Name (Print) ____ **448 or 548**

This project deals with OpenSSL. OpenSSL is an open source toolkit that implements Secure Sockets layer and Transport Layer Security protocols. For this project OpenSSL will be used to create message digests (one way hash function results), public and private keys and some encryption algorithms. OpenSSL is available on the linux machines in EB246 and EB216.

1. Download the project zip file from Canvas. The following files (“Desert.jpg”, “plain1.txt”, “plain2.txt”, “Lecture.pdf”, “file.enc”, “File.txt” and “Clint.wav”) should be in that zip file.

2. On a linux computer in the lab, open a terminal window and run ***openssl*** to get the prompt **OpenSSL>** You can type help to see a listing of all the commands available, or go to https://www.openssl.org/docs/man1.0.1/apps/

3. What is the version number of ***openssl***? ___________________

To see a listing of commands in ***openssl*** type in the word help to obtain help. To obtain help with specific commands in ***openssl,*** type the command followed by –h (i.e. dgst –h)

Part I. *Creating Digests (output is a string of hexadecimal digits)*

4. Create the MD5 digest of the file “Lecture.pdf” by ***dgst –md5 Lecture.pdf***

   How long is the digest created (in terms of hexadecimal digits)? ____________

   What are the leftmost three hexadecimal digits of the digest? ____________

   What are the rightmost three hexadecimal digits of the digest? ____________

5. Create the SHA1 digest of the file “Lecture.pdf” by ***dgst –sha1 Lecture.pdf***

   How long is the digest created (in terms of hexadecimal digits)? ____________

   What are the leftmost three hexadecimal digits of the digest? ____________

   What are the rightmost three hexadecimal digits of the digest? ____________

6. For the file “Clint.wav”, repeat Steps 4 and 5 and answer the following questions.

   What are the three leftmost hexadecimal digits of the MD5 digest? ____________

   What are the three rightmost hexadecimal digits of the MD5 digest? ____________

   What are the three leftmost hexadecimal digits of the SHA1 digest? ____________

   What are the three rightmost hexadecimal digits of the SHA1 digest? ____________

7. What is the content of the text file “plain1.txt”? ____________________________________

   What is the content of the text file “plain2.txt”? ____________________________________

   What are the three leftmost hexadecimal digits of the MD5 digest of “plain1.txt”? ____________

   What are the three rightmost hexadecimal digits of the MD5 digest of “plain2.txt? ____________

II. *Generation of Public and Private Keys*

8. Get more information about the ***genrsa*** option by ***genrsa -h***
9. Generate a private key, which is saved to a file named "privatekey.pem", by using
   genrsa -out privatekey.pem 2048
   How long (in bits) is the generated key? _____________
10. Verify the private key generated by ***rsa -check -in privatekey.pem***
    Is the RSA key OK? _____________
11. Produce a public version of the private RSA key generated in Step 9, by running
    rsa –in privatekey.pem -pubout -out pubkey.pem

III. *Signing Digests with Private keys and Verifying Digests with Public Keys*

12. If you want to ensure that the digest you created in Section I will not be modified without your permission, you can sign the digest using your private key generated in Section II.
    Sign the MD5 digest of the file "plain1.txt" by running
    dgst -md5 -sign privatekey.pem -out plain1.txt.md5 plain1.txt
    What is the size (in bytes) of the signed digest file? _______________

13. To verify a signed digest, you will need the file from which the digest was derived, the signed digest file, and the signer's public key. Run
    dgst -md5 -verify pubkey.pem -signature plain1.txt.md5 plain1.txt

    What is the response of the OpenSSL? ____

14. Now run ***dgst –md4 -verify pubkey.pem -signature plain1.txt.md5 plain1.txt***

What is the response of the OpenSSL? ____

Is the verification successful? Why is the verification successful or unsuccessful?

IV. Encryption

You can list all the ciphers you can employ for encryption by running ***list-cipher-commands***

15. In the following steps, we choose *aes-256-cbc* for encryption and decryption.

    What does “256” mean? ____

    What does “cbc” mean? ____

16. Encrypt the file “plain1.txt” by running

    ```
    enc -aes-256-cbc -a  -in plain1.txt -out cipher1
    ```

    Enter the password ***test***. View the content of the encrypted file.

17. Encrypt the file “plain1.txt” again by running the same encryption

    ```
    enc -aes-256-cbc -a  -in plain1.txt -out cipher2
    ```

    Enter the same password ***test***. View the content of the encrypted file.
    Compare the cipher texts “cipher1” and “cipher2”. Are they exactly the same? _______

    Why? (Hint: Repeat Steps 16 and 17 by using an additional option -p to view the keys used for encryption.)

18. Decrypt the file “file.enc” using *aes-256-cbc* (with the –a option) and password ***test2***

    What is the command you run? ____
    (Hint: Find out the options available by enc -h)

    What is the plain text (there are two lines followed by a blank line)?

V. Data Compression: *gzip, bzip2,* and *compress*

Note: these are run from the regular command prompt. They are not part of openssl

Read the man page of gzip, bzip2 and compress.. In some cases, a compression algorithm results in a larger file than the original. To see this effect, you need to make sure that the flag for forcing compression is used.

19. Read the ***man*** page of ***gzip***, ***bzip2***, and ***compress***.
    What is the version number of ***gzip***? _____________

    What is the option used to decompress a file (produced by either ***gzip*** or ***bzip2***)? ___________

    What is the command used to decompress a file produced by ***compress***? _________________

20. What is the size (in bytes) of the file "Lecture.pdf"? _________________

    What is the size (in bytes) of the file "Clint.wav"? _________________

    What is the size (in bytes) of the file "Desert.jpg"? _________________
    Note: All file sizes are exact number of bytes (not in Kbytes)

21. Compress the file Lecture.pdf by running ***gzip Lecture.pdf***
    What is the name of the compressed file? ___________________________

    What is the size of the compressed file (in bytes)? ________________

    What is the compression ratio (original file size / compressed file size)? ________________

22. Compress the file by running ***bzip2 Lecture.pdf***
    Note: You need to first recover the original file by decompressing the output of Step 21.
    What is the name of the compressed file? ___________________________

    What is the size (in bytes) of the compressed file? ___________________

    What is the compression ratio (original file size / compressed file size)? _________________

23. Compress the file by running ***compress Lecture.pdf***
    Again, you need to recover the original file by decompressing the output of Step 22.
    What is the name of the compressed file? _________________________

    What is the size (in bytes) of the compressed file? ___________________

    What is the compression ratio (original file size / compressed file size)? _______________

24. Now compress the other three files ("Clint.wav", "File.txt" and "Desert.jpg") by repeating Steps 21, 22 and 23. Fill in the three tables on the next page with the sizes of the compressed files (in exact number of bytes) and the compression ratio. For some of the files, the compression ratio will be less than 1 – compressed file is larger than original. **Be sure to use the flag/option that forces compression of a file even if the compressed file will be larger.**

Compression table for Lecture.pdf

Compression Type	Size in Bytes	Compression Ratio
Original (Uncompressed)		------------
gzip		
bzip2		
compress		

Compression table for Clint.wav

Compression Type	Size in Bytes	Compression Ratio
Original (Uncompressed)		------------
gzip		
bzip2		
compress		

Compression table for Desert.jpg

Compression Type	Size in Bytes	Compression Ratio
Original (Uncompressed)		------------
gzip		
bzip2		
compress		

Compression table for File.txt

Compression Type	Size in Bytes	Compression Ratio
Original (Uncompressed)		------------
gzip		
bzip2		
compress		

25. Comment on the data in the above tables: which file compressed the worst? Best? How good was compression on the JPEG image? What is/are the reason(s) for the results with the JPEG image?