CS 6377 Introduction to Cryptography

Programming Project[1]

Assigned on Marc 28th, 2011
Due on May 2nd, 2011

Google Docs is a free, Web-based word processor, spreadsheet, presentation, and form application offered by Google. It allows users to create and edit documents online while collaborating in real-time with other users.[2] One problem with Google Docs is, you need to trust Google with respect to security of your files. In this project, you need to write a program to securely open, process and store text files in encrypted format using Google Docs and verify that it has not been modified while it is stored on Google Docs.

In order to do this, first, you need to encrypt your message so that no one else should be able to read your private message details without knowing the encryption key. To do this, use AES Algorithm in CTR mode. Second, in addition to privacy, you should also guarantee the authenticity of your message. When your message is encrypted, your program should calculate HMAC of the ciphertext and append the HMAC value of the ciphertext to the encrypted message. Whenever you want to access your message, your implementation should first verify that the content of the stored message has not been changed by recalculating the HMAC value of the ciphertext. By doing so, you will make sure that if someone has altered the contents of the encrypted message on the Google servers. If the HMAC value of the retrieved message is different from the recalculated HMAC, then you should inform the user that contents have been modified.

You are also encouraged to build up a user interface that shows your program working for each process including encryption of your original message, calculating and appending the HMAC code, uploading, downloading, decryption and etc.

Here is a sample of the implementation (Secure Notepad):

1. Encryption (as shown in Figure 1): Create a new encrypted file using an input password (key) and an underlying encrypt algorithm. After that, the

---

[1] Please form groups of size two or three.

[2] Google Documents List Data API:
http://code.google.com/apis/documents/overview.html

HMAC should be calculated using the encrypt message along with the key and appended to the end of this message with a separator. Then by pressing the "save file" button, you can save the file at Google Docs.

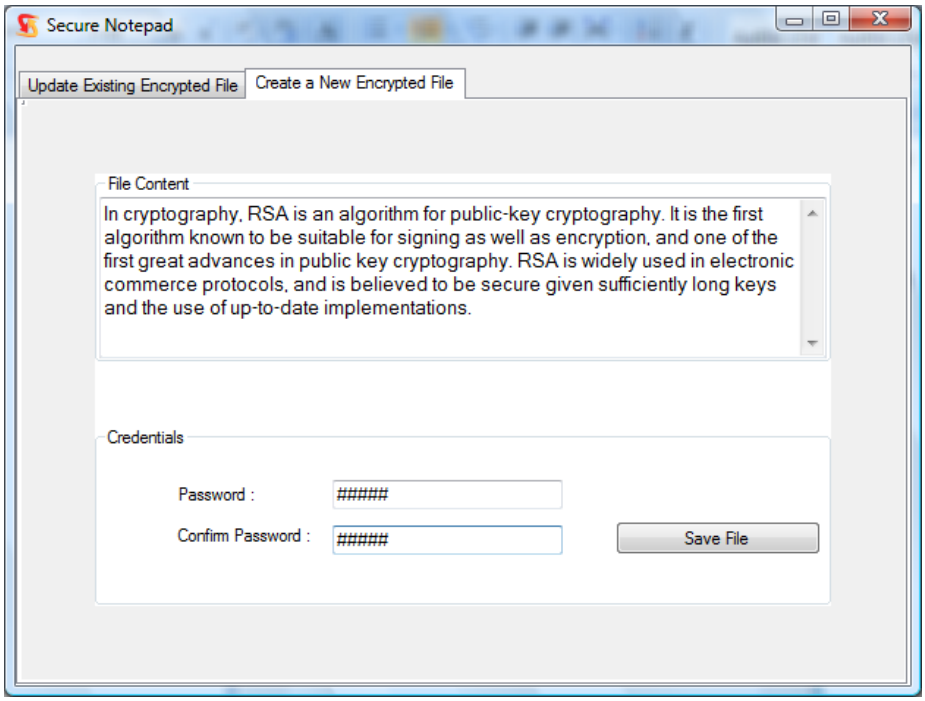

Figure 1

2. Decryption (as shown in Figure 2): fetch the encrypted file from the Google Docs and separate the encrypted message from the HMAC value according to the separator. Then you are able to calculate the HMAC value again using the encrypted message and compared it with the original one. If their values are not equal, which means the message content has been changed, prompt a warning that tells the user the original message has been modified. Otherwise, the message will be successfully decrypted using the correct key you provide and shown in the "File content" area. Of course, the message must be exactly as the same as the original one.

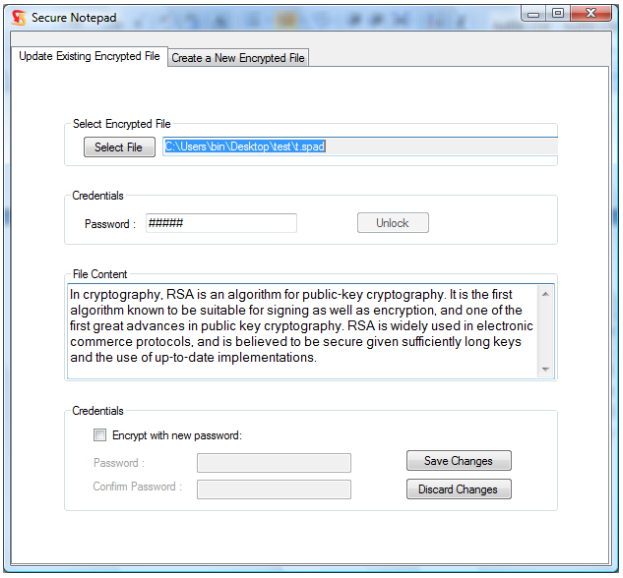

Figure 2

Grading

Please submit all the source files and related documentation as a zip file using Webct before the deadline. Your project grade will be determined based on the correctness of the implementation and the demo you will give to TA.

Alternative Projects

If you want, you can come up with your own project idea but before starting a different idea you must come and discuss with me.