CSE 410/565 Spring 2022
Homework 2 – Performance Report

Name: Sai Mahesh Pullagura
UBNo: 50419368
UBName: spullagu

Configuration Details:

- **Programming language –** Python 3.8.10 ([GCC 9.3.0] on linux).

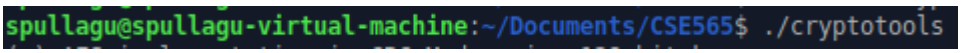
- **Library option** **–** cryptography version 36.0.1

  Additionally, the below steps are followed to update the cryptography package in the Ubuntu virtual machine (UB CSE Fall 2021 virtual machine image).

  sudo apt-get update
  sudo apt-get -y install python3-pip
  pip3 install cryptography –upgrade

- **Name of the shell script file** -- cryptotools
  (Note that the file does not have the extension.sh).

  To generate the output, we run the command → **./cryptotools** .

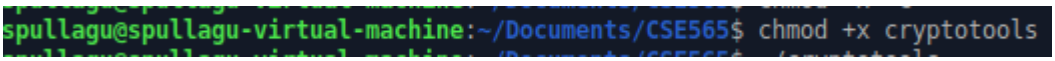
  

  Please note that in the above shell script file "***cryptotools***", the commands to execute the actual python file "***cryptotools.py***" is included as follows:

  #!/bin/sh
  python3 cryptotools.py

  Thus when we run the command *"./cryptotools"*, it invokes the indicated python file and generates the final output.

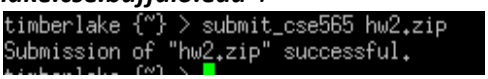
  Furthermore, in order to execute the shell script, I have given the execute permission using the below command → **chmod +x cryptotools.**

- **Other files** along with the shell script that are included in the uploaded zip folder are:

  Python file → cryptotools.py
  Sample text files on which the cryptography operations are performed:
  Sample_File_1KB.txt
  Sample_File_10MB.txt
  Sample_File_1MB.txt (used in the case of RSA implementation).

➔ *Finally, the shell script cryptotools, python file cryptotools.py and the 3 sample text files mentioned above are placed in a single folder* ***hw2****, zipped it, and submitted into the server* ***"timberlake.cse.buffalo.edu"****.*



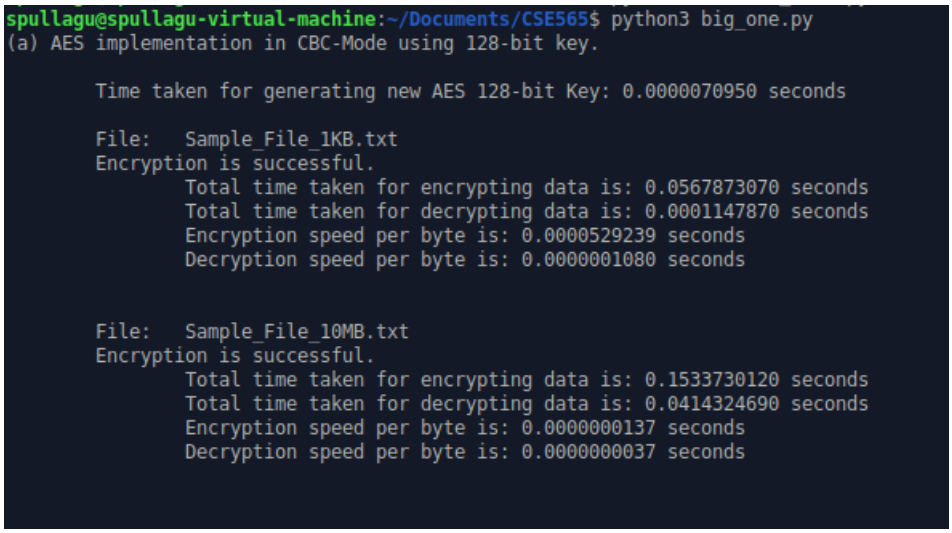
Performance View:

- We have implemented all 8 tasks like encryption, hashing and signature operations and time results are noted.
- The various time results measured in each task are:

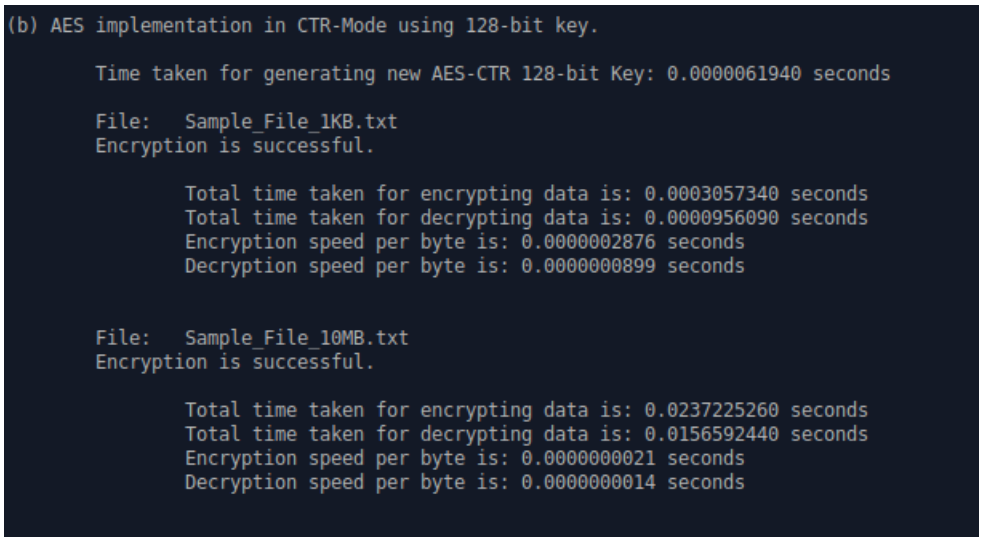
  Key generation time, Encryption/ Decryption time, Encryption/ Decryption speed per byte, Time taken for Hashing the files and Signature/Verification time.

The below screenshots present the exact measured times for each task.

1. AES implementation in CBC-Mode using 128-bit key.



2. AES implementation in CTR-Mode using 128-bit key.



3. AES implementation in CTR-Mode using 256-bit key.

```
(c) AES implementation in CTR-Mode using 256-bit key.

        Time taken for generating new AES-CTR 256-bit Key: 0.0000061480 seconds

        File:   Sample_File_1KB.txt
        Encryption is successful.

                Total time taken for encrypting data is: 0.0002546650 seconds
                Total time taken for decrypting data is: 0.0002075670 seconds
                Encryption speed per byte is: 0.0000002396 seconds
                Decryption speed per byte is: 0.0000001953 seconds


        File:   Sample_File_10MB.txt
        Encryption is successful.

                Total time taken for encrypting data is: 0.0283968370 seconds
                Total time taken for decrypting data is: 0.0153213050 seconds
                Encryption speed per byte is: 0.0000000025 seconds
                Decryption speed per byte is: 0.0000000014 seconds
```

4. RSA implementation with PKCS #1 v2 padding-OAEP using 2048-bit key.

```
(d) RSA implementation with PKCS #1 v2 padding-OAEP using 2048-bit key.

        Time taken for generating new RSA 2048-bit Key: 0.2677032630 seconds

        File:   Sample_File_1KB.txt
        For the data of length 1030 ,the computed chunk_size is 190 bytes per each chunk and the total number of chunks are 6 .
        Encryption is successful.
                Total time taken for encrypting data is: 0.0005683230 seconds
                Total time taken for decrypting data is: 0.0050883360 seconds
                Encryption speed per byte is: 0.0000005518 seconds
                Decryption speed per byte is: 0.0000049401 seconds


        File:   Sample_File_1MB.txt
        For the data of length 1048576 ,the computed chunk_size is 190 bytes per each chunk and the total number of chunks are 5519 .
        Encryption is successful.
                Total time taken for encrypting data is: 0.3515159110 seconds
                Total time taken for decrypting data is: 4.7093110010 seconds
                Encryption speed per byte is: 0.0000003352 seconds
                Decryption speed per byte is: 0.0000044911 seconds
```

5. RSA implementation with PKCS #1 v2 padding-OAEP using 3072-bit key.

```
(e) RSA implementation with PKCS #1 v2 padding-OAEP using 3072-bit key.

        Time taken for generating new RSA 3072-bit Key: 0.5039777900 seconds

        File:   Sample_File_1KB.txt
        For the data of length 1030 ,the computed chunk_size is 318 bytes per each chunk and the total number of chunks are 4 .
        Encryption is successful.
                Total time taken for encrypting data is: 0.0004494030 seconds
                Total time taken for decrypting data is: 0.0097833500 seconds
                Encryption speed per byte is: 0.0000004363 seconds
                Decryption speed per byte is: 0.0000094984 seconds


        File:   Sample_File_1MB.txt
        For the data of length 1048576 ,the computed chunk_size is 318 bytes per each chunk and the total number of chunks are 3298 .
        Encryption is successful.
                Total time taken for encrypting data is: 0.3081243449 seconds
                Total time taken for decrypting data is: 8.0886169109 seconds
                Encryption speed per byte is: 0.0000002939 seconds
                Decryption speed per byte is: 0.0000077139 seconds
```

6. Hash Function implementations - SHA-256, SHA-512, SHA3-256.

```
(f) Hash Function implementations - SHA-256, SHA-512, SHA3-256.

        SHA-256 Function.
                File:   Sample_File_1KB.txt
                Time taken for computing hash value is:0.0001291590 seconds
                Hash Time per byte: 0.0000001254 seconds

                File:   Sample_File_10MB.txt
                Time taken for computing hash value is:0.0370510010 seconds
                Hash Time per byte: 0.0000000033 seconds



        SHA-512 Function.
                File:   Sample_File_1KB.txt
                Time taken for computing hash value is:0.0001942360 seconds
                Hash Time per byte: 0.0000001886 seconds

                File:   Sample_File_10MB.txt
                Time taken for computing hash value is:0.0262581550 seconds
                Hash Time per byte: 0.0000000023 seconds



        SHA3-256 Function.
                File:   Sample_File_1KB.txt
                Time taken for computing hash value is:0.0002188350 seconds
                Hash Time per byte: 0.0000002125 seconds

                File:   Sample_File_10MB.txt
                Time taken for computing hash value is:0.0471375420 seconds
                Hash Time per byte: 0.0000000042 seconds
```

7. DSA implementation using 2048-bit key and SHA-256 hash function.

```
(g) DSA implementation using 2048-bit key and SHA-256 hash function.

        Time taken for generating new DSA 2048-bit Key: 0.2449247750 seconds

        File:   Sample_File_1KB.txt
                Time taken for producing signature on the file is: 0.0390070680 seconds
                Time taken for verifying signature on the file:    0.0005925110 seconds
                Compute time per byte for signature is: 0.0000378709 seconds
                Compute time per byte for verification is: 0.0000005753 seconds
                The data is Authentic.


        File:   Sample_File_10MB.txt
                Time taken for producing signature on the file is: 0.0382111520 seconds
                Time taken for verifying signature on the file:    0.0372774630 seconds
                Compute time per byte for signature is: 0.0000000034 seconds
                Compute time per byte for verification is: 0.0000000033 seconds
                The data is Authentic.
```

8. DSA implementation using 3072-bit key and SHA-256 hash function.

```
(h) DSA implementation using 3072-bit key and SHA-256 hash function.

        Time taken for generating new DSA 3072-bit Key: 1.6440058770 seconds

        File:   Sample_File_1KB.txt
                Time taken for producing signature on the file is: 0.0011818780 seconds
                Time taken for verifying signature on the file:    0.0010812420 seconds
                Compute time per byte for signature is: 0.0000011475 seconds
                Compute time per byte for verification is: 0.0000010497 seconds
                The data is Authentic.


        File:   Sample_File_10MB.txt
                Time taken for producing signature on the file is: 0.0479524750 seconds
                Time taken for verifying signature on the file:    0.0482889050 seconds
                Compute time per byte for signature is: 0.0000000043 seconds
                Compute time per byte for verification is: 0.0000000043 seconds
                The data is Authentic.
```

Comments:

1. How per byte speed changes for different algorithms between small and large files?

- The expected performance is that the files with larger size have faster per byte speed when compared to the file with smaller size.
- And as expected, for all of the 8 algorithms implemented, the files with larger size comparatively have faster speed per byte with respect to the files with small size.
- It means that with respect to the 1KB file, the speed per byte is faster for 1MB file and when compared with 1MB file, the 10MB file has even better per byte speed.

2. How encryption and decryption times differ for a given encryption algorithm?

- On a whole, the time taken by encryption and decryption functionalities will be dependent on the size of the file. So, for a file with large size (eg: 10MB), it will take more time to encrypt and decrypt such large data when compared to a file with smaller size (eg: 1KB file).
- Apart from that, the encryption and decryption times also depend on the type of algorithm being implemented.
- In case of AES algorithms in both CBC and CTR modes (with different key sizes), the encryption and decryption operations almost take same time with encryption taking slightly more time than decryption. This is a expected behaviour as encryption is usually performed sequentially while decryption is done in parallel in case of AES.
- Coming to RSA algorithms with different key sizes, the observation is that encryption operation takes less time when compared to decryption and the gap is even large in case of large files. This is also an expected behaviour because, the public key used in encryption is chosen manually while the private key used in decryption would require to be computed each time. As indicated in the screenshots of RSA implementation, the encryption operation only takes few nano to micro seconds but the decryption process takes seconds to compute and determine the plain text.

- Lastly, when considering the DSA algorithm of different key sizes, both operations take almost similar time to complete.

3. How key generation, encryption, and decryption times differ with the increase in the key size?

- When considering the AES algorithm, an increase in key size from 128 to 256 bits do not significantly affect the key generation time. Also, the change in key size DO NOT considerably impact the encryption and decryption times. Hence in case of AES algorithm, change in key size do not greatly affect the key generation, encryption, and decryption times which is also an expected behaviour.
- With respect to RSA and DSA algorithms, change in key sizes DO affect the time taken for key generation ranging from few microseconds for a small key size (2084-bit) to few seconds for a large key size (3072-bit). Even though, there is a large rise in key size, encryption and decryption operations still get completed in relatively faster time when applied on larger files. Thus, we can say that change in key size do affect the key generation time but it does not impact the encryption and decryption times.

4. How hashing time differs between the algorithms and with increase of the hash size?

- For files of small size like the 1KB file, SHA-256 comparatively hashes faster with respect to the other 2 approaches.
- However, for files with larger size like the 10MB file, SHA-512 hashes relatively quicker than the other 2.
- And with respect to increase in hash size, hashing time is small for large key sizes and this change can be clearly seen while hashing large files. Hence as expected, SHA-512 works efficiently among the 3 techniques.

5. How performance of symmetric key encryption (AES), hash functions, and public-key encryption (RSA) compare to each other?

- As a symmetric key encryption like AES makes use of single key for encryption as well as decryption, it will work faster when compared to public-key encryption like RSA which uses 2 keys each for encryption and decryption. However, on the security perspective, RSA is much secure compared to AES.
- When being compared to hashing functions, AES take similar time as hashing (like SHA-256) but RSA is relatively slower compared to hashing ad AES functions.