SIL765
Assignment 2

Yash Gupta (2013CS10302), Ujjwal Sinha (2012CH70185)

Project Selection:

A1 = last_4_digits_of_entry_no_of_first_student = 0302
A2 = last_4_digits_of_entry_no_of_second_student = 0185

k = A1+A2 mod 6 = 487 mod 6 = 1
Where k is the project number. Thus, project 1 was selected.

Project 1: Timestamping a document

Problem statement:

This application relates to time-stamping a document that one may have prepared some time ago, and returning it in a secure manner with the date/time appended to it. Assuming that the "GMT date and time-stamping server" has the correct GMT date and time, it time-stamps the document (in some standard format) with the current GMT data/time and a digital signature. At any time, it should be possible to establish the fact that the document existed at the date/time stamped, and that the document has not been modified.
Further:

1. How do you ensure privacy, in that the server does not see/have/keep the original document?
2. How do you share the document with others in a secure manner with the date/time preserved, and integrity un-disturbed?

Also ensure that the user has (and uses) the correct "public-key" of the date/time stamping server.

4. Would access to "public-key certificate" issued by a certification authority be an issue?

A trusted **digital timestamp** gives strong legal evidence that the contents of original work existed at a point-in-time and have not changed since that time. The procedures maintain complete privacy of original documents themselves.

Steps involved in timestamping a document which was prepared some moments ago:

1. The client application creates a hashed value (as a unique identifier of the data or file that needs to be timestamped) and sends it to the Time Stamping Authority server. We have used SHA1 algorithm to create a hash value of the file that needs to be timestamped.
2. The Time Stamping Authority (TSA) combines the hash and other information, including the authoritative time. The result is hashed using SHA1 algorithm and then it's digitally signed with the TSA's private key, creating a signature which is sent back to the client along with the timestamp.
3. The timestamp token is received by the client application and recorded within the document.
4. When the resulting timestamped data or file is opened in the future, the client application will use the TSA's public key to authenticate the TSA (i.e. validate that the timestamp came from a trusted TSA) and re-calculate a hash of the original data. This new hash is compared to the originally created hash (step 1 above). If any changes have been made to the data since the timestamp was applied, this hash check will fail and warning messages will be shown saying that the data has been altered and it should not be trusted.

Further:

1. How do you ensure privacy, in that the server does not see/have/keep the original document?
   Ans: To ensure privacy, the document is hashed using SHA1 algorithm before sending it to the server. In this case, the server doesn't know the actual contents of the file and after timestamping, it returns a digitally signed copy of the document hash along with the timestamp and thus the document is not stored in the server.

2. How do you share the document with others in a secure manner with the date/time preserved, and integrity un-disturbed?
   Ans: To share the document in a secured manner, we can use RSA algorithm to encrypt the digitally signed document with receiver's public key and the receiver can then decrypt it using his private key, ensuring that confidentiality is maintained.
   The signature acquired from the TSA can be used to ensure integrity and that date/time is preserved. (Provided the receiver trusts the TSA)

3. Also ensure that the user has (and uses) the correct "public-key" of the date/time stamping server.
   Ans: To ensure that the user uses the correct public-key of the TSA, we need to have a trusted central authority in the system.
   This authority could be the TSA itself (In which case we can obtain the public key from TSA and assume it is the correct key), or a different entity that provides a certificate which binds TSA's public key to TSA's identity.

4. Would access to “public-key certificate” issued by a certification authority be an issue?
   Ans: Yes, access to a public key certificate is a major issue, as if the communication channel b/w the certification authority (CA) is compromised, the user can store invalid “public-key certificate” for the CA, which could instead be crafted by a malicious man-in-the-middle.
   This malicious certificate could then indicate to the user that the public-key of a malicious TSA is valid and signed by the CA, resulting in malicious time stamp signatures being obtained by the user.
   To counter-act this attack, it is often the case that root-CA certificates are distributed in the firmware by the manufacturer of devices. These root-CA certificates can then authorize other CA’s recursively, in a tree like fashion.