**IT IS 6240/8240, Fall 2014, Assignment A1**

**Privacy Enhanced Cloud Storage Admitting Deduplication**

The goal of this assignment is to design and implement a software utility that enhances a user’s privacy when storing data files in a cloud storage, specifically DropBox. Your software should accomplish a few goals:

* Encrypting a file, prior to being uploaded to the cloud storage, in such a way that it admits deduplication by the cloud storage service provider, and
* Decrypting the encrypted file upon downloading

And to get additional points as an bonus:

* Sharing the file with other users in a secure manner.

**Convergent encryption**

Convergent encryption is a self-encryption technique that encrypts a file with a key derived from the file itself. The key is obtained by applying a one-way hash to the file. What follows is an outline of steps for convergent encryption by a cloud storage user Alice:

1. Given a file *F*, Alice derives an encryption key *K* from the file by applying SHA-256 to *F*.
2. She then encrypts the file *F* into a ciphertext *C* with AES in Counter mode (AES-CTR) under the key *K*.
3. She protects the key *K* by encrypting it into *W* using her own public key.
4. She then uploads both *C* and *W* to the cloud storage server which will ensure that *C* and *W* are stored together.

At a later stage, Alice can decrypt the encrypted file *C* into the original file *F* by

1. Downloading both *C* and *W* from the cloud storage server.
2. Extracting the key *K* from *W* by the use of her private RSA decryption key.
3. Decrypting the ciphertext *C* with AES-CTR under the key *K* to recover the original file *F*.

**Deduplication**

Consider two users uploading two encrypted files *C1* and *C2*. If *C1* and *C2* are identical, then their original files are considered to be identical too. The storage server can save a significant amount of storage space by storing one copy of the encrypted file only, say *C1*, while providing both users’ with the same pointer to the location where the encrypted file *C1* is stored.

An efficient way to check whether two encrypted files *C1* and *C2* are identical is to compare their one-way hash values, that is to compare SHA256(*C1*) against SHA256(*C2*). If the two hash values are the same, then the two encrypted files, and as a result the two original files, are considered to be identical.

Typically the storage server would store SHA256(*C1*) as part of the metadata set for the encrypted file *C1*.

**Programming language and software tools**

You are required to complete the cryptographic programming project in Python ([www.python.org](http://www.python.org)), by utilizing the PyCrypto cryptographic package (www.pycrypto.org, or www.dlitz.net/software/pycrypto).

A tutorial on Python can be found at

<https://wiki.python.org/moin/BeginnersGuide>

Examples for using PyCrypto can be found at

<https://www.dlitz.net/software/pycrypto/doc/>

<http://www.laurentluce.com/posts/python-and-cryptography-with-pycrypto/>

<https://leanpub.com/pycrypto/read>

Information on DropBox API can be found at <https://www.dropbox.com/developers>

**Grading policy**

Code functioning correctly: 80%

Report: 20%

**Submission**

You are required to

1. Demonstrate your completed software tool during a class at the end of the semester. Your code should include the following components:
   1. Client side
      1. Generate public and private key pairs;  
         all users have their own public-private key pairs.
      2. do convergent encryption of a file prior to uploading.
      3. decrypt a files upon downloading.
      4. Sharing
         1. Generate a re-sealed key together with a pointer (URL) for sharing a file with a friend (assuming the friend’s public key is already known)
         2. Upon receiving a pointer and a sealed key, do all the required decryption operations to obtain the original file.
   2. Server side
      1. Check an incoming file for deduplication: if its hashed value is identical to the hash value of an existing file, mark the new file as a duplicate.
2. Submit your code, together with a report. The report should contain the following information:
   1. the design and implementation of your software tool
   2. benefits and shortcomings of the tool
   3. a user guide on how to use it.

**Bonus points (+10%)**

**Adding a Feature for File Sharing**

If Alice wishes to share her file *F* with her friend Bob, Alice can do so by

1. Fetching *W* together with the pointer (URL) to the encrypted file *C* on the storage server.
2. Extracting the key *K* from *W* by the use of her private RSA decryption key.
3. Re-sealing the key *K* by encrypting it into *WB* using Bob’s public RSA encryption key.
4. Informing Bob of *WB* together with the pointer to *C*.

Upon receiving *WB* and the pointer to *C*, Bob can extract the key *K* from *WB* with his private RSA decryption key. He can then use *K* to decrypt *C* and obtain *F*.

Diagram for Convergent Encryption

***F*: Original File**

**SHA-256**

***K***

***C*: Encrypted File**

**Public Key**

***W*: Encrypted *K***

**AES-CTR**

**RSA-OAEP**

**Please include the following declaration together with your signatures on the cover page of your reports:**

1. If it’s an individual assignment, use the following text for the declaration of independent completion:

I \_\_\_\_\_\_\_\_\_\_\_\_\_ declare that I have completed this assignment completely and entirely on my own, without any consultation with others. I understand that any breach of the UNCC Code of Academic Integrity may result in severe penalties.

1. If it’s a group assignment, use the following text for the declaration of independent completion:

|  |  |  |  |
| --- | --- | --- | --- |
| We declare that we have completed this assignment completely and entirely on our own, without any consultation with others. We understand that any breach of the UNC Charlotte Code of Academic Integrity may result in severe penalties.  We also declare that we have agreed on the following percentages of contributions to the completion of the assignment by individual group members: | | | |
| Name | Contribution (%) | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |