Assignment-3
Network and Security Systems

mcs202456

April 2021

1 Functionality

Below functionalities are included:

1.1 Trusted Third Party

Steps implemented in code

1. Receiving request from myserver along wih public key of myserver.
2. Generates openssl X509 digital certificate for myserver.
3. Sends generated certificate to myserver.
4. On receiving request from myclient along with public key of myclient.
5. Generates openssl certificate for myclient. Sends generated certificate to myclient.

1.2 MyServer

Handshake Protocol
Phase 1 -

1. Request TTP for certificate.
2. Waits for client to start handshake.
3. Client hello received with Info (version,random,sessionId,cipherSuite,keyexchange,compressMethod)
4. Server chooses verion(min of supported by client and server, cipherSuite(As RSA AES) ,compressMethod(zip)).
5. Server sends server hello with selection parameters.

Phase 2 :

1. Server sends its certificate to client.
2. Server sends keyexchange and shares its temporary generated RSA public key to be utilised for encrypting secret key.
3. Server sends Hello done.

Phase 3 :

1. Server receives certificate from client.
2. Server receives encrypted secret key which it decrypts with its private key.
3. Server receives certificate verify message.

Phase 4 :

1. Server receives Finish message from client.

Record Protocol :

1. Message to be transmitted is converted to fragments.
2. Each flagment is compressed with algo(zip) decided in handshake protocol.
3. Each fragment is MAC is calculated by algo(sha256) decided in handshake.
4. Message is appended with MAC and sequence number and encrypt it with shared secret key from client during client keyexchange in handshake.
5. ssl header is added at start and fragment is send to client.

1.3 MyClient

Handshake Protocol
Phase 1 -

1. Request TTP for certificate.
2. Receives certificate from TTP
3. client start handshakeby sending Info (version,random,sessionId,cipherSuite,keyexchange,compressMetho)
4. client receives selver hello info.

Phase 2 :

1. client receiuves certificate from server.
2. client receives server public key
3. Server receives Hello done.

Phase 3 :

1. Client sends its certificate to server.
2. Client sends secret key encrypted with server public key shared via server keyexchange message.
3. client verifies server certificate.

Phase 4 :

1. client sends Finish message to server.

Record Protocol :

1. Receives packets from server.
2. Check and remove ssl header.
3. Decrypt using secret key and decided algo(AES) in handshake.
4. Seperate MAC and create new mac from message with algo decided in handshake(sha256)
5. Compare new MAC with received mac from server to verify Integrity of message.
6. Receive all packets and do same for all.
7. construct message from all packets by identifying order with sequence number.

2 Security

The protocol is secure against following attacks:

1. replay attack: client,server hello message in handshake includes timestamps. Also, digital certificates have a validity period andfor every new communication new secret key is shared during handshake. So, protocol ensures that replay attackcould not occur.
2. Eavesdropping: The messages are send in encrypted manner. Also, the secret key shared in handshake is encrypted by server public key. So, eavesdropping is not possible.
3. Fabrication: Messages include MAC(message digest). which are verified at client to check the integrity of message. So, messages cannot be modified.
4. man-in-the-middle: The protocol authenticates sender with digital certificate from third party and verifies So, man in middle attack is not possible.

3 Computation cost

The computation cost includes:

1. Encryption/Decryption : RSA encryption of 16 bit secret key and AES encryption of message to be sent (18 fragments of 4 byte each).
2. Calculating and verifying MAC : hashing Cost(sha256) depends on length of messages used. hashing cost is proportional to (18 fragments of 4 byte each).
3. compression algorithm: compression cost is also proportional to message length (18 fragments of 4 byte each).

4 communication cost

The communication cost includes messages transmitted during handshake:

- Total messages transmitted from TTP to server or client are 4.This is one time cost till the certificate is valid. This handshaking is not required unti certificate expires.
- Total handshake messages transmitted between server and client are 12.
- Total message transmitted = 12*4 = 60
- Each packet transmitted has SIZE = 1024
- total transmitted bytes = 49152

5 Environment

5.1 Utilized Libraries

- RSA
- Pycrypto
- Pyopenssl
- hashlib

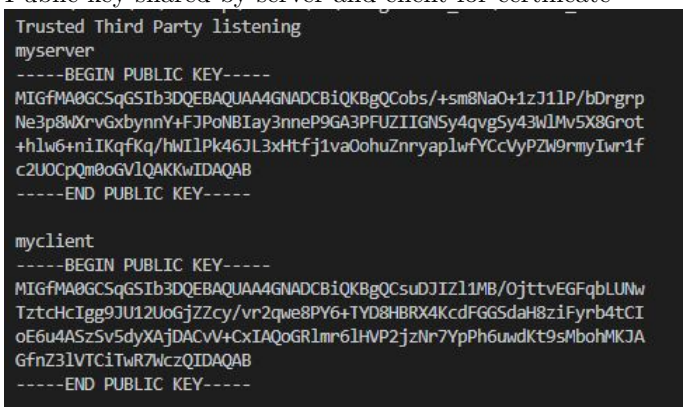
6 How to Run

- Ensure that required libraries are installed
- Run on one terminal : "python my$_t$*tp.py*" *Runonanotherterminal* : "*pythonmy$_s$erver.py*"

- Run on third terminal : "python my$_c$*lient.py*"

7 Screenshots

Public key shared by server and client for certificate



```
Trusted Third Party listening
myserver
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCobs/+sm8NaO+1zJ1lP/bDrgrp
Ne3p8WXrvGxbynnY+FJPoNBIay3nneP9GA3PFUZIIGNSy4qvgSy43WlMv5X8Grot
+hlw6+niIKqfKq/hWIlPk46JL3xHtfj1vaOohuZnryaplwfYCcVyPZW9rmyIwr1f
c2UOCpQm0oGVlQAKKwIDAQAB
-----END PUBLIC KEY-----

myclient
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCsuDJIZl1MB/OjttvEGFqbLUNw
TztcHcIgg9JU12UoGjZZcy/vr2qwe8PY6+TYD8HBRX4KcdFGGSdaH8ziFyrb4tCI
oE6u4ASzSv5dyXAjDACvV+CxIAQoGRlmr6lHVP2jzNr7YpPh6uwdKt9sMbohMKJA
GfnZ3lVTCiTwR7WczQIDAQAB
-----END PUBLIC KEY-----
```

client certificate.

```
Client certificate from TTP:
Certificate:
    Data:
        Version: 2 (0x1)
        Serial Number: 1 (0x1)
        Signature Algorithm: dsa_with_SHA256
        Issuer: CN=myttp
        Validity
            Not Before: Apr 15 07:33:54 2021 GMT
            Not After : Apr 15 07:33:54 2022 GMT
        Subject: CN=myclient
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (1024 bit)
                Modulus:
                    00:ea:55:af:df:32:59:2d:22:c8:45:0f:11:36:4b:
                    25:b8:27:9c:c8:8f:1b:03:3a:74:45:74:b1:b3:90:
                    bc:d7:ca:1b:8d:7a:e5:23:55:f3:61:63:94:de:6e:
                    c3:2d:00:2b:75:25:92:87:c2:fa:4e:37:dc:a6:6d:
                    e7:2b:0e:67:44:1f:b9:a2:b6:93:6e:d3:aa:c1:02:
                    4b:08:cf:03:72:34:2c:db:14:80:ad:c5:58:75:d4:
                    81:02:2e:9c:b3:2d:7e:91:54:2a:6e:dd:59:4b:50:
                    42:05:21:dc:50:a4:ec:c2:66:58:dc:2c:48:33:1e:
                    3c:44:0b:56:a1:90:9f:97:c3
```

Server certificate.

```
Certificate:
    Data:
        Version: 2 (0x1)
        Serial Number: 1 (0x1)
        Signature Algorithm: dsa_with_SHA256
        Issuer: CN=myttp
        Validity
            Not Before: Apr 15 07:25:04 2021 GMT
            Not After : Apr 15 07:25:04 2022 GMT
        Subject: CN=myserver
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (1024 bit)
                Modulus:
                    00:c7:46:3d:ed:88:64:66:c6:34:f
1:c0:a3:25:21:
                    79:c8:a6:d5:a5:1b:8f:49:9e:73:5
d:ef:74:46:4b:
                    24:2c:dd:e2:64:22:07:33:f4:f9:86:5d:92:f5:e2:
                    c5:67:a3:77:4a:5d:c8:d8:6c:a8:f0:26:0e:6b:0c:
                    9a:2a:30:22:2e:6e:1b:dc:7d:44:9b:b5:52:bd:35:
                    16:b5:53:84:76:b4:29:d8:02:d2:02:77:37:c6:c1:
                    94:78:ae:08:12:47:3d:fd:b9:8d:e7:41:73:8d:b7:
                    0e:31:f0:15:e6:f9:9c:f6:6d:74:c1:bb:6e:0b:12:
                    17:f3:10:e9:5d:85:f9:4d:0d
```

Server handshake and message sending.

```
TLS Handshake begins at client
Server Hello Message

{'version': '2.0', 'random': [200753887425221463427323765322
825566416537932657027534065296291 60], '1618484096.988847'], 'se
ssionId': '2', 'cipherSuite': {'keyexchange': 'RSA_AES', 'algo': 'SHA256'}, 'compressio
Server certificate received
Successfully received hello Done message
send client certificate
Successfully send certificate to server
Client Key exchange for exchanging secret key
b'WhkisvA3zL597GsJN-w4eDdCFygsu15eFu3_D2gFFEc='
PublicKey(840181212172416455761361987320084605521470865914279434784777910569941039133 31
47, 65537)
b'u/r\x91\x0e\xf6?\xe4\x8b\x90\xae\x9f1\xd7a\x8c"\xcd\xf8\xc1\x82\x17\xd2\x82\xc5\xf0\x
xa2E\xfd\xb7\xd2!\xd1\x17\xc2\x12y'
TLS handshaking success
TLS Record begins at client
The OTP for transferring Rs 100000 to your friend account is 256345
```

Server handshake and message sending.

```
Server starts
Get certificate from TTP
Server certificate received From TTP:

Certificate:
    Data:
        Version: 2 (0x1)
        Serial Number: 1 (0x1)
        Signature Algorithm: dsa_with_SHA256
        Issuer: CN=myttp
        Validity
            Not Before: Apr 15 10:54:50 2021 GMT
            Not After : Apr 15 10:54:50 2022 GMT
        Subject: CN=myserver
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (1024 bit)
                Modulus:
                    00:9a:98:79:b6:c8:4c:d8:63:15:3a:e4:b0:58:6
3:
                    c1:e6:e8:3c:0d:93:dd:53:09:b4:fd:ce:a1:10:2
e:
                    45:06:57:c9:2a:b5:55:57:2e:e2:76:3c:98:e9:4
c:
                    75:65:91:9c:f7:30:59:cf:62:b6:a4:7f:12:ca:7
d:
                    2a:78:c0:f4:05:11:cc:64:51:1b:14:3a:57:5c:5
6:
                    2a:d7:35:49:07:69:4c:b2:6c:8c:d2:e1:90:04:1
6:
                    94:7d:c2:2e:9c:2d:34:9c:60:1b:ca:43:67:58:3
b:
                    e6:5d:99:19:e4:e5:fd:61:02:ae:bc:8c:b8:47:0
2:
8484096.987848'], 'sessionId': '0', 'cipherSuite': {'keyexchange': ['RSA_AES', 'ECDSA_
pressionMethod': ['zip', 'tar']}
Server hello End
Server certificate send
server public key send for key exchange
Start Hello Done
Client certificate received
Secret key received from client
TLS handshaking success
TLS Record begins at server
```