grades can be assigned to different group members based on the effort done by each, and it is totally the**

responsibility of each group member to show during the demo the amount of work done. Failing to demonstrate your efforts will result in deduction of grades to the corresponding group member.

GIT Repository

Each group will be assigned a GIT repository which will be configured with the group members public keys. The repository will have 4 directories, one for ache phase. Each group is required to submit each phase code under its corresponding directory.

Delays

You have up to 3 calendar days of delay in each phase (**except for the final phase**) after which the corresponding phase will not be accepted and your grade in that case will be ZERO. For every day (of the 3 allowed days per phase), a penalty of 10% will be deducted from the total grade of the corresponding phase.

Let's Start

Please download the first phase handout (***The Building Block***) off black board and start working on it right away.