

<b>Course Name:</b>	<b>Information Security (116U01L602)</b>	<b>Semester:</b>	VI
<b>Date of Performance:</b>	03 / 02 / 2025	<b>DIV/ Batch No:</b>	A-4
<b>Student Name:</b>	Hyder Presswala	<b>Roll No:</b>	16010122151

**Title: Application of RSA Algorithm for various security services like confidentiality, authentication, signature, non-repudiation and integrity**

**Objectives:**

To write a program to convert plain text into cipher text using Caesar cipher and Transposition cipher

**Expected Outcome of Experiment:**

**CO1** :- Explain various security goals, threats, vulnerabilities and controls  
**CO2** :- Apply various cryptographic algorithms for software security

**Books/ Journals/ Websites referred:**

1. Security in Computing
2. Cryptography and Network Security
3. Cryptography and Network Security: Principles and Practice

**Pre Lab/ Prior Concepts:**

**New Concepts to be learned:**

**Abstract:**

## Related Theory:

### Implementation Details with Output :

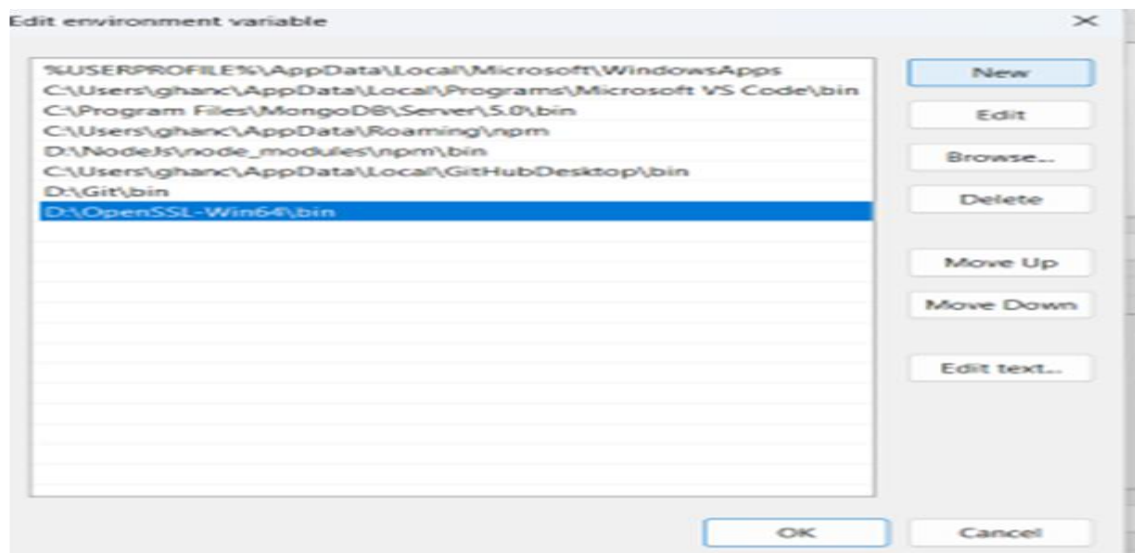
#### 1. Enlist all the Steps followed and various options explored

##### I. Download and Install OpenSSL

1. Download the Download Win64 OpenSSL v3.0.8 (EXE) from the website.

Download Win32/Win64 OpenSSL		
Download Win32/Win64 OpenSSL today using the links below!		
File	Type	Description
Win64 OpenSSL v3.4.0 Light <a href="#">EXE</a>   <a href="#">MSI</a>	5MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.4.0 (Recommended for users by the creators of <a href="#">OpenSSL</a> ). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.4.0 <a href="#">EXE</a>   <a href="#">MSI</a>	221MB Installer	Installs Win64 OpenSSL v3.4.0 (Recommended for software developers by the creators of <a href="#">OpenSSL</a> ). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.4.0 Light <a href="#">EXE</a>   <a href="#">MSI</a>	4MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v3.4.0 (Only install this if you need 32-bit OpenSSL for Windows). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.

2. Install and update the Path variable in Environment Variables.



## II. Generating RSA private /public key pair

1. Generate key pair using: opensslgenrsa -out mykey1.key 1024
2. Extract public key from key pair using: opensslrsa -in mykey1.key - pubout  
-out mypublickey.key

```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Student>cd Desktop

C:\Users\Student\Desktop>cd InfoSec

C:\Users\Student\Desktop\InfoSec>openssl rsa -in mykey1.key - pubout -out mypublickey.key
rsa: Extra option: "pubout"
rsa: Use -help for summary.

C:\Users\Student\Desktop\InfoSec>openssl genrsa -out mykey1.key 1024

C:\Users\Student\Desktop\InfoSec>openssl rsa -in mykey1.key -pubout -out mypubkeybhai.key
writing RSA key
```

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC+I1/I2f6M37qH/sIe+S3dap/2
UeW1Vw7LDH8Du0dXwUEzeLeHZq4RUHh3CV0chWIZ1o0d6qRTWfSHuA/ngDgPvBXg
UHF58p6oDTxAqd2OvFgW01QNPDT/yYpsXefvFZp1hc9BZ5zGFw00Q2J0m1fsB8SsX
LMS+GwLy/oR4fLwLFQIDAQAB
-----END PUBLIC KEY-----
```

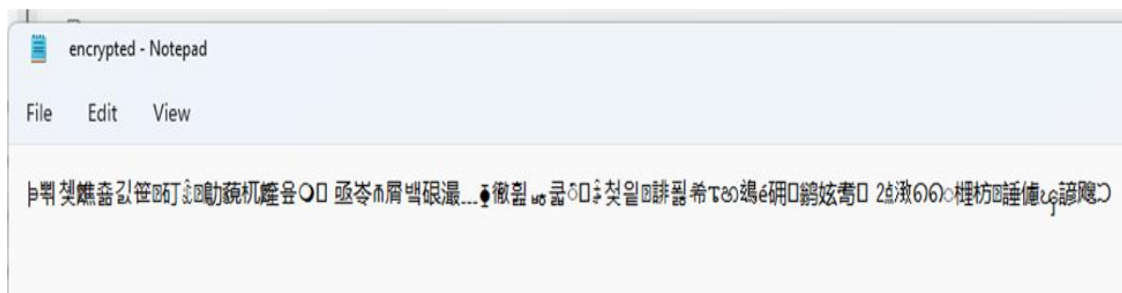
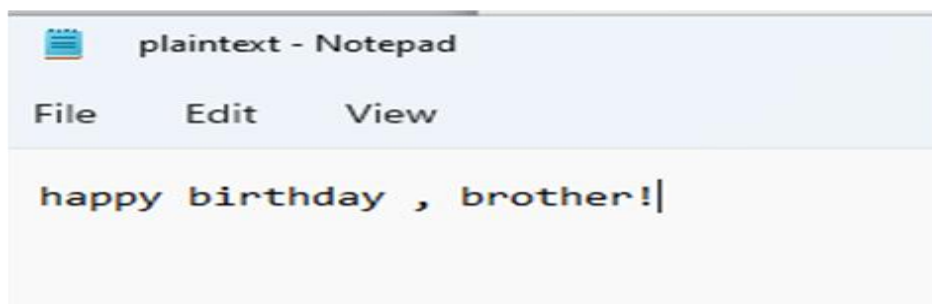
```
-----BEGIN PRIVATE KEY-----
MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggljAgEAAoGBAL41L8jZ/ozfuof+
wh77k12qn/ZR5a7WbssMcEO7R1fBQTN4t4dmrhFQeHcJXRyFYjOKjR3qpFNYVIfA
D+uAOA+8FeBQ1/kGnqNPECp3Y68wBbSJA0803/J1mxdS+8VmmKFz0FnnfYXBczY
k6aV+wEFKxdQzz4bAvL+hHh8vAsVAgMBAAECgYEA14IcG7zQ5N90R4sH9qKSj8W9
X4hR/R58ZuI2kYwd2ecEB1dhGqmduT1Zw4Z0WpQILRdR6g0sC7d5Nq4XQE4xUxuz
hkd1norQeupb4mPaDZ91VuQ0MFicZrbXGMT8To6AGd/DRnWjUKdsS8G3PbsOMg3
9bXadRKChb/VNTxcccUCQQO7tIM9nS3/Fqy35kUQYU4qE4tghmm47QK3Gd3x/0yS
vSRz/kmHLf1ZX625Tt3IcLjeIkmZ1gCzR1xQ1ryIUHAPAkEAwwC4Yx5PC2+cjRFf
73EwK20Y1Fdb6200EwPwS4qorrcrYCAQe7yzeHgwWz1ftYz3cBH3c+zKx/kr7d3H
XEkumJAHZSrShI1eaYa1jfqQfCN1RhS6vcg+aCxOv+z4b3VMibv/8nErnRTP+uk
qQC8xAgRTz+X/19ekbHzQC1W3tZFNQ3Adg/jA0Dh6Y1++vF8rOd1DwFR/F93k3D4
sMtK71v9mI3kQV/SAHm46SjvvQ5/hkxGVKnGkKw1MeoC0/rvWm1tuQ3BA3fSm8fc
EQwzfZn2h1GQIB3NdmYd339zLDUMsI8T6F33M1XaD1p6aX70khfUuBf1kZzE1d4A
NVQqowmX+Ln8OU=
-----END PRIVATE KEY-----
```

### III. Public Key Encryption

Using the public key we can encrypt the text in plaintext.txt using:

```
opensslpkeyutl - encrypt -in plaintext.txt -inkeymypublickey.key -pubin -out  
encrypted.txt
```

```
C:\Users\Student\Desktop\InfoSec>openssl pkeyutl -encrypt -in plaintext.txt -inkey mypubkeybhai.key -pubin -out encrypted.txt  
C:\Users\Student\Desktop\InfoSec|
```



## IV. Hash Functions

Create a file to be hashed.

Use the command: `openssl dgst -sha256 hashfile.txt`

```
C:\Users\Student\Desktop\InfoSec>openssl dgst -sha256 hashingkafilebro.txt
SHA2-256(hashingkafilebro.txt)= e9d188f5fb9c7dcae9be7d5209025a3a454835807a1afc411f3b90dee0985263
C:\Users\Student\Desktop\InfoSec>|
```

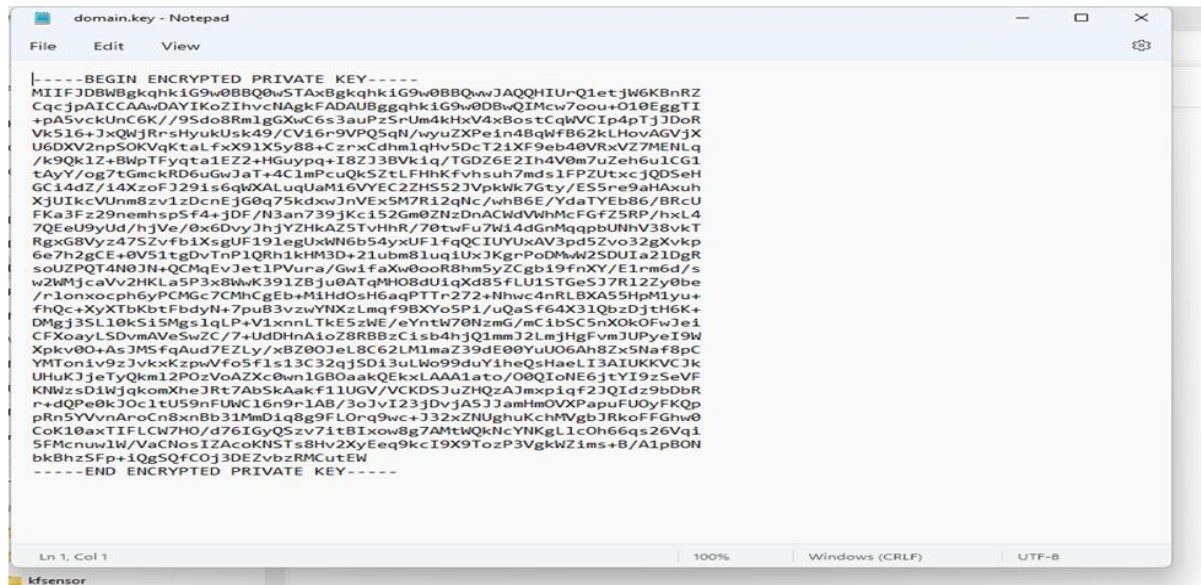


## V. Certificate Creation

1. Create a private key using:

`openssl genrsa -des3 -out domain.key 2048`

```
C:\Users\Student\Desktop\InfoSec>openssl genrsa -des3 -out domain.key 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
C:\Users\Student\Desktop\InfoSec>|
```



```
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFJDBWBgkqhkiG9w0BBQwwJAQHHIURQ1etJW6K8nRZ
CqcJpAICCAwDAYIKoZIhvcNAgkFADAUBggqhkiG9w0DBwQIMcw7oou+010EggTI
+pA5vckUnC6K//9Sdo8RmlgXwC6s3auPzSrUm4kHxV4xBostCqWVCIP4ptJJDOR
Vk516+JxQWJRrsHyukUsk49/CV16r9VPQ5qN/wyuzXPein4BqWfB62kLHovAGVjX
U6DXV2npSOKVqKtaLfxX91X5y88+CzrxChmlqHv5DcT21XF9eb40VRxVZ7MENLq
/k9Qk1Z+BWpTFyqta1EZ2+HGuyppq+I8ZJ3BVk1q/TGDZ6E2IH4V0m7uZeh6u1CG1
tAyY/og7GmcKRD6uGwJaT+4C1mPcuQk5ZtLFHhkfVhshuh7mds1FFZUtxcJQDSeH
GC14dZ/14XzofJ291s6qWALuqUaM16VYEC2ZHS52JVPkV7Gty/ES5re9ahAuh
XjUIKcVUnm8zv1zDcnEjG0q75kdxwJnVEx5M7R12qNc/whB6E/YdaTYEb86/BRcU
FKa3Fz29nemhsp5f4+jDF/N3an739jKc152Gm0ZNzDnACWdVWhMcFGfZ5RP/hxL4
7QEeU9yUd/hjVe/0x6DvyJhJYZHkAZ5TvhHr/70twFu7W14dGnMqpbUNhV38vkT
RgxG8Vyz47S2vfb1XsGUF19legUxwN6b54yxUF1fqQCUIYUxAV3pd5Zvo32gXvKp
6e7h2gCE+0V51tgDvTnP1QRh1kHM3D+21ubm81uq1UxJKgrPoDmW2SDUIa21DgR
soUZPQT4H0JN+QCMqEvJettIPVura/Gw1faXw0oR8hm5yZCgb19fnXY/E1rm5d/s
w2WMjcaVv2HKLAsP3x8WwK391ZBju0ATqMHO8dU1qXd85FLU15TGeSj7R1Z2zy0be
/r1onxocph6yPCMGc7CMhCgEb+M1HdOsH6aqPTTr272+Nhwc4nRLBXA55HpM1yu+
fhQc+XyXTbKbtFbdyN+7puB3vzwYNXzLmqf9BXyo5P1/uQa5f64X31QbzDjtH6K+
DMgJ3SL0kS15Mgs1qLP+V1xnnLTkE5zWE/eYntw70NzmG/mC1bSC5nX0k0FwJei
CFXoayLSDvmAveSwZC/7+UdDHnAioZ8RBBzC1sb4hjQ1mmJ2LmjHgFvmJUPyeI9W
Xpkv08+ASJMSfqaud7EZLy/xB200Jel8C62LM1maZ39dE00YU006Ah8Zx5Naf8pC
YMTon1v9zJvKxKzpuF5F1s13C32qj5D13uLWo99duYiHeQsHaeL13AIUKKVCJk
UHuK3jeTyQkm12PozVoAZXc0wn1GBOaakQEKLAAA1ato/00QI0NE6jtYI9zSeVF
KNWzsdIwJqkcmXheJrt7AbSkAakf11UGV/VCKDSJuzHQZa3mxpiqf2JQIdz9bDbR
r+dQPe0k3Oc1tU59nFUWC16n9r1AB/3oJvI23jDvJASJJamHmOVXPapuFUOyFKQp
pRn5YVvnAroCn8xnbB31MmD1q8g9FLOrq9wc+332xZNUghuKchMVgbJRkoFFGhw0
CoK10axTIFLCW7H0/d76IGyQ5zv71tB1xow8g7AMtWQKNCYNKgLiC0h66qs26Vq1
5FMcnuw1W/VacNnosIZacoKISTs8hv2XyEeq9kcI9X9TozP3VgkWZims+B/A1p80N
bkBhzSFp+1QgSQFC0j3DEZvzbRMCutEW
-----END ENCRYPTED PRIVATE KEY-----
```

2. Create a certificate using the key using:

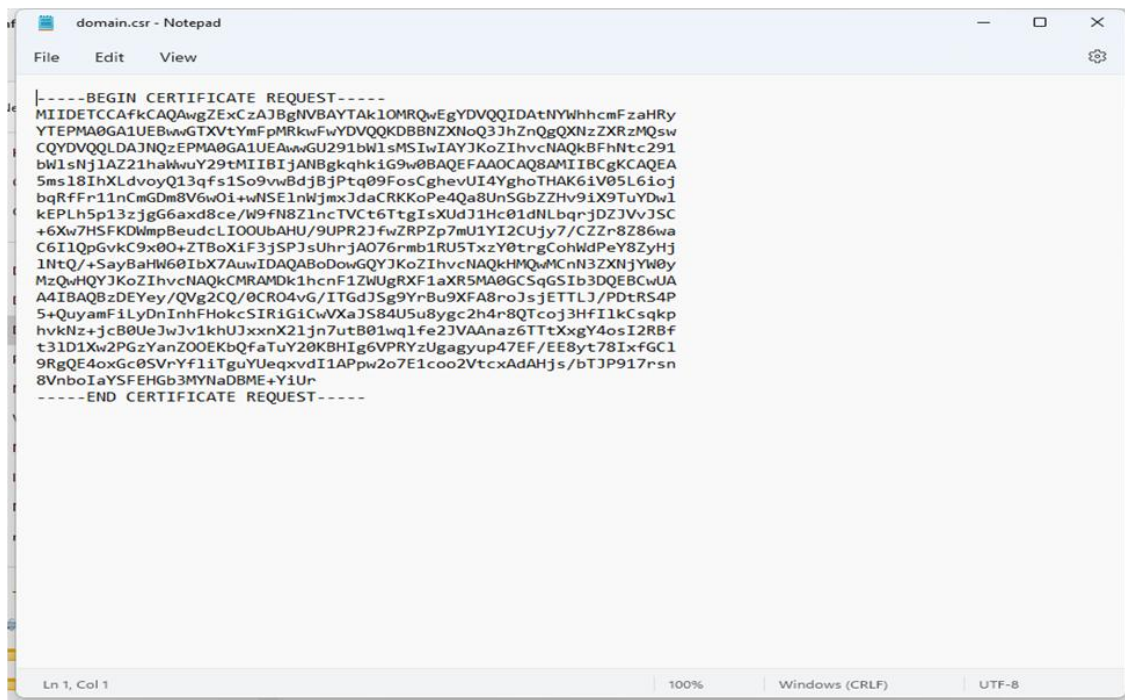
`openssl req -key domain.key -new -out domain.csr`

Enter all the necessary information for the certificate

```
C:\Users\Student\Desktop\InfoSec>openssl req -key domain.key -new -out domain.csr
Enter pass phrase for domain.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:Maharashtra
Locality Name (eg, city) []:Mumbai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MeshCraft Assets
Organizational Unit Name (eg, section) []:MC
Common Name (e.g. server FQDN or YOUR name) []:Soumil
Email Address []:msoumil69@gmail.com

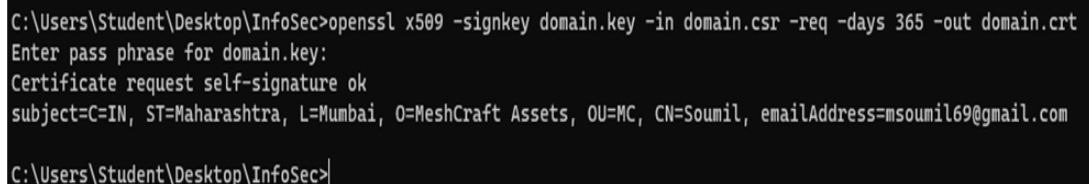
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:swescam234
An optional company name []:Marquee Equity

C:\Users\Student\Desktop\InfoSec>
```



3. Create a self signed certificate used:

```
openssl x509 -signkey domain.key -in domain.csr -req -days 365 -out domain.crt
```



```
C:\Users\Student\Desktop\InfoSec>openssl x509 -signkey domain.key -in domain.csr -req -days 365 -out domain.crt
Enter pass phrase for domain.key:
Certificate request self-signature ok
subject=C=IN, ST=Maharashtra, L=Mumbai, O=MeshCraft Assets, OU=MC, CN=Soumil, emailAddress=msoumil69@gmail.com
C:\Users\Student\Desktop\InfoSec>
```

4. View the certificate using:

```
openssl x509 -text -noout -in domain.crt
```



```
66:9b:23:16:d7:09:a6:7d:32:4e:94:46:d6:b2:c9:c6:b2:e7:
43:fc:d9:93:25:2f:d5:7c:f6:e2:1e:a6:2e:4a:0d:3d:61:d8:
c7:76:bd:a8:9e:42:41:c4:08:7a:ae:29:71:37:8a:b6:be:bf:
b9:89:86:02:60:84:15:4f:6d:63:a4:89:8a:1c:a0:02:89:6d:
07:25:6c:99:5c:f5:a5:c5:6b:1e:8d:6e:40:0e:98:00:29:d0:
90:e3:dc:a5:1a:1d:cc:26:05:49:33:5e:71:72:b8:d3:f3:ea:
2f:e6:77:40:09:dc:89:ea:3f:4e:29:92:45:2c:94:6e:81:f5:
c4:fd:3d:33:8f:2c:95:71:0d:54:f3:bd:84:76:4f:e8:20:d0:
b5:44:ea:eb:5e:ea:d7:0c:fd:a4:b9:cd:b8:06:c0:5b:c5:f9:
61:73:08:67:f6:61:66:c0:80:97:5d:fa:f6:34:a7:e6:c3:b7:
05:fe:e4:30:f6:17:53:f2:43:dd:82:75:ed:52:35:5b:88:7b:
a9:db:19:82:e1:0f:cb:30:06:ea:50:cb:1e:45:82:17:e4:fc:
ca:b5:a1:fd
```

C:\Users\Student\Desktop\InfoSec>

C:\Users\Student\Desktop\InfoSec>openssl x509 -text -noout -in domain.crt

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

14:d0:47:70:5c:ac:d7:da:e7:b4:d6:3d:1f:04:75:aa:f0:6b:bd:79

Signature Algorithm: sha256WithRSAEncryption

Issuer: C=IN, ST=Maharashtra, L=Mumbai, O=MeshCraft Assets, OU=MC, CN=Soumil, emailAddress=msoumil69@gmail.com

Validity

Not Before: Jan 21 09:45:28 2025 GMT

Not After : Jan 21 09:45:28 2026 GMT

Subject: C=IN, ST=Maharashtra, L=Mumbai, O=MeshCraft Assets, OU=MC, CN=Soumil, emailAddress=msoumil69@gmail.com

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

```
00:e6:6b:25:f0:88:57:2d:db:e8:c9:0d:77:a9:fb:
35:4a:8f:6f:c0:17:63:06:33:ed:ab:4f:45:a2:c0:
a0:85:eb:d4:23:86:20:86:84:c7:00:ae:a2:57:4e:
4b:ea:2a:23:6e:a4:5f:16:bd:75:9c:29:86:0e:6f:
15:eb:03:a2:fb:03:52:12:59:d6:8e:6c:49:75:a0:
91:28:aa:0f:7b:84:1a:f1:49:d2:19:b6:59:1e:ff:
62:5f:d4:ee:60:3c:25:90:43:cb:87:9a:75:df:38:
e0:1b:a6:b1:77:c7:1e:fd:6f:5f:37:c6:65:9d:c4:
d5:0a:de:93:b6:02:2c:5d:47:49:d4:77:34:d5:d3:
4b:6e:aa:e3:0d:92:55:bc:94:82:fb:a5:f0:ec:74:
85:28:35:a6:a4:17:ae:75:c2:c8:38:e5:1b:00:75:
3f:f5:43:d1:d8:97:f0:65:13:d9:a7:b9:94:d5:82:
36:09:48:f2:ef:f0:99:66:bf:19:f3:ac:1a:0b:a2:
25:42:91:af:90:2f:71:d0:ef:99:4c:1a:17:88:5d:
e3:48:f2:6c:52:1a:e3:00:ee:fa:ae:66:f5:45:4e:
53:c7:36:34:b6:b8:02:a2:15:9d:3d:e6:3c:67:21:
e3:94:db:50:ff:e4:9a:c8:16:87:5b:ad:08:6d:7e:
c0:bb
```

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Subject Key Identifier:

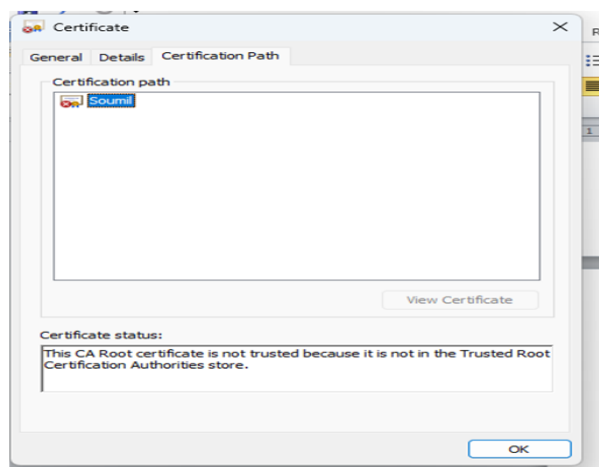
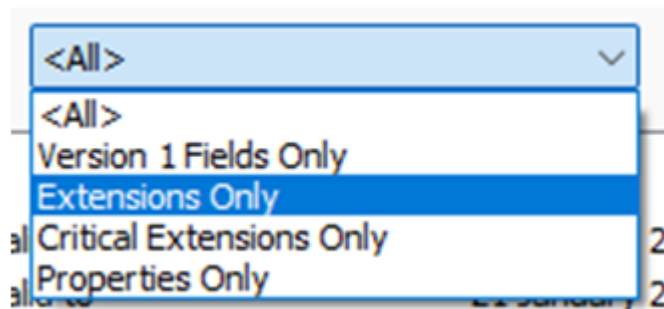
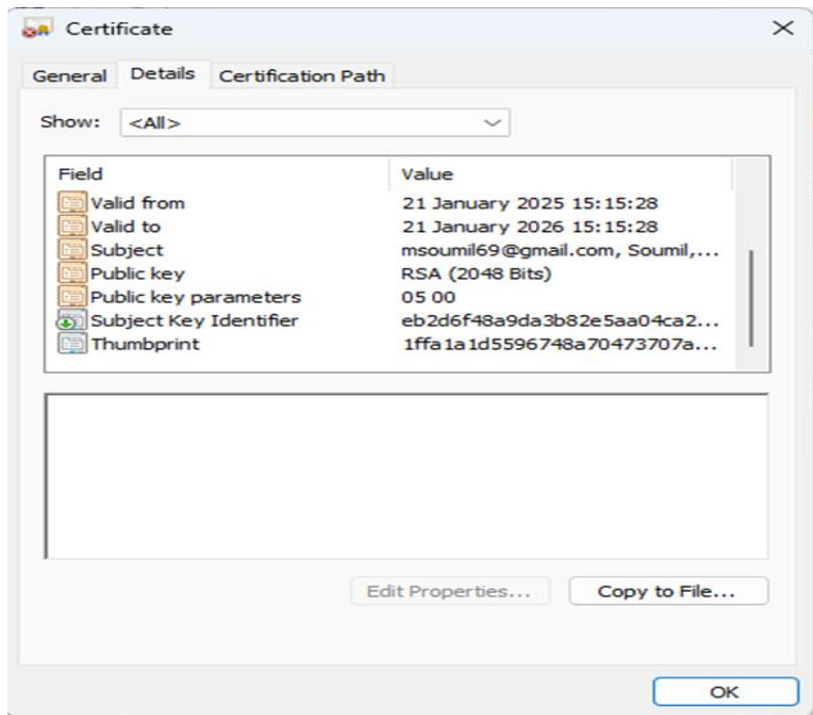
EB:2D:6F:48:A9:DA:3B:82:E5:AA:04:CA:2D:D1:7E:36:64:4C:16:EA

Signature Algorithm: sha256WithRSAEncryption

Signature Value:

```
8b:f2:ea:3a:8d:4b:27:18:75:4a:6a:5e:eb:a0:cf:d1:82:85:
b3:c7:95:b4:85:98:71:2f:e0:2e:b5:ed:39:ee:f2:31:10:01:
```





## VI. Digital signature

### 1. Create private and public key using:

-openssl genrsa -aes128 -passout pass<phrase>: -out private.pem 4096

-openssl rsa -in private.pem -passin pass:<phrase> -pubout -out public.pem

```
C:\Users\Student\Desktop\InfoSec>openssl genrsa -aes128 -passout pass:soumaiya -out private.pem 4096
```

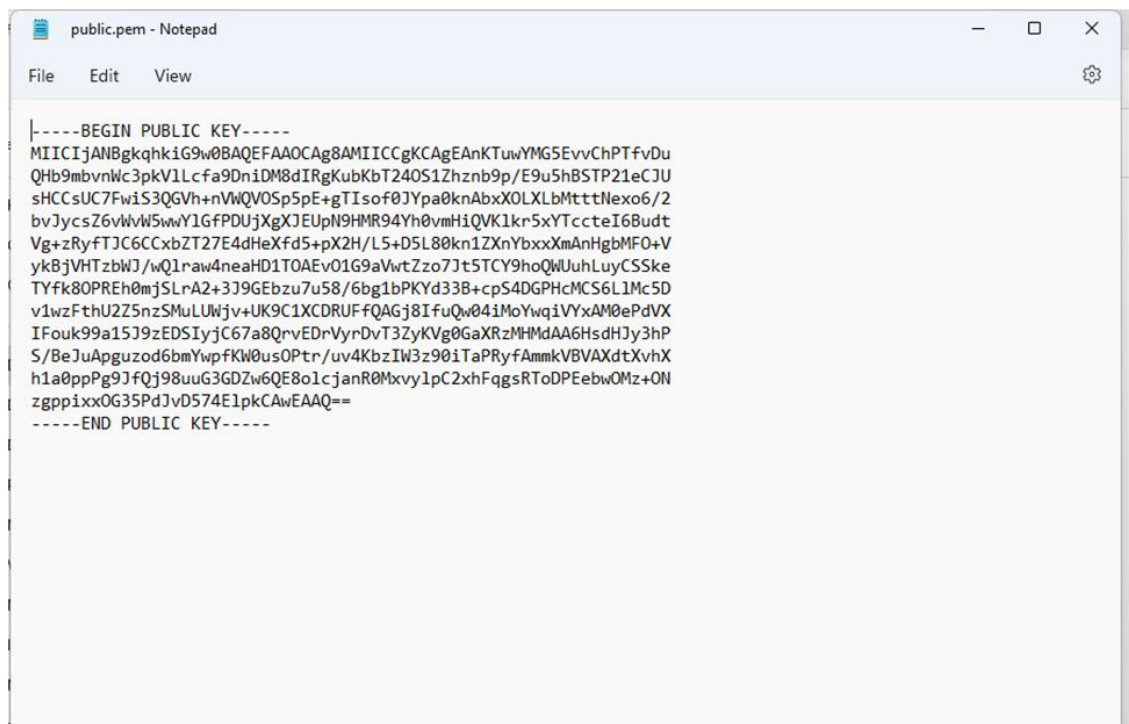
```
C:\Users\Student\Desktop\InfoSec>
```

```
private.pem - Notepad
File Edit View

-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIEZTBGqkqhkiG9w0BBQowJAKBgqkqhkiG9w0BBQowJAQoQ1f1MaGPMz7B0wLb
UHQFA3CCAwDAAYTKoZiHvCNagKFA0adBg1ghk8BZQMEAQ1EENAYY5c8wu5gtFQq
r/oFwoeEggg1Qkf1Lt5jx5pt4y1jAYk7DaY+CHTAV7PN4my+eRvMIzsu39qYAmHds
mk/8yEt131Vn613xvCXkN18Bx8B1gAaK50cUxmRHHkNvDRG1LEqmhUgRQ01A3Y
KNpAKARKbDvON286ef03a0UEC6x7f/AShmDk4bpuJMGkCXHKZLV1xwQ11I/Nkz9m
TL3hg52pm7vP6t31jovdJD/z899yH11zxb8DRtNggSnUELYKXx8JvYAnKEDPn11
q2d4ztb11625p+pl1QC+1/NB5D950wLbFRAMTVRKy9Q9g094JN0c8LnyTELL8YU
B2zq2v0kNpHYAFb3jnPQGT8fAV8vCq1XenB3AfejxvTRV5GAYy9AwmmN81znag1BH
x0m1xwU979+25g2w3/944xPpELXsvEI+wxzP75QBssuaDE1+oGwBpK/r/2TrPw8
dXcy0021Ywd422z0MVN6IAVBfxcJ2qk3nyLt+kqKMO8R5m12Fyzc1Ww22B2K752
IPXbeF/kU+opc81P4tE1M1CB/BAV3x3b2A3j0ecVku6ynYg4ch3C1TC0GL73UyRz
myRfXjVhb1QnkF32q5UKYxeB0JHHC156Px9NA1NRBw0AAkvJWtJgK3v5tBXmgD
LV1jY8r9yB1855LP1T4fncsXg54HK5EF6y0B8yH91Y19dV29e8APF/vyTiw
TEtugKxxpG/C4/A4TrjJ8EIKW10rnrQ/323Gu14ZS1MX27oyJAKR6FC3tWCCp05
NtUUEK+em+GH13t82aUBwvJm2Ahr/VYrcpv3a0/pz3Qhw3tmeIvvXgPRKRhvcGmM
tmvYy0HEEKmKcLVV118Ahp7nwB1CQvZu/kvz8BeyQ58QeY91ydy3koTbcZ88kpy
FL4bIB8j5NymBNCOrFXKD5e0obVA1MN7eboFFZe21rg/T8YJ6T2b8fYqos+5ERmoJ
BMB1TCkRKk1fgoQ8C8moTnf3pBL11b83ceq0Ygpb8cVznBB3X39yZ+Hvdk0/L
Jg11rI6f1p408eyCQJwh+hwcn5n5bE2RthhspeVdAv7YIMxevba4aw23F4ELgR3
1CA/KRw9kU/5DRH0gg2BqKcDEcVnntxGUB89SH08Thuj1p2k3H+2J9N4xrf021vR
vBneG3XvquQAs3HjTr5tp5sYex1vy59yz24p/n5U1bWcJL1m+315mr3HfAR0L9k2
h/COPEASxyU8BDFowXPKoTJw4B+3E5K41yHTk8IQqd8R7PCKSKqYB8h9X1wF
I69x10d+r7/2Frlv+TP/dQAexh3OqqX6IQcMt/Rk2HvTCRKH0sUPVVGUwa/FwB
4n1cM8dkNyxU+UHRHARq5/521q0cXN57V2xUwZ60JnKm1cWpYB72HqPwDp1
Qc4u0Qw7yemvYFF6rBLphq131Qbq134bV7Y9R+57n212fwe3h1TEPy866UJHO2F6D
sB55QT0MnI02qLYu1ghy51jFVR00zo+ossxyvTmoTCNV0BEO9DZ8h+DOH1HBquye
hXGhnySkARHXCRAacbv2F78d316yEE1MeSU0gwhsX05qCukK4OZOLnezcS0Zy
fJ5epo/1k6bmK2tfw6hW8xsvEQG5qC3dz59z/VNV4D/z+/LQwdJok3K0LSV1K1U
61np956pvsQR3S07zA22Invad3uWY/qz59cm1ngHz/oHo39V51b5PCbbn1jEB861G
wGB691FSP5an31UBBQfBUUuQsAw8T28z99ef9Ec+3pC+hBndIXfN-NPT/7d5a1m
qs2f350fnn6+8Hkn35sXNRDn/BG40D+51He90RGQw89H5gDqB2ge372wskzp
CMBBrrnzkeA1H98c4K85gKfeZqMxPCSpL3hsTh3966s+zN8B21g15/nzyofyuu
LEgekP3v113o95+kP8p90tMnIERcf2KMuy17vjqqNzX1o4BuxGmfVJw4pkAH15JM
K/b+33EHFSLTz0n1GfYCPK089B5oukcNPR87v74thq112eUz1d4a21GE8m5f8K5
+3JQ793/CTEYEm/sQZCmQbA8ouUtdzu7FT5u+00Pj3TfVjpkfcwtvB6C51v+3DUK
v3/engUwYCV3n325Vqntak6x+hgGnQ76+XdqCQsK1G1hRt+furF33tQKf+CvW1cB
5/37DFINIVv8u51mi/0R61D3g2mFKP25jYCWac/WEtVvdm7R5eKQ+hau4qu0+HW
+/3/8gU0ZF3Q51dWR/nTnRf/3uHc1Yd7cvf1e0K3P74fW6yev1h2T3HDB8QTY8
rNCULr1Xb0gVOTYzSGCOVG1WYtn+1wnwLzPGTQVVGWPPUw4Y5r6T78ZyVkyXcM
Zx3rbmkaw0P/N64fM1NQ7TLTveXhe6XsdJubkPU1mQ0B90Qqb5eJQx2/5pFmHC
Y0s1QuVvAXyP77ybBhD3k4sk1pudZ1/ffFfYKEROle70R3o8hW0mN8fD3Buc6U
9dc88D1VfqtDuz8P8GHHBC1AIWfWxrn4mu1aay7vCPXf9p5V3wckUHZE1Rh1+Ux5
fA123V6E0RYffdnq41CCSKcySuct4w1oA2z/z47v1dhRegrdfgFB3MA3TZY8ct2
XDLwFVDVfKxYHCKPZaa16yx31HPIXPeyAk1Gvt4IDc26PwRBpkTbCb16iul/V5k
c40Qxc2N5Beyze0pXv3U5Xhqd/26J23YfzboYzX06KTU9d2ru/9C1q1/3RnK
+z8Yw+1g6KcXf1ku5vDUJ31LFR1qM5gtYkn097jmdh0et6iV/A1kQ51KvT4zvJed
6dydQKXrQnp18Cg8BwB82UL+9m1LQjbtncUw26c14/GuN8TB28qKq01R1T
YNKR01R000QGHXKXTR61lvk1HTOM55AHtgw5cnvd3J41GY4JMT00VU1G8A3YUJT
BAUMR55vfeCfLWAH5kdNw17Zp0/QaMVC1JNweJET8211oNkyTos1ezMIDV1K8I
80Ea1j1gryd0PyT/Zdzp72dr3w9hy35H1YmHhcqA1E8DKPxeV7U0w6Y8y5Xbo3m
Puz6ompFjCohkDRMk2Tg0B5C1rQmg37I/BfNB8JPuuybtGy8021zmKa3xnZL7RC
ybs/cOQ2y4zmU1Lg6Fe2Q2Yfd177xw1vVt79H30uN6NCVq70Scuz0e
-----END ENCRYPTED PRIVATE KEY-----
```

```
C:\Users\Student\Desktop\InfoSec>openssl rsa -in private.pem -passin pass:soumaiya -pubout -out public.pem
writing RSA key
```

```
C:\Users\Student\Desktop\InfoSec>
```



```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAGeAnKTuwYMG5EvvChPTfvDu
QHb9mbvnlWc3pkV1Lcfa9DniDM8dIRgKubKbT240S1Zhzn9p/E9u5hBSTP21eCJU
sHCCsUC7FwiS3QGVh+nVwQV0Sp5pE+gTIsof0JYpa0knAbxX0LXLbMtttNexo6/2
bvJycsZ6vWw5wY1GfPDUjXgXJEU9N9HMR94Yh0vmHiQVK1kr5xYtcteI6Budt
Vg+zRyfTJC6CCxbZT27E4dHeXfd5+pX2H/L5+D5L80kn1ZXnYbxxXmAnHgbMF0+V
ykBjVHTzbWJ/wQ1raw4neaHD1T0AEvO1G9aVwtZzo7Jt5TCY9hoQWUuhLuyCSSke
TYfk80PREh0mjSLrA2+3J9GEbzu7u58/6bg1bPKYd33B+cpS4DGPHeMCS6L1Mc5D
v1wzFthU2Z5nzSMuLUWjv+UK9C1XCDRUFfQAGj8IfuQw04iMoYwqiVYxAM0ePdVX
IFouk99a15J9zEDSIyjC67a8QrvEDrVyrDvT3ZyKVg0GaXRzMHMdaA6HsdHJy3hP
S/BeJuApguzod6bmYwpfKW0usOPtr/uv4KbzIW3z90iTaPRyfAmmkVBVAXdtXvhX
h1a0ppPg9JfQj98uuG3GDZw6QE8o1cjanR0MxvylpC2xhFqgsRT0DPEebwOMz+ON
zgppixxOG35PdJvD574E1pkCAwEAAQ==
-----END PUBLIC KEY-----
```

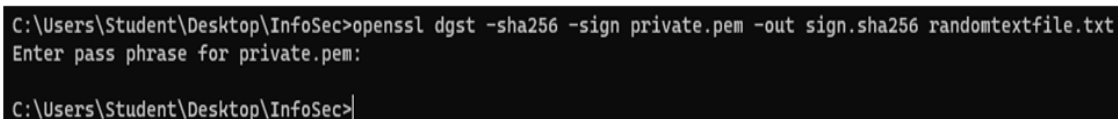
2. Create a text file



```
randomtextfile - Notepad
File Edit View
Where does our brain work on the insertion?
```

3. Generate the signature of a file using:

```
-openssl dgst -sha256 -sign<private key> -out
/tmp/sign.sha256<file> -openssl base64 -in
/tmp/sign.sha256 -out <signature>
```



```
C:\Users\Student\Desktop\InfoSec>openssl dgst -sha256 -sign private.pem -out sign.sha256 randomtextfile.txt
Enter pass phrase for private.pem:
C:\Users\Student\Desktop\InfoSec>
```

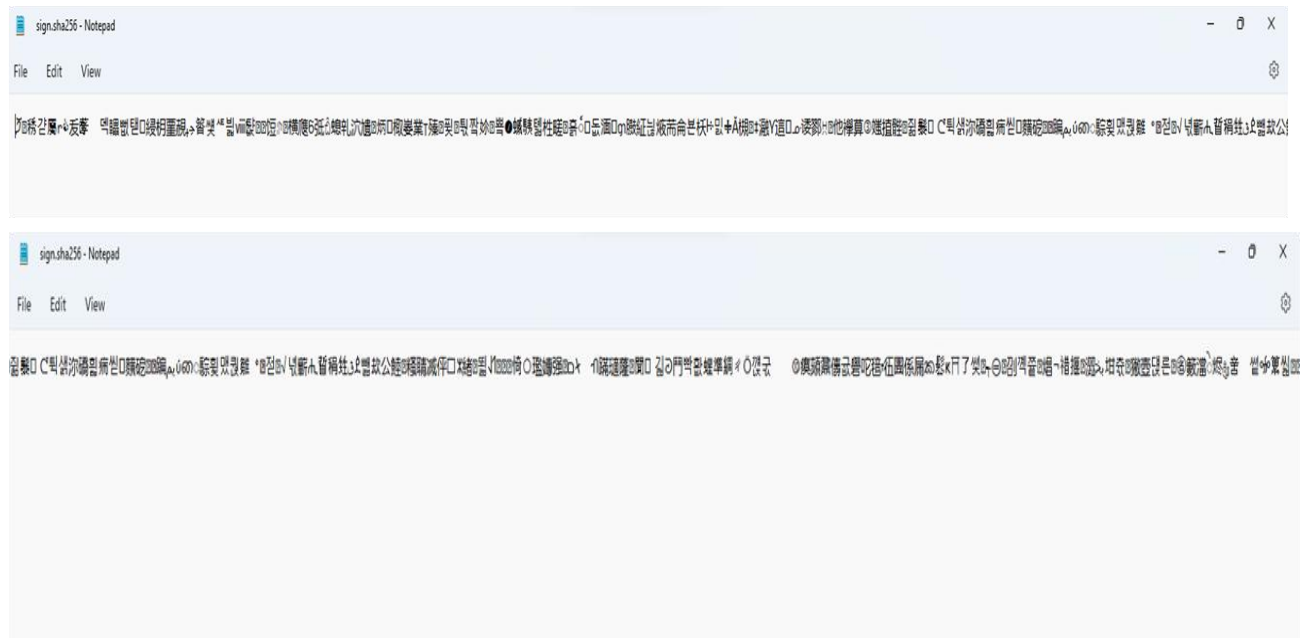
4. Verify the signature using:

```
-openssl base64 -d -in <signature> -out /tmp/sign.sha256 <file>
```

```
C:\Users\Student\Desktop\InfoSec>openssl base64 -d -in randomtextfile.txt  
C:\Users\Student\Desktop\InfoSec>|
```

-openssl dgst -sha256 -verify <pub-key> -signature /tmp/sign.sha256

```
C:\Users\Student\Desktop\InfoSec>openssl dgst -sha256 -verify public.pem -signature sign.sha256 randomtextfile.txt  
Verified OK  
C:\Users\Student\Desktop\InfoSec>|
```



**Conclusion:**

Thus, in this experiment the concept of RSA algorithm for various security services like confidentiality, authentication, signature, non-repudiation and integrity was understood and applied by developing a website.

**Post-Lab Questions:**

2.1 In the RSA algorithm,  $p=7$ ,  $q=11$  and  $e=13$ , then what will be the value of  $d$ ?

In the **RSA algorithm**, the value of  **$d$**  is calculated as follows:

**Given:**

- $p=7$
- $q=11$
- $e=13$

**Step 1: Compute  $n$**

$$n=p \times q=7 \times 11=77$$

**Step 2: Compute  $\phi(n)$**

$$\phi(n)=(p-1) \times (q-1)=(7-1) \times (11-1)=6 \times 10=60$$

**Step 3: Compute  $d$**

$d$  is the modular multiplicative inverse of  $e$  modulo  $\phi(n)$ , meaning:

$$d \times e \equiv 1 \pmod{\phi(n)}$$

$$d \times 13 \equiv 1 \pmod{60}$$

To find  $d$ , we need to solve:

$$d=e^{-1} \pmod{60}$$

This means finding  $d$  such that:

$$(13 \times d) \pmod{60}=1$$

Using the **Extended Euclidean Algorithm**:

We solve for d in:

$$13^{-1} \bmod 60$$

Using the Euclidean Algorithm:

1.  $60 \div 13 = 4$  remainder  $60 - (13 \times 4) = 8$
2.  $13 \div 8 = 1$  remainder  $13 - (8 \times 1) = 5$
3.  $8 \div 5 = 1$  remainder  $8 - (5 \times 1) = 3$
4.  $5 \div 3 = 1$  remainder  $5 - (3 \times 1) = 2$
5.  $3 \div 2 = 1$  remainder  $3 - (2 \times 1) = 1$
6.  $2 \div 1 = 2$  remainder  $2 - (1 \times 2) = 0$

Since the GCD is **1**, we can backtrack:

$$\begin{aligned} 1 &= 3 - (1 \times 2) \\ 1 &= 3 - (1 \times (5 - 1 \times 3)) = 2 \times 3 - 1 \times 5 \\ 1 &= 2 \times (8 - 1 \times 5) - 1 \times 5 = 2 \times 8 - 3 \times 5 \\ 1 &= 2 \times 8 - 3 \times (13 - 1 \times 8) = 5 \times 8 - 3 \times 13 \\ 1 &= 5 \times (60 - 4 \times 13) - 3 \times 13 = 5 \times 60 - 23 \times 13 \\ -23 \times 13 &\equiv 1 \pmod{60} \end{aligned}$$

Since d must be positive:

$$d = 60 - 23 = 37$$

Therefore  $d = 37$

2.2 Discuss various cryptanalysis attacks possible to be carried out on RSA

- **Integer Factorization Attack:** This is the most well-known attack on RSA. It involves factoring the large composite number N (which is the product of two large primes) to retrieve the private key. The difficulty of this attack depends on the size of NN.
- **Timing Attacks:** These attacks exploit the time taken by the RSA algorithm to perform certain operations. By analyzing the time variations, an attacker can deduce information about the private key.
- **Side-Channel Attacks:** These attacks exploit information leaked during the physical implementation of the RSA algorithm, such as power consumption, electromagnetic radiation, or even sound.
- **Chosen Ciphertext Attacks:** In this attack, the attacker chooses a ciphertext and attempts to decrypt it to gain information about the private key or plaintext.
- **Quantum Computing Attacks:** Although still theoretical, quantum computers could potentially break RSA by efficiently solving problems that are currently intractable, such as integer factorization.

2.3 Comment on drawbacks of RSA. Discuss solution(s) over the same.

### **Drawbacks of RSA**

**Somaiya Vidyavihar University**  
**K J Somaiya School of Engineering**

**1. Computational Complexity:**

- RSA requires significant computational resources, especially for key generation, encryption, and decryption. This can be a drawback for devices with limited processing power.

**2. Large Key Sizes:**

- To ensure security, RSA requires large key sizes (2048 bits or more), resulting in slower encryption and decryption processes, which can be inefficient for some applications.

**3. Vulnerability to Quantum Computing:**

- Future quantum computers could potentially break RSA by efficiently solving integer factorization problems, posing a significant risk to the long-term security of RSA.

**Solutions**

**1. Hybrid Cryptosystems:**

- Combine RSA with symmetric key algorithms (e.g., AES) to mitigate computational complexity. RSA can be used to securely exchange a symmetric key, which is then used for efficient data encryption and decryption.

**2. Elliptic Curve Cryptography (ECC):**

- ECC offers similar security to RSA but with much smaller key sizes, resulting in faster computations and reduced resource requirements. ECC is a potential alternative to RSA for many applications.

**3. Post-Quantum Cryptography:**

- Research into post-quantum cryptographic algorithms is ongoing. These algorithms are designed to be secure against quantum computing attacks. Implementing post-quantum cryptography can future-proof systems against quantum threats.