

CSC 574 Computer Network Security

Assignment 2

Omkar Parkhe (ojparkhe@ncsu.edu)

July 3, 2019

1 Details of testing Environment:

Following are the details of our system specifications on which we have executed the program using the input file provided by Prof. Brad Reaves.

1. Input file: input.txt (100 Mb)
2. System Specifications:
 - (a) Java: jdk 1.8
 - (b) Operating System: Windows 10
 - (c) RAM: 16 GB
 - (d) Hard Disk: 1 TB
 - (e) Processor: i7, 7th generation, 2.70 GHz

2 Outputs

Note: All time values displayed in this report are in **seconds**.

2.1 AES-128 Encryption Scheme

Fig.1 shows time taken for encrypting and decrypting using AES-128 on an input file of size 100mb. Fig.2 shows the mean and median of AES-128 encryption and decryption algorithm for 100 iterations over input size 100mb.

Mean encryption time is calculated by finding sum over all the time taken for 100 iterations of AES-128 encryption and then dividing it by 100. Median encryption time is calculated by first sorting the seconds array which stores the computation time for each AES-128 encryption iteration. Now median is the average of 50th and 51st element in the seconds array.

Mean decryption time is calculated by finding sum over all the time taken for 100 iterations of AES-128 decryption and then dividing it by 100. Median decryption time is calculated by first

```
oiparkhe@bn17-111: ~/Downloads/finalcodes
File Edit View Search Terminal Help
---[AES128]---
[1] AES128 Encrypt Running Time = 0.792093657
[1] AES128 Decrypt Running Time = 0.79299512
[2] AES128 Encrypt Running Time = 0.801039
[2] AES128 Decrypt Running Time = 0.959761085
[3] AES128 Encrypt Running Time = 0.801981268
[3] AES128 Decrypt Running Time = 0.869238124
[4] AES128 Encrypt Running Time = 0.804377627
[4] AES128 Decrypt Running Time = 0.858495173
[5] AES128 Encrypt Running Time = 0.809895119
[5] AES128 Decrypt Running Time = 0.873555495
[6] AES128 Encrypt Running Time = 0.802724441
[6] AES128 Decrypt Running Time = 0.86133951
[7] AES128 Encrypt Running Time = 0.802152673
[7] AES128 Decrypt Running Time = 0.868974268
[8] AES128 Encrypt Running Time = 0.803295697
[8] AES128 Decrypt Running Time = 0.869410519
```

Figure 1: AES-128 Encryption and Decryption: Input size 100mb

```
oiparkhe@bn17-111: ~/Downloads/finalcodes
File Edit View Search Terminal Help
[91] AES128 Encrypt Running Time = 0.79118269
[91] AES128 Decrypt Running Time = 0.843763781
[92] AES128 Encrypt Running Time = 0.791276176
[92] AES128 Decrypt Running Time = 0.833847985
[93] AES128 Encrypt Running Time = 0.79194575
[93] AES128 Decrypt Running Time = 0.845474423
[94] AES128 Encrypt Running Time = 0.800214448
[94] AES128 Decrypt Running Time = 0.835563461
[95] AES128 Encrypt Running Time = 0.791201709
[95] AES128 Decrypt Running Time = 0.846757527
[96] AES128 Encrypt Running Time = 0.787534272
[96] AES128 Decrypt Running Time = 0.835714318
[97] AES128 Encrypt Running Time = 0.800322512
[97] AES128 Decrypt Running Time = 0.845415196
[98] AES128 Encrypt Running Time = 0.789766305
[98] AES128 Decrypt Running Time = 0.83195015
[99] AES128 Encrypt Running Time = 0.796751522
[99] AES128 Decrypt Running Time = 0.845803738
[100] AES128 Encrypt Running Time = 0.789760881
[100] AES128 Decrypt Running Time = 0.845799962
AES 128 Encryption Mean :0.7931189294099997
AES 128 Decryption Mean :0.8473306498800001
Encryption Median :0.790856802
Decryption Median :0.8436531469999999
```

Figure 2: AES-128 Encryption and Decryption- mean and median: Input size 100mb

sorting the seconds array which stores the computation time for each AES-128 decryption iteration. Now median is the average of 50th and 51st element in the seconds array.

The mean and median of both encryption and decryption for AES 128 is stated below:

Mean of AES-128 encryption time = 0.793118929

Median of AES-128 encryption time = 0.790856802

Mean of AES-128 decryption time = 0.84733064

Median of AES-128 decryption time = 0.843653146

2.2 AES-256 Encryption Scheme

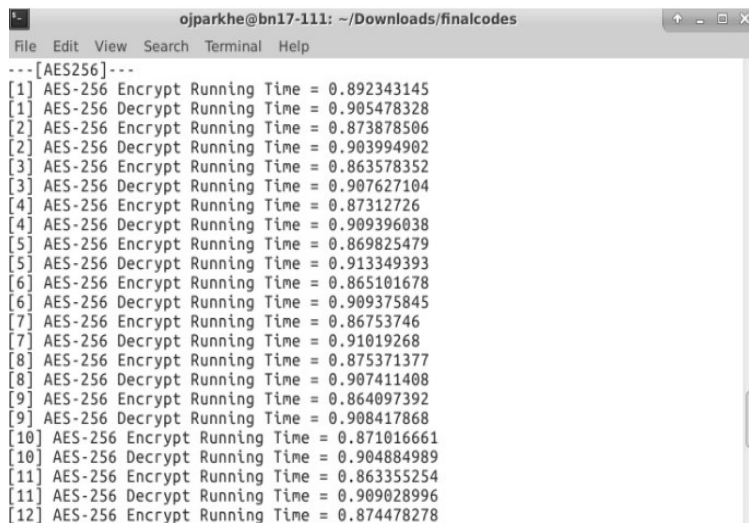


Figure 3: AES-256 encryption and decryption: Input size 100mb

Fig.3 shows time taken for encrypting and decrypting using AES-256 on an input file of size 100mb. Fig.4 shows the mean and median of AES-256 encryption and decryption algorithm for 100 iterations over input size 100mb.

Mean encryption time is calculated by finding sum over all the time taken for 100 iterations of AES-256 encryption and then dividing it by 100. Median encryption time is calculated by first sorting the seconds array which stores the computation time for each AES-256 encryption iteration. Now median is the average of 50th and 51st element in the seconds array.

Mean decryption time is calculated by finding sum over all the time taken for 100 iterations of AES-256 decryption and then dividing it by 100. Median decryption time is calculated by first sorting the seconds array which stores the computation time for each AES-256 decryption iteration. Now median is the average of 50th and 51st element in the seconds array.

```
oiparkhe@bn17-111: ~/Downloads/finalcodes
File Edit View Search Terminal Help
[92] AES-256 Encrypt Running Time = 0.854343781
[92] AES-256 Decrypt Running Time = 0.905162314
[93] AES-256 Encrypt Running Time = 0.857824049
[93] AES-256 Decrypt Running Time = 0.905054734
[94] AES-256 Encrypt Running Time = 0.851560607
[94] AES-256 Decrypt Running Time = 0.901759825
[95] AES-256 Encrypt Running Time = 0.871567319
[95] AES-256 Decrypt Running Time = 0.90382855
[96] AES-256 Encrypt Running Time = 0.852467248
[96] AES-256 Decrypt Running Time = 0.901054041
[97] AES-256 Encrypt Running Time = 0.858708377
[97] AES-256 Decrypt Running Time = 0.905560726
[98] AES-256 Encrypt Running Time = 0.854367074
[98] AES-256 Decrypt Running Time = 0.903272135
[99] AES-256 Encrypt Running Time = 0.861050227
[99] AES-256 Decrypt Running Time = 0.910529099
[100] AES-256 Encrypt Running Time = 0.851556118
[100] AES-256 Decrypt Running Time = 0.90118482
AES-256 Encryption Mean :0.8612149267199999
AES-256 Decryption Mean :0.9071297660799996
AES-256 Encryption Median :0.860085832
AES-256 Decryption Median :0.9063512915
```

Figure 4: AES-256 encryption and decryption- mean, median: Input size 100mb

The mean and median of both encryption and decryption for AES 256 is stated below:

Mean of AES-256 encryption time = 0.8612149267
Median of AES-256 encryption time = 0.860085832
Mean of AES-256 decryption time = 0.907129766
Median of AES-256 decryption time = 0.906351291

2.3 RSA-1024 Encryption Scheme

```
oiparkhe@bn17-111:~/Downloads/finalcodes$ java -jar cryptotest.jar /home/oiparkhe/Desktop/input.txt
Picked up _JAVA_OPTIONS: -Xmx2048m
---RSA 1024---
[0] RSA Encrypt Running Time = 67.54469729600285
[0] RSA Decrypt Running Time = 1246.7431681061007
[1] RSA Encrypt Running Time = 67.1835619150009
[1] RSA Decrypt Running Time = 1249.5315164040387
Encryption Mean and Median: 67.36412960550187
Decryption Mean and Median: 1248.1373422550696
```

Figure 5: RSA-1024 Encryption and decryption: Input size 100mb

Fig.5 shows time taken for encrypting and decrypting using RSA-1024 on an input file of size 100mb. It also displays the mean and median of RSA-1024 encryption and decryption algorithm for 2 iterations over input size 100mb.

Mean encryption time is calculated by finding sum over all the time taken for 2 iterations of RSA-1024 encryption and then dividing it by 2. Median encryption time will be the same as the mean encryption time because we have only two iterations for RSA.

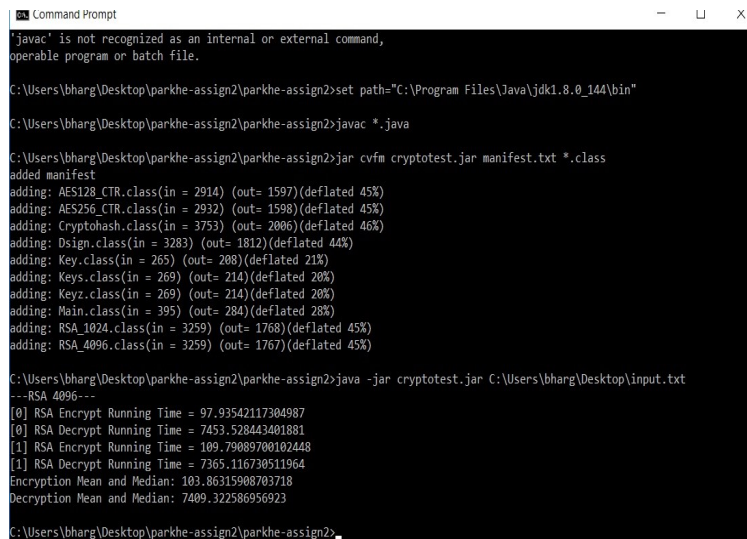
Mean decryption time is calculated by finding sum over all the time taken for 2 iterations of RSA-1024 decryption and then dividing it by 2. Median decryption time will be the same as the mean decryption time because we have only two iterations for RSA.

The mean and median of both encryption and decryption for RSA 1024 is stated below:

Mean and median of RSA-1024 encryption time = 67.364129

Mean and median of RSA-1024 decryption time = 1248.137342255

2.4 RSA-4096 Encryption Scheme



```

C:\Users\bharg\Desktop\parkhe-assign2\parkhe-assign2>set path="C:\Program Files\Java\jdk1.8.0_144\bin"

C:\Users\bharg\Desktop\parkhe-assign2\parkhe-assign2>javac *.java

C:\Users\bharg\Desktop\parkhe-assign2\parkhe-assign2>jar cvfm cryptotest.jar manifest.txt *.class
added manifest
adding: AES128_CTR.class(in = 2914) (out= 1597)(deflated 45%)
adding: AES256_CTR.class(in = 2932) (out= 1598)(deflated 45%)
adding: Cryptohash.class(in = 3753) (out= 2006)(deflated 46%)
adding: Dsign.class(in = 3283) (out= 1812)(deflated 44%)
adding: Key.class(in = 265) (out= 208)(deflated 21%)
adding: Keys.class(in = 269) (out= 214)(deflated 20%)
adding: Keyz.class(in = 269) (out= 214)(deflated 20%)
adding: Main.class(in = 395) (out= 284)(deflated 28%)
adding: RSA_1024.class(in = 3259) (out= 1768)(deflated 45%)
adding: RSA_4096.class(in = 3259) (out= 1767)(deflated 45%)

C:\Users\bharg\Desktop\parkhe-assign2\parkhe-assign2>java -jar cryptotest.jar C:\Users\bharg\Desktop\input.txt
--- RSA 4096 ---
[0] RSA Encrypt Running Time = 97.93542117304987
[0] RSA Decrypt Running Time = 7453.528443401881
[1] RSA Encrypt Running Time = 109.79089700102448
[1] RSA Decrypt Running Time = 7365.116730511964
Encryption Mean and Median: 103.86315908703718
Decryption Mean and Median: 7409.322586956923
C:\Users\bharg\Desktop\parkhe-assign2\parkhe-assign2>

```

Figure 6: RSA-4096 Encryption and decryption: Input size 100mb

Fig.6 shows time taken for encrypting and decrypting using RSA-4096 on an input file of size 100mb. It also displays the mean and median of RSA-4096 encryption and decryption algorithm for 2 iterations over input size 100mb.

Mean encryption time is calculated by finding sum over all the time taken for 2 iterations of RSA-4096 encryption and then dividing it by 2. Median encryption time will be the same as the mean encryption time because we have only two iterations for RSA.

Mean decryption time is calculated by finding sum over all the time taken for 2 iterations of RSA-4096 decryption and then dividing it by 2. Median decryption time will be the same as the mean decryption time because we have only two iterations for RSA.

The mean and median of both encryption and decryption for RSA 4096 is stated below:

Mean and median of RSA-4096 encryption time = 103.8631590870

Mean and median of RSA-4096 decryption time = 7409.322586956923

2.5 MD5 hash

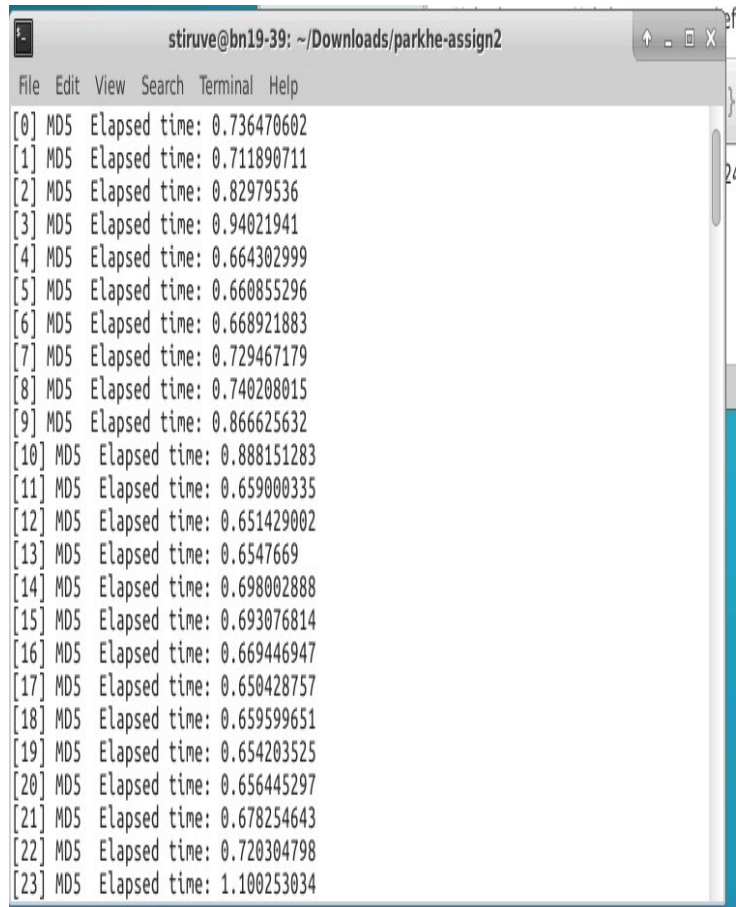
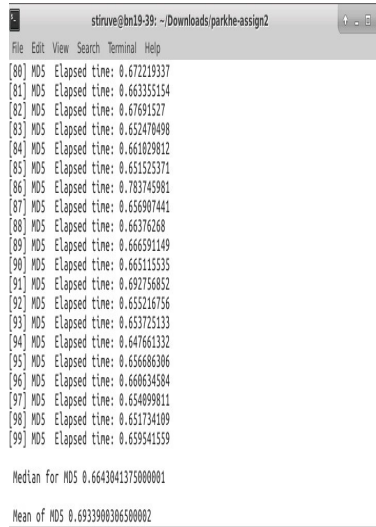


Figure 7: MD5 hash: Input size 100mb

Fig.7 This shows time taken for computing MD5 hash on an input file of size 100mb. Fig.8 shows the mean and median of MD5 hash algorithm for 100 iterations over input size 100mb.

Mean is calculated by finding sum over all the time taken for 100 iterations of MD5 computation and then dividing it by 100. Median is calculated by first sorting the seconds array which stores the computation time for each MD5 iteration. Now median is the average of 50th and 51st element in the seconds array.

The mean and median for performing hash using MD5 is stated below:



```
struve@bn19-39: ~/Downloads/parkhe-assign2
File Edit View Search Terminal Help
[80] MD5 Elapsed time: 0.672219337
[81] MD5 Elapsed time: 0.663355154
[82] MD5 Elapsed time: 0.67691527
[83] MD5 Elapsed time: 0.652470498
[84] MD5 Elapsed time: 0.661029812
[85] MD5 Elapsed time: 0.651525371
[86] MD5 Elapsed time: 0.783745981
[87] MD5 Elapsed time: 0.656907441
[88] MD5 Elapsed time: 0.66376268
[89] MD5 Elapsed time: 0.666591149
[90] MD5 Elapsed time: 0.665115535
[91] MD5 Elapsed time: 0.692756852
[92] MD5 Elapsed time: 0.655216756
[93] MD5 Elapsed time: 0.653725133
[94] MD5 Elapsed time: 0.647661332
[95] MD5 Elapsed time: 0.656686386
[96] MD5 Elapsed time: 0.660634584
[97] MD5 Elapsed time: 0.654099811
[98] MD5 Elapsed time: 0.651734109
[99] MD5 Elapsed time: 0.659541559

Median for MD5 0.6643041375000001
Mean of MD5 0.6933980306500002
```

Figure 8: MD5 hash: Mean and median for computation

Mean of MD5 = 0.693398030650

Median of MD5 = 0.664348413

2.6 SHA1 Hash

Fig.9 shows the time taken for computing hash value using SHA1 algorithm. Fig.10 shows the mean and median of SHA1 hash algorithm for 100 iterations over input size 100mb.

Mean is calculated by finding sum over all the time taken for 100 iterations of SHA1 computation and then dividing it by 100. Median is calculated by first sorting the seconds array which stores the computation time for each SHA1 iteration. Now median is the average of 50th and 51st element in the seconds array.

The mean and median for performing hash using SHA1 is stated below:

Mean of SHA1 = 0.8839792637400002

Median of SHA1 = 0.863254081

2.7 SHA256 Hash

Fig.11 shows the time taken for computing hash value using SHA256 algorithm. Fig.12 shows the mean and median of SHA256 hash algorithm for 100 iterations over input size 100mb.

Mean is calculated by finding sum over all the time taken for 100 iterations of SHA256 computation and then dividing it by 100. Median is calculated by first sorting the seconds array which stores the computation time for each SHA256 iteration. Now median is the average of 50th and

```
stiruve@bn19-39: ~/Downloads/parkhe-assign2
File Edit View Search Terminal Help
[0] SHA1 Elapsed time: 1.194658682
[1] SHA1 Elapsed time: 0.950414264
[2] SHA1 Elapsed time: 0.865316721
[3] SHA1 Elapsed time: 0.886208977
[4] SHA1 Elapsed time: 0.871539843
[5] SHA1 Elapsed time: 0.859847815
[6] SHA1 Elapsed time: 0.87529168
[7] SHA1 Elapsed time: 0.874303025
[8] SHA1 Elapsed time: 0.861555065
[9] SHA1 Elapsed time: 0.847169937
[10] SHA1 Elapsed time: 0.926867727
[11] SHA1 Elapsed time: 0.984531485
[12] SHA1 Elapsed time: 0.860814983
[13] SHA1 Elapsed time: 0.874241236
[14] SHA1 Elapsed time: 0.854757087
[15] SHA1 Elapsed time: 0.857286085
[16] SHA1 Elapsed time: 0.854028997
[17] SHA1 Elapsed time: 0.870125683
[18] SHA1 Elapsed time: 0.902334685
[19] SHA1 Elapsed time: 0.854546532
[20] SHA1 Elapsed time: 0.89901449
[21] SHA1 Elapsed time: 0.857898344
[22] SHA1 Elapsed time: 0.860519126
[23] SHA1 Elapsed time: 0.875568477
```

Figure 9: SHA1 hash: Input size 100mb

```
stiruve@bn19-39: ~/Downloads/parkhe-assign2
File Edit View Search Terminal Help
[80] SHA1 Elapsed time: 0.905694937
[81] SHA1 Elapsed time: 0.852571113
[82] SHA1 Elapsed time: 0.857680108
[83] SHA1 Elapsed time: 0.864417141
[84] SHA1 Elapsed time: 0.859056682
[85] SHA1 Elapsed time: 0.931755398
[86] SHA1 Elapsed time: 0.901924955
[87] SHA1 Elapsed time: 0.899807429
[88] SHA1 Elapsed time: 0.911820094
[89] SHA1 Elapsed time: 0.873538476
[90] SHA1 Elapsed time: 0.875576334
[91] SHA1 Elapsed time: 0.905403127
[92] SHA1 Elapsed time: 0.898056655
[93] SHA1 Elapsed time: 0.879706334
[94] SHA1 Elapsed time: 0.921641795
[95] SHA1 Elapsed time: 0.910481778
[96] SHA1 Elapsed time: 0.865713727
[97] SHA1 Elapsed time: 0.937463612
[98] SHA1 Elapsed time: 0.876547627
[99] SHA1 Elapsed time: 0.956388195
Mean of SHA1 0.89405861762
Median for SHA1 0.8783884395
```

Figure 10: SHA1 mean and median for 100 iterations: Input size 100mb


```
stiruve@bn19-39: ~/Downloads/parkhe-assign2
File Edit View Search Terminal Help
[0] SHA256 Elapsed time: 1.52136007
[1] SHA256 Elapsed time: 1.362270811
[2] SHA256 Elapsed time: 1.379889712
[3] SHA256 Elapsed time: 1.335846444
[4] SHA256 Elapsed time: 1.325848
[5] SHA256 Elapsed time: 1.373555815
[6] SHA256 Elapsed time: 1.340163329
[7] SHA256 Elapsed time: 1.353729843
[8] SHA256 Elapsed time: 1.37892307
[9] SHA256 Elapsed time: 1.338583104
[10] SHA256 Elapsed time: 1.413087291
[11] SHA256 Elapsed time: 1.343663234
[12] SHA256 Elapsed time: 1.365587257
[13] SHA256 Elapsed time: 1.343855626
[14] SHA256 Elapsed time: 1.337629519
[15] SHA256 Elapsed time: 1.373119712
[16] SHA256 Elapsed time: 1.324908924
[17] SHA256 Elapsed time: 1.3300769
[18] SHA256 Elapsed time: 1.333485692
[19] SHA256 Elapsed time: 1.421577846
[20] SHA256 Elapsed time: 1.312990574
[21] SHA256 Elapsed time: 1.340359357
[22] SHA256 Elapsed time: 1.385167064
[23] SHA256 Elapsed time: 1.373860464
```

Figure 11: SHA256 hash: Input size 100mb

```
stiruve@bn19-39: ~/Downloads/parkhe-assign2
File Edit View Search Terminal Help
[79] SHA256 Elapsed time: 1.435862034
[80] SHA256 Elapsed time: 1.3794999
[81] SHA256 Elapsed time: 1.344459124
[82] SHA256 Elapsed time: 1.321503933
[83] SHA256 Elapsed time: 1.364123883
[84] SHA256 Elapsed time: 1.382572825
[85] SHA256 Elapsed time: 1.356083892
[86] SHA256 Elapsed time: 1.414664136
[87] SHA256 Elapsed time: 1.356278336
[88] SHA256 Elapsed time: 1.402109079
[89] SHA256 Elapsed time: 1.337558345
[90] SHA256 Elapsed time: 1.325340478
[91] SHA256 Elapsed time: 1.314559673
[92] SHA256 Elapsed time: 1.311677197
[93] SHA256 Elapsed time: 1.318494041
[94] SHA256 Elapsed time: 1.313406667
[95] SHA256 Elapsed time: 1.312325209
[96] SHA256 Elapsed time: 1.310397148
[97] SHA256 Elapsed time: 1.314019099
[98] SHA256 Elapsed time: 1.312276157
[99] SHA256 Elapsed time: 1.313889224
Mean of SHA256 1.39801832965080004
Median for SHA256 1.356133233
```

Figure 12: SHA256 mean and median for 100 iterations: Input size 100mb

51st element in the seconds array.

The mean and median for performing hash using SHA256 is stated below:

Mean of SHA256 = 1.3980183296500004

Median of SHA256 = 1.356133233

2.8 Digital Signature using SHA256 and RSA 4096

```
generated the required Sign key
Public and Private keypair is ready
-----
[0] Signing time0.867780794
[0] verifying time0.787368712
result = true
[1] Signing time0.828975698
[1] verifying time0.895405829
result = true
[2] Signing time0.841243228
[2] verifying time0.8094753
result = true
[3] Signing time0.829402717
[3] verifying time0.822039379
result = true
[4] Signing time0.832702502
[4] verifying time0.808194949
result = true
[5] Signing time0.841059866
[5] verifying time0.799730742
result = true
[6] Signing time0.878885394
[6] verifying time0.795408721
result = true
[7] Signing time0.838620818
[7] verifying time0.822475918
```

Figure 13: Digital signature Input size 100mb

Fig.13 shows time taken for computing digital signature using SHA256 and RSA 4096 on an input file of size 100mb. Fig.14 shows the mean and median of digital signature algorithm for 100 iterations over input size 100mb.

Mean is calculated by finding sum over all the time taken for 100 iterations of digital signature computation and then dividing it by 100. Median is calculated by first sorting the seconds array which stores the computation time for each digital signature iteration. Now median is the average of 50th and 51st element in the seconds array.

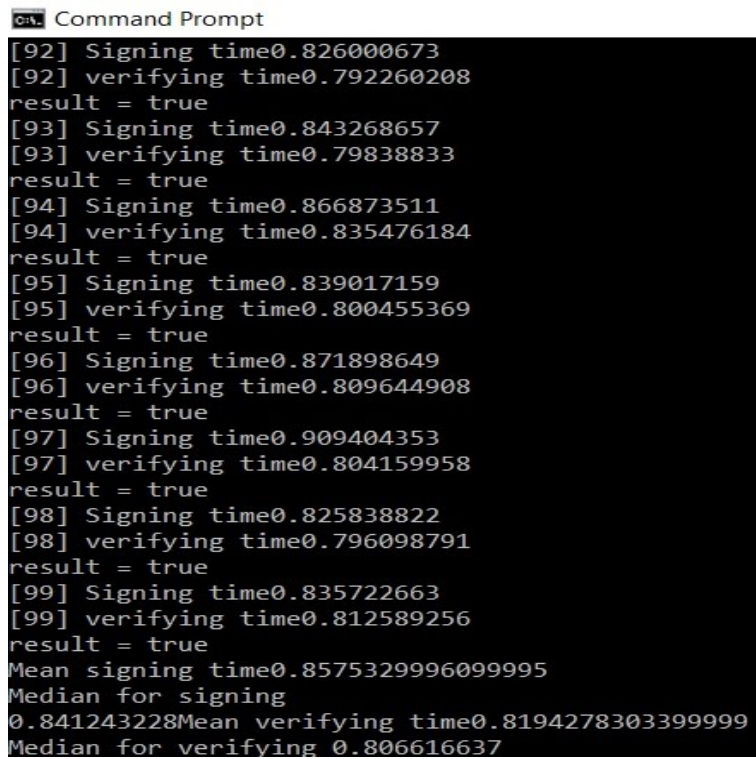
The mean and median for using digital signature is stated below:

Mean Of Signing time = 0.8575329996099995

Median of signing = 0.841243228

Mean of verifying time = 0.8194278303399999

Median for verifying time = 0.806616637



```
C:\> Command Prompt
[92] Signing time0.826000673
[92] verifying time0.792260208
result = true
[93] Signing time0.843268657
[93] verifying time0.79838833
result = true
[94] Signing time0.866873511
[94] verifying time0.835476184
result = true
[95] Signing time0.839017159
[95] verifying time0.800455369
result = true
[96] Signing time0.871898649
[96] verifying time0.809644908
result = true
[97] Signing time0.909404353
[97] verifying time0.804159958
result = true
[98] Signing time0.825838822
[98] verifying time0.796098791
result = true
[99] Signing time0.835722663
[99] verifying time0.812589256
result = true
Mean signing time0.8575329996099995
Median for signing
0.841243228Mean verifying time0.8194278303399999
Median for verifying 0.806616637
```

Figure 14: Digital signature mean and median for Input size 100mb

3 Performance analysis of cryptography protocols

3.1 Encryption and decryption algorithms

We have used AES 128, AES 256, RSA 1024, and RSA 4096 for encrypting and decrypting our input file.

1. **Discussion of performance encryption algorithms:**

RSA uses very complex mathematical computations (modular exponentiation) as compared to AES. Therefore it will always take more time to perform encryption than AES. As RSA 4096 encrypts 3 times bigger block than that of RSA 1024, it will take more time to encrypt. AES 256 uses 256 bit key to encrypt whereas AES 128 uses only 128 bit key to perform encryption. As AES 256 uses a bigger key, it will take more time to encrypt as compared to AES 128. And hence, RSA 4096 takes the highest time for encryption followed by RSA 1024 and then followed by AES256 and finally AES128.

Following are mean encryption time taken by all encryption algorithms ranked in increasing order:

Mean encrypting time for RSA 4096 = 103.8631590870

Mean encrypting time for RSA 1024 = 67.3641296055

Mean encrypting time for AES 256 = 0.87976189004

Mean encrypting time for AES 128 = 0.8054710887003

2. **Discussion of performance decryption algorithms:**

RSA uses very complex mathematical computations (modular exponentiation) as compared to AES. Therefore it will always take more time to perform decryption than AES. As RSA 4096 decrypts 3 times bigger block than that of RSA 1024, it will take more time to decrypt. AES 256 uses 256 bit key to decrypt whereas AES 128 uses only 128 bit key to perform decryption. As AES 256 uses a bigger key, it will take more time to decrypt as compared to AES 128. Hence, RSA 4096 takes the highest time for decryption followed by RSA 1024 and then followed by AES256 and finally AES128.

Following are mean decryption time taken by all decryption algorithms ranked in increasing order:

Mean decrypting time for RSA4096 = 7409.3225869569

Mean decrypting time for RSA1024 = 1248.13734225506

Mean decrypting time for AES256 = 0.90639052212

Mean decrypting time for AES128 = 0.86136988536

3. **Discussion of performance difference of RSA 1024 and RSA 4096 in encryption and decryption:**

The KeyPairGenerator class in the javax.crypto library will always output a smaller public key exponent and a bigger private key exponent. And this difference in the key exponents results in exponential computational time difference in encryption and decryption of messages. Hence, every RSA encryption will take less time as compared to the time taken for decryption.

3.2 Comparison of algorithms for computing hash

We use MD5, SHA1, SHA256 for computing hash over the given input file.

Mean for computing hash using MD5 = 0.6882856291099998

Mean for computing hash using SHA1= 0.8839792637400002

Mean for computing hash using SHA256 = 1.3980183296500004

We observe that SHA256 takes more time and followed by SHA1 and finally MD5 in computing hash over a large input file. The reason for this is MD5 produces 128 bit hash over the entire file by splitting it into block size 128 bits and iterating it for 64 rounds. While, SHA1 produces 160 bit hash over the entire file by splitting it into block size of 160 bits and iterating it for 80 rounds. Similarly SHA256 produces 256 bit hash over the entire file by splitting it into block size of 256 bits and iterating it for 64 rounds. Hence SHA256 takes more time to compute the hash.

3.3 Analyzing performance of Digital Signature done using SHA256 and RSA4096

Mean for signing using digital signature= 0.85753299.

Mean for verifying the digital signature= 0.819427830.

Here we use SHA256 to generate the hash over the file and next sign it with RSA-4096. Signing is done by RSA-4096 private key and verification is done by RSA-4096. Signing and verifying time is however less when we compare it with encryption and decryption time with RSA-4096 because of the size of the input given.

For digital signature we sign over the hash which is just 256 bits while encryption and decryption is done over 100 mb.

We can observe that it takes more time in creating the digital signature than verifying the digital signature. It is because RSA uses very complex mathematical computations (modular exponentiation). The KeyPairGenerator class in the javax.crypto library will always output a smaller verifying exponent and a bigger signing exponent. And this difference in the key exponents results in exponential computational time difference in signing and verification of signature. Hence, every signature verification will take less time as compared to the time taken for signing.

4 Rank Order of Cryptography Algorithms

The tables 1, 2 and 3 presents each operation (enc/dec/mac/sign/verify) for each relevant algorithm in a rank order by median time.

Algorithm	Encryption	Decryption
AES 128	0.790856802	0.8436531469999999
AES 256	0.860085832	0.9063512915
RSA 1024	67.36412960550187	1248.1373422550696
RSA 4096	103.86315908	7409.32258695692

Table 1: Encryption schemes

Algorithm	Mac
MD5	0.656486605
SHA 1	0.863254081
SHA 256	1.356133233

Table 2: Mac Algorithms

Algorithm	Sign	Verify
SHA 256 and RSA 4096	0.841243228	0.806616637

Table 3: Signature Algorithms