CS 444 Programming 3
Xiaoxiao Liu (xliu91)

Part 1 & Part2:

Server:

Command: $ python server.py *host port file*

ex run:
with no option:
python server.py 127.0.0.1 8801 index.html

with cert path:
python server.py --cacert ./ssl/certificate.pem 127.0.0.1 8801 index.html

with version:
python server.py --sslv23 --cacert ./ssl/certificate.pem 127.0.0.1 8801 index.html

with cipher:
python server.py --sslv23 --cacert ./ssl/certificate.pem --cipher ECDHE-RSA-AES256-GCM-SHA384 127.0.0.1 8801 index.html

Client:

Note: For the purpose of this homework, to test part1&2, on the client-side should not specify certificate file path (but DOES NOT mean this parameter won't work, its effectiveness can be tested in part3), otherwise will change to the result of part3.

Command: $ python client.py <*ssl/tsl version*> <*ciphers*> *host port file*

ex run:
no option:
python client.py 127.0.0.1 8801 index.html

with version:
python client.py --tlsv1.1 127.0.0.1 8801 index.html

with cipher:
python client.py --ciphers ECDHE-RSA-AES256-GCM-SHA384 127.0.0.1 8801 index.html

Part3:

Server: *(no change)*
Client:

NOTE: If client specifies certificate, then switch to this part, client-side print will be ONLY the certificate got from client

Command: $ python client.py --cacert *path* <*ssl/tsl version*> <*ciphers*> *host port file*

ex run:
with only cert:
python client.py --cacert ./ssl/certificate.pem 127.0.0.1 8801

with version:
python client.py --tlsv1.2 --cacert ./ssl/certificate.pem 127.0.0.1 8801

with ciphters:
python client.py --tlsv1.2 --ciphers ECDHE-RSA-AES256-GCM-SHA384 --cacert ./ssl/certificate.pem 127.0.0.1 8801

Other Note:

1. To test that user input option params are correctly accepted and used, change the global variable "option_test_switch" value to 1 (can do on both client and server).
2. All ssl files are stored in ./ssl/ directory