

**Name: Tooba Khan**  
**Entry Number: 2021JCS2245**  
**Report for Assignment 3 of SIL765**  
**Readme for Problem 2.**

## Problem 2:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

j3Y49vnU+0zMtMGs96lE8sunvFBNLt6xDDcDRFHii3phSrCPREshRwk5M5d
+AK7Z
VxxsmTg/I68ZH4pXQHnagIZWdr4+ArCcovbRIfQaQQu3nSDy3h29w5u10yj
nVSI+
GwIDAQABoxMwETAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBAUAA4I
BAQBZ
89HfzNSxDBMJ2RuWCGTVm6Ch3Y88Md1T2kbF8f+UKNfQV10XvWU/M5K8+mn
wUaV8
oTexI5otqLqQoa84LsvsWgtqK34Pxhgt/vyTdX5ZDC2r0ywAxoVi1WHyHpd
huZVb
9MCr3ZcoC7wZXbYTxIY4612hXnVBINGiqsMSe0iY1e6a4yThurzdy401vmz
sr4uX
wxNo/s43Wfca9KgSsJs0oqt7VygvN+611TFuyb5Kgimk/a8K0xpTnudaVVP
FD5WB
0l7bJzKZq3QY4aas+eU9PRps8GbEsFwTD9FfJd+LJFcjPBEL6Dyv8AcMatX
84835
nHtvDkMSP49ljzS0QsLU
-----END CERTIFICATE-----

Client Certificate received is Valid.
.....End of Phase 3.....
.....

.....Beginning of Phase 4.....
.....

Pre-Master-secret received.
The calculated master Secret is:
.....End of Phase 4.....
.....

.....End of Handshake.....
.....

.....Beginning of Record Protocol.....
.....

Message has been sent.
toobakhan@Toobas-MacBook-Air TLSA-3 % □
```

*Figure 1 Server Side (my\_server.py)*

```

r97
s6RmVoPn0BtbZsQU0xsvbarRafj8XH0jKXTMLy9R+0S0qyD0h37X1q2TtrAHq
22H
yjRa9fp5b2uEJtDmK9jek+Fyn63mLhdPCVKo6LQbEyBB10c24tu1HESNylg5/
/Ck
TODi1Mow
-----END CERTIFICATE-----

Server Certificate received is Valid.
Server has finished phase 2.
.....End of Phase 2.....
....

.....Begining of Phase 3.....
....

Sent certificate to server.
.....End of Phase 3.....
....

.....Begining of Phase 4.....
....

Pre-Master-secret sent.
The calculated master Secret is:
.....End of Phase 4.....
....

.....End of Handshake.....
.....

.....Begining of Record Protocol.....
.....

The message is authentic.
The recieved data is:- The OTP for transferring Rs 1,00,000 t
o your friend's account is 256345.
toobakhan@Toobas-MacBook-Air TLSA-3 % █

```

Figure 2 Client side(my\_client.py)

```
zsh + v [] 
toobakhan@Toobas-MacBook-Air TLSA-3 % python3 my_ttp.py
b'-----BEGIN CERTIFICATE REQUEST-----\nMIICmDCCAYACAQAwUzEL
MAkGA1UEBhMCSU4xDjAMBgNVBAgMBURlbGhpMQ4wDAYD\nVQQHDAVEZWxoA
TEPMA0GA1UECgwGU2VydMvYMRMwEQYDVQQDDApzZXJ2ZXIuY29t\nnMIIBIj
ANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwg/qvy0WIE1c+Vd+DH0y\
nucHUHjPmmDpGMjg2tRxrJkpmxyq1bUKQyJtQKopQTl0Yk+ZW8T5n4B0KHU
YWAR+0\ng/2dyhaNGDMAtz3vuZyz048TGqYsANUMtwMthIgidzDl7IEgDxi
gG6pr/rXUNbn3\nRhNPSQ/oCfQAM72rXcdwp4U4IqefED2IIm7T/YbAk09w
QpX9XttjmChmwLmx/IvC\nqTQS1lk/a2rveQMkokXWvtFyaIP9/nnFiMy+r
h9DTLSQGTyqWPZqdJ0XoctWgl2Z\nYt5AroARQxYwKEe90CjeRmFVCxmiEv
9oQCnqZgb1TbEAwox5VCk0vzySAsKRmmio\nnZQIDAQABoAAwDQYJKoZIhvc
NAQELBQADggEBABfjk6C/9wSMBdC54b7YJ/Nh8jG05\nnUYDF9iBDw46nbhS1
trCmFTA9KuQn70GagR75IJJ+XZLGB+d+ZQEcwg+8UmMf8WJk\nnS0ZAtyFsm
Ecp38A3/sywywnYYaN+UDlfQF3ibX7/03q1P7e8gdxbb13zj+krJd4\nnW0
ebpfLn5Pts2BSMEX20vyzkTx5TW1Lj+eGIm+VDMtviF6/LR5kcw0gXGHind
Tax\nnr2B2hZwcrbBeSIznnNIzbo4307d9NJDDX4+Sz6SCa9pD30Q38r1KCx
Ddwt0eHlpK\nnu3+XHZnu2r0/wUzfK7xRc/tTKE2HhWzi5Ps5b282A6aA5j/
A4XA9Qle3p9I=\n-----END CERTIFICATE REQUEST-----\n'
<cryptography.hazmat.backends.openssl.rsa._RSAPublicKey obj
ect at 0x1010e0d90>
b'-----BEGIN CERTIFICATE REQUEST-----\nMIICoTCCAYKCAQAwXDEL
MAkGA1UEBhMCSU4xFTATBgNVBAgMDFV0dGFyUHJhZGVz\nnaDEQMA4GA1UEB
wwHTHVja25vdzEPMA0GA1UECgwGQ2xpbmV0MRMwEQYDVQQDDApj\nnbGllbn
QuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA\nv/Z4Nqdh\
nXV2aI3+80LLdNKsU/GWzMhNdwtRS5mR0xXJXv3e5j\nrMJfRY0X56+LIN02t
0fdAvd\nnrq0K51JymCreT0V/F6MKzbFQoaP6R5zZxRzudSMXkWBQACJk0Xf
afxZ4mgZKXLF0\nnm6DeEv10Z5C6vpM96zPXZ+pd2gvnAVvD225H/2THW/LV
1AJGqfMoauHD1nv6FG73\nnMkz0ZvesyQBmj3Y49vnU+0zMtMGs96LE8sunv
FBNLT6xDdCDRFHii3phSrCPREsh\nnRwk5M5d+AK7ZVxxsmTg/I68ZH4pXQH
nagIZWdr4+ArCcovbRI\nfQaQQu3nSDy3h29\nnw5u10yjnVSI+GwIDAQABoAA
wDQYJKoZIhvcNAQELBQADggEBAIbMlGcBk40Pc7Qm\nn+MfGoxyLLSXWPKUv
4MJH4yTT32ILk0DwzGmNn0SAFZT7AYu5y0UKR2adlGSo6CmH\nn20ArGpNk4
abHi46uQibUvoGQL81S1LmJ/F3E80vd0AwI2zmFYVqUTKg\njohA2Kt/3\nnUx
3k3jSl6q7xnbEPNTt0w9ao8t1TmYR9CPwreia\nkf7Jy+Rpa5pgn/zxfW0FbX
U47\nnWhcTudNkb7UpLYGRLpDOW6XPQN4mMachv+0TJ2MluXMAJevXQF659d
E7BGWLR\nEcj\nnYk7xwj10CJfhkaR2JBQGF5E5z904uFL03gHwF2qIG6kJW2U
JKsUa0SkEdiU0oQfd\nnAxwoW/w=\n-----END CERTIFICATE REQUEST--
---\n'
<cryptography.hazmat.backends.openssl.rsa._RSAPublicKey obj
ect at 0x102108b80>
toobakhan@Toobas-MacBook-Air TLSA-3 %
```

Figure 3 TTP side(my\_ttp.py)

**To run the code, following steps have to be followed:**

1. Run python3 my\_ttp.py
2. Run python3 my\_server.py
3. Run python3 my\_client.py

**Security and efficiency of the protocol:**

1. TLS provides message authentication which is provided by the use of certificates and a trusted third party in my implementation.
2. It is safe against replay attacks because nonces are generated by both the client and server and they are used in the generation of session keys.
3. TLS provides encryption by using symmetric key to encrypt the data being exchanged. This key is generated by sharing a premaster key encrypted using public keys of client and server.
4. TLS works with most of the web browsers.
5. TLS is invisible to the client because it is implemented below the application layer.
6. It can also work with servers running on Windows 2003.
7. TLS also provides a wide range of encryption methods making it flexible enough for different use cases.