Appendix A Encryption times for spatial files within DBFS

Table 1 shows encryption (*Enc.*) and decryption (*Dec.*) times of different sizes of spatial files using 256-bit AES encryption (32-character key) using three approaches:

- within ECMSDK using `DBMS_CRYPTO_TOOLKIT`,
- within ECMSDK using OpenSSL encryption tool, and
- within Windows NTFS using OpenSSL encryption tool.

The next three subsections include the encryption-decryption performance results from these approaches.

The first two approaches compare the encryption and decryption performance of files within ECMSDK using (a) the `DBMS_CRYPTO_TOOLKIT` within the Oracle 11g database (implemented in PL/SQL) against (b) the performance provided by the OpenSSL tool (a Windows executable, called externally using PL/SQL from our encryption scripts). The third approach compares the encryption of spatial files using the OpenSSL tool within ECMSDK versus those stored directly on the OS (i.e. Windows NTFS).

In all experiments, the tests were performed on a machine with the following specifications: an Intel 1.6 GHz i5-8265u processor with 8GB RAM on Dell Inspiron system and the software utilized consisted of Oracle 11g Database Release 2, running on Windows 10 Operating System.

GIS File	Filesize (Mb)	AES		OpenSSL on CMSDK		OpenSSL on NTFS	
		Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
france-points.shp	1.50	1.13	1.07	0.48	0.17	0.04	0.04
france-waterways.shp	6.32	4.73	4.49	0.55	0.29	0.06	0.05
france-natural.shp	11.67	8.74	8.30	1.15	0.45	0.11	0.07
indonesia-natural.shp	11.95	8.95	8.50	1.20	0.55	0.07	0.07
germany-points.shp	13.89	10.41	9.88	1.25	0.92	0.07	0.14
britain-waterways.shp	16.12	12.08	11.46	1.93	1.31	0.08	0.09
china-buildings.shp	26.53	19.87	18.87	2.25	1.90	0.12	0.34
india-natural.shp	32.13	24.07	22.85	3.81	2.38	0.13	0.37
china-natural.shp	33.75	25.28	24.00	4.08	2.68	0.14	0.48
india-waterways.shp	36.21	27.13	25.75	4.18	2.77	0.15	0.56

Table 1: Comparing three approaches of applying AES encryption and decryption of spatial files[1].

[1] The ESRI shapefiles used in the investigation are sourced from publicly available datasets (https://mapcruzin.com/download-free-arcgis-shapefiles.htm).

A.1 Within ECMSDK using DBMS_CRYPTO_TOOLKIT

Figure A1 shows the encryption and decryption time for spatial files in ECMSDK being computed from within PL/SQL code using SQLplus. One by one, each file's ID is passed to the encryption procedure `encrypt()` along with the encryption key. The time capturing has been coded within PL/SQL code.

To retrieve the file id of a file, the following SQL can be used.

```
SELECT co.content, doc.name
FROM ecmsdk.odmv_document doc, ecmsdk.odm_contentobject co
WHERE doc.name LIKE '%.shp' AND doc.CONTENTOBJECT = co.id;
```

The id of files retrieved using the above query is used as parameter in the encryption and decryption processes:

```
set echo on
set serveroutput on

call encrypt(23402,'passwordpasswordpasswordpassword');
commit;
call decrypt(23402,'passwordpasswordpasswordpassword');
commit;
```

This PL/SQL encryption and decryption procedure is available within the "performanceResults\Approach1_WithinECMSDKUsingDBMSCrypto_PLSQL" folder in the GitHub project repository: https://github.com/sharmapn/DBFSFileCrypto/

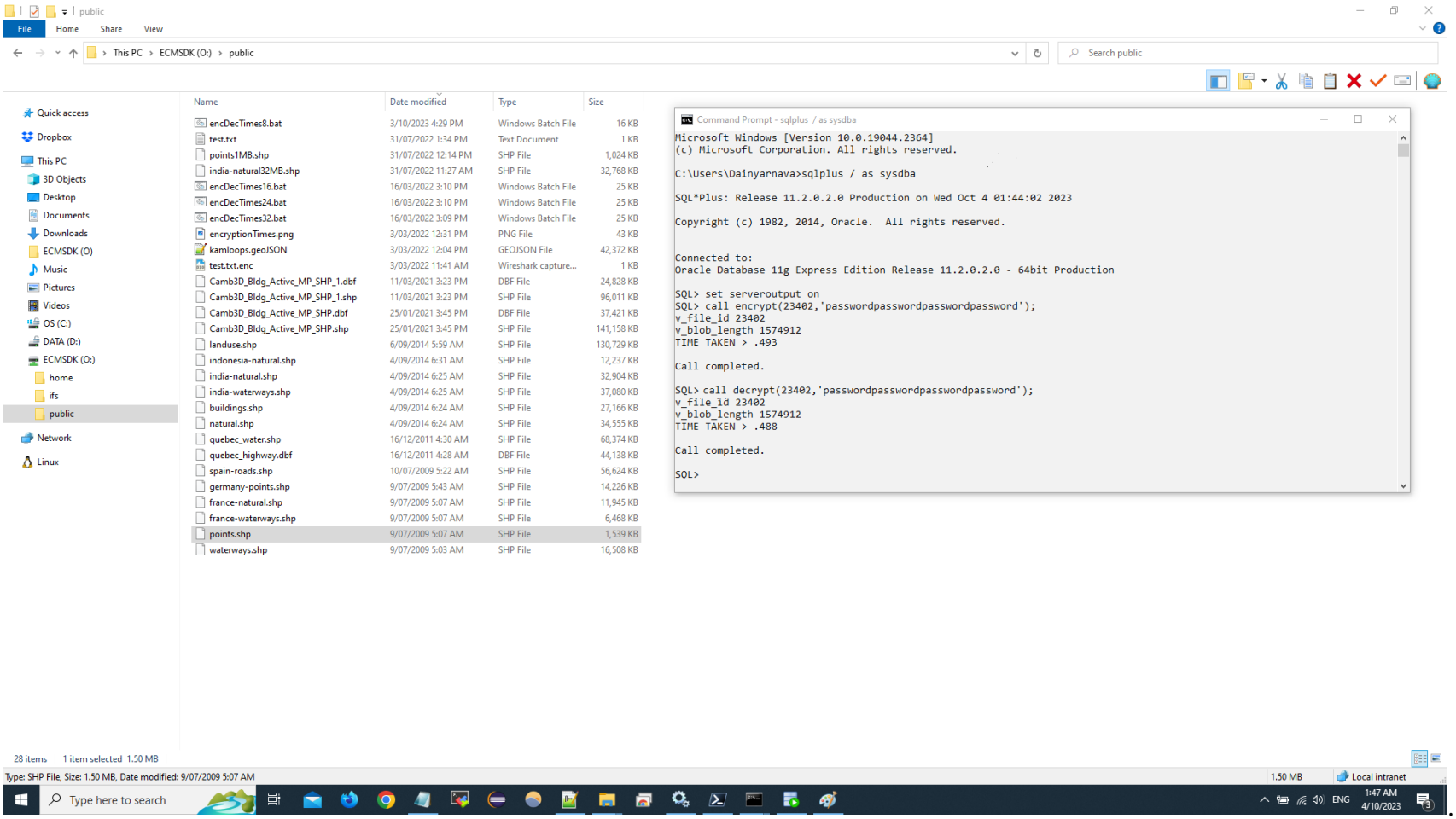


Figure A.1: Encryption and Decryption time for spatial files in ECMSDK being computed using SQLPlus.

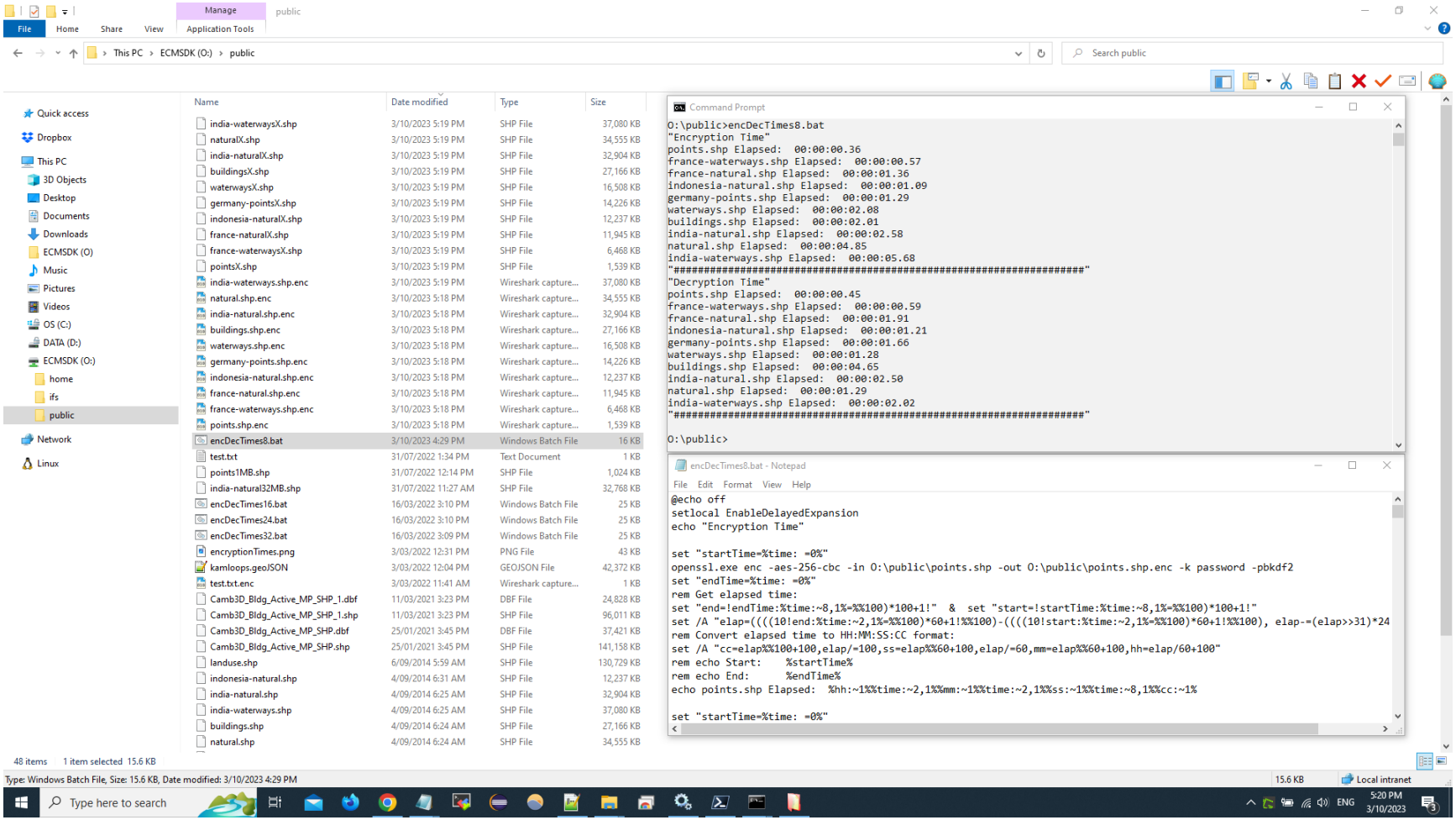
A.2 Within ECMSDK using OpenSSL Toolkit

Figure A2 shows the encryption and decryption time for spatial files in ECMSDK being computed using the OpenSSL tool. A batch script encrypts and then later decrypts each file one by one for different lengths of the encryption keys. The filename and the encryption key is passed as argument. The time capturing has been coded within the batch script.

Here is one encryption and decryption pair of code:

```
echo "Encryption Time"

set "startTime=%time: =0%"
openssl enc -aes-256-cbc -in O:\public\points.shp -out O:\public\points.shp.enc -k
password -pbkdf2
set "endTime=%time: =0%"
set "end=!endTime:%time:~8,1%=%%100)*100+1!"  &  set
"start=!startTime:%time:~8,1%=%%100)*100+1!"
set /A "elap=((((10!end:%time:~2,1%=%%100)*60+1!%%100)-
((((10!start:%time:~2,1%=%%100)*60+1!%%100), elap-=(elap>>31)*24*60*60*100"
set /A
"cc=elap%%100+100,elap/=100,ss=elap%%60+100,elap/=60,mm=elap%%60+100,hh=elap/60+100"
echo points.shp Elapsed:
%hh:~1%%time:~2,1%%mm:~1%%time:~2,1%%ss:~1%%time:~8,1%%cc:~1%;



echo "Decryption Time"

set "startTime=%time: =0%"
openssl enc -d -aes-256-cbc -in O:\public\points.shp.enc -out O:\public\pointsX.shp -k
password -pbkdf2
set "endTime=%time: =0%"
rem Get elapsed time:
set "end=!endTime:%time:~8,1%=%%100)*100+1!"  &  set
"start=!startTime:%time:~8,1%=%%100)*100+1!"
set /A "elap=((((10!end:%time:~2,1%=%%100)*60+1!%%100)-
((((10!start:%time:~2,1%=%%100)*60+1!%%100), elap-=(elap>>31)*24*60*60*100"
rem Convert elapsed time to HH:MM:SS:CC format:
set /A
"cc=elap%%100+100,elap/=100,ss=elap%%60+100,elap/=60,mm=elap%%60+100,hh=elap/60+100"
rem echo Start:    %startTime%
rem echo End:      %endTime%
echo points.shp Elapsed:
%hh:~1%%time:~2,1%%mm:~1%%time:~2,1%%ss:~1%%time:~8,1%%cc:~1%
```

The full batch scripts for the encryption and decryption procedures are available within the "performanceResults/Approach2_WithinECMSDKUsingOpenSSL" folder at the GitHub project repository: https://github.com/sharmapn/DBFSFileCrypto/



.

Figure A.2: Encryption and Decryption time for spatial files in ECMSDK using OpenSSL command line tool.

A.3 Within NTFS using OpenSSL Toolkit

Figure A3 shows the encryption and decryption time for spatial files in ECMSDK being computed from within PL/SQL code using SQLplus. One by one, each file's ID is passed to the encryption procedure `encrypt()` along with the encryption key. The time capturing has been coded within PL/SQL code.

To retrieve the file id of a file, the following SQL can be used.

```
echo "Encryption Time"

set "startTime=%time: =0%"
openssl enc -aes-256-cbc -in D:\OpenSSLencryptionTimes\points.shp -out
D:\OpenSSLencryptionTimes\points.shp.enc -k password -pbkdf2
set "endTime=%time: =0%"
rem Get elapsed time:
set "end=!endTime:%time:~8,1%=%%100)*100+1!"  &  set
"start=!startTime:%time:~8,1%=%%100)*100+1!"
set /A "elap=((((10!end:%time:~2,1%=%%100)*60+1!%%100)-
((((10!start:%time:~2,1%=%%100)*60+1!%%100), elap-=(elap>>31)*24*60*60*100"
rem Convert elapsed time to HH:MM:SS:CC format:
set /A
"cc=elap%%100+100,elap/=100,ss=elap%%60+100,elap/=60,mm=elap%%60+100,hh=elap/60+100"
rem echo Start:    %startTime%
rem echo End:      %endTime%
echo points.shp Elapsed:
%hh:~1%%time:~2,1%%mm:~1%%time:~2,1%%ss:~1%%time:~8,1%%cc:~1%;

echo "Decryption Time"

set "startTime=%time: =0%"
openssl enc -d -aes-256-cbc -in D:\OpenSSLencryptionTimes\points.shp.enc -out
D:\OpenSSLencryptionTimes\pointsX.shp -k password -pbkdf2
set "endTime=%time: =0%"
rem Get elapsed time:
set "end=!endTime:%time:~8,1%=%%100)*100+1!"  &  set
"start=!startTime:%time:~8,1%=%%100)*100+1!"
set /A "elap=((((10!end:%time:~2,1%=%%100)*60+1!%%100)-
((((10!start:%time:~2,1%=%%100)*60+1!%%100), elap-=(elap>>31)*24*60*60*100"
rem Convert elapsed time to HH:MM:SS:CC format:
set /A
"cc=elap%%100+100,elap/=100,ss=elap%%60+100,elap/=60,mm=elap%%60+100,hh=elap/60+100"
rem echo Start:    %startTime%
rem echo End:      %endTime%
echo points.shp Elapsed:
%hh:~1%%time:~2,1%%mm:~1%%time:~2,1%%ss:~1%%time:~8,1%%cc:~1%
```

The full batch scripts for the encryption and decryption procedures are available within the "performanceResults/Approach3_WithinNTFSUsingOpenSSL" folder at the GitHub project repository: https://github.com/sharmapn/DBFSFileCrypto/

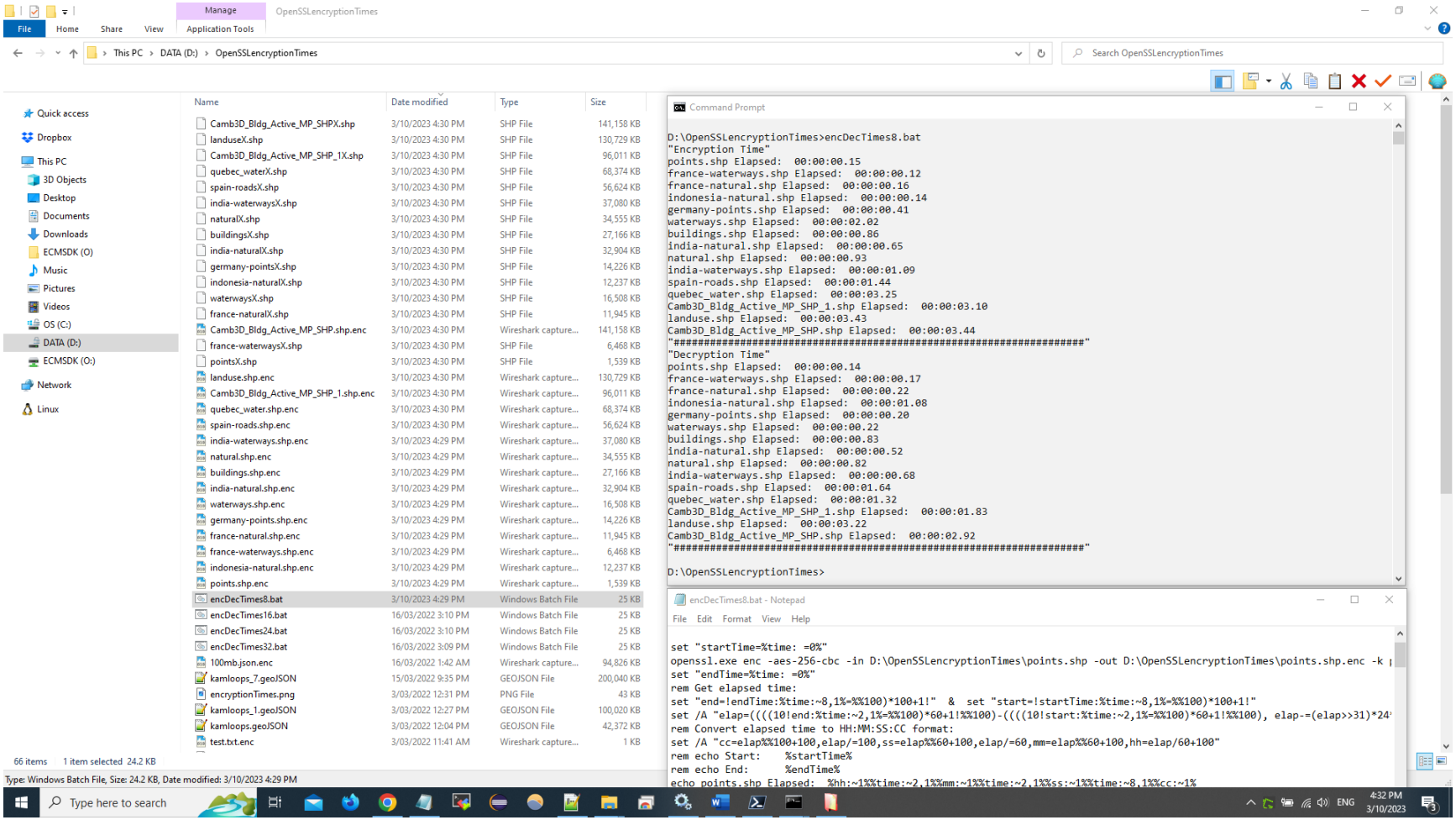


Figure A.3: Encryption and Decryption time for spatial files in Windows NTFS using OpenSSL command line tool.