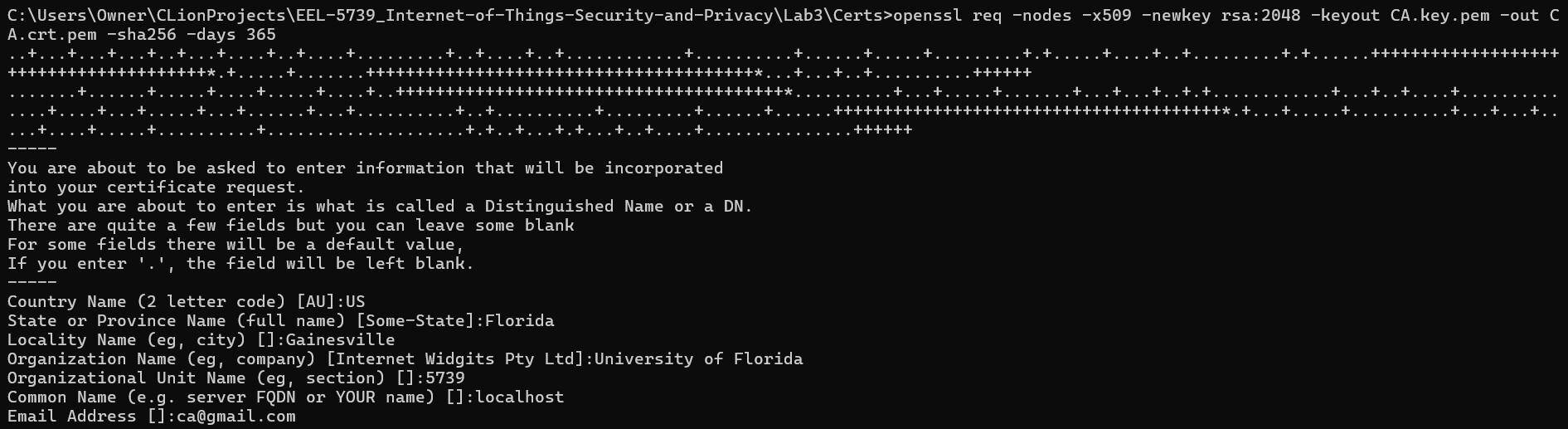
# Create Self-Signed Certificate to act as Certificate Authority:

## Create Certificate Authority

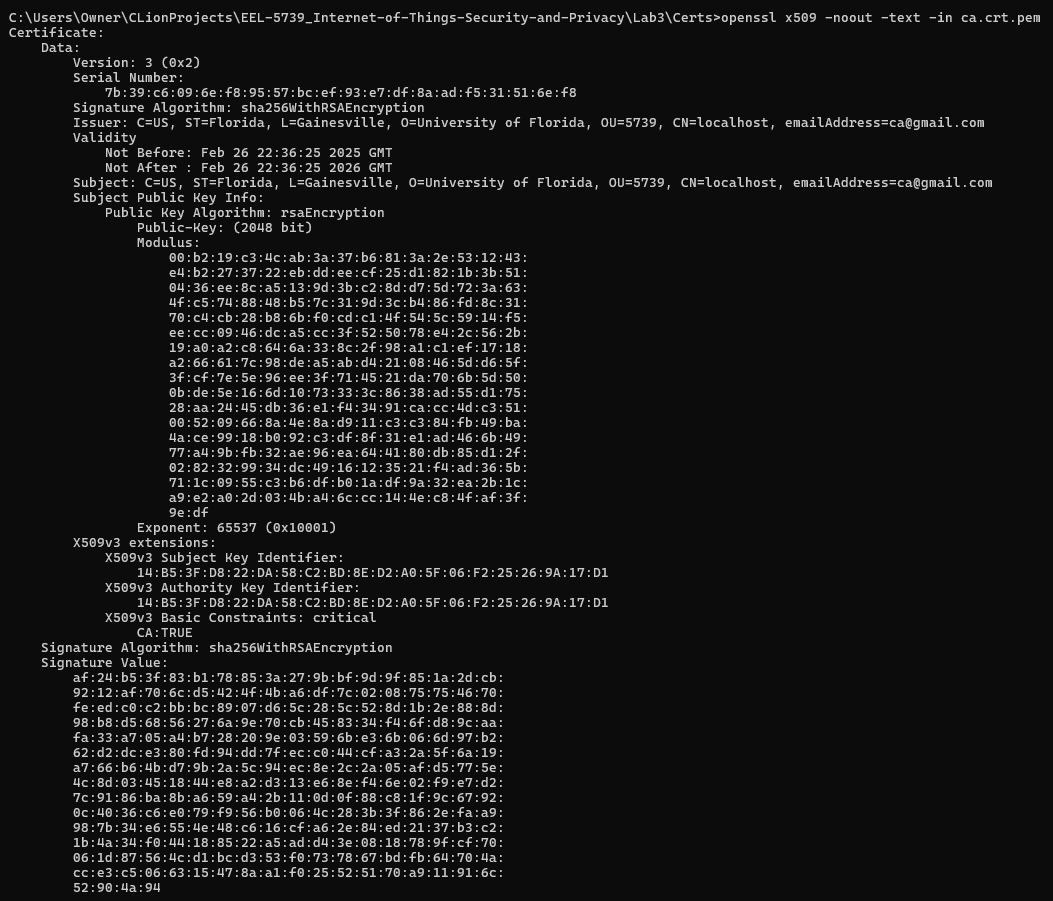
openssl req -nodes -x509 -newkey rsa:2048 -keyout CA.key.pem -out CA.crt.pem -sha256 -days 365

Note: The Common Name (CN) must match your computer’s IP address. I recommend using localhost for Lab 3



## Verify Certificate Authority:

openssl x509 -noout -text -in ca.crt.pem



## Verify Certificate matches Key:

openssl rsa -noout -modulus -in ca.key.pem | openssl sha256

openssl x509 -noout -modulus -in ca.crt.pem | openssl sha256



# Create and Sign a Certificate:

## Create a Private RSA Key:

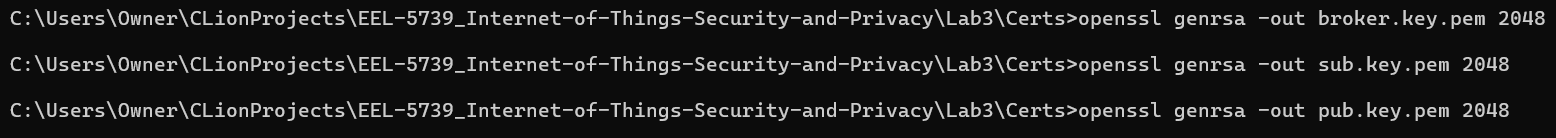
openssl genrsa -out client.key.pem 2048



openssl genrsa -out broker.key.pem 2048

openssl genrsa -out sub.key.pem 2048

openssl genrsa -out pub.key.pem 2048

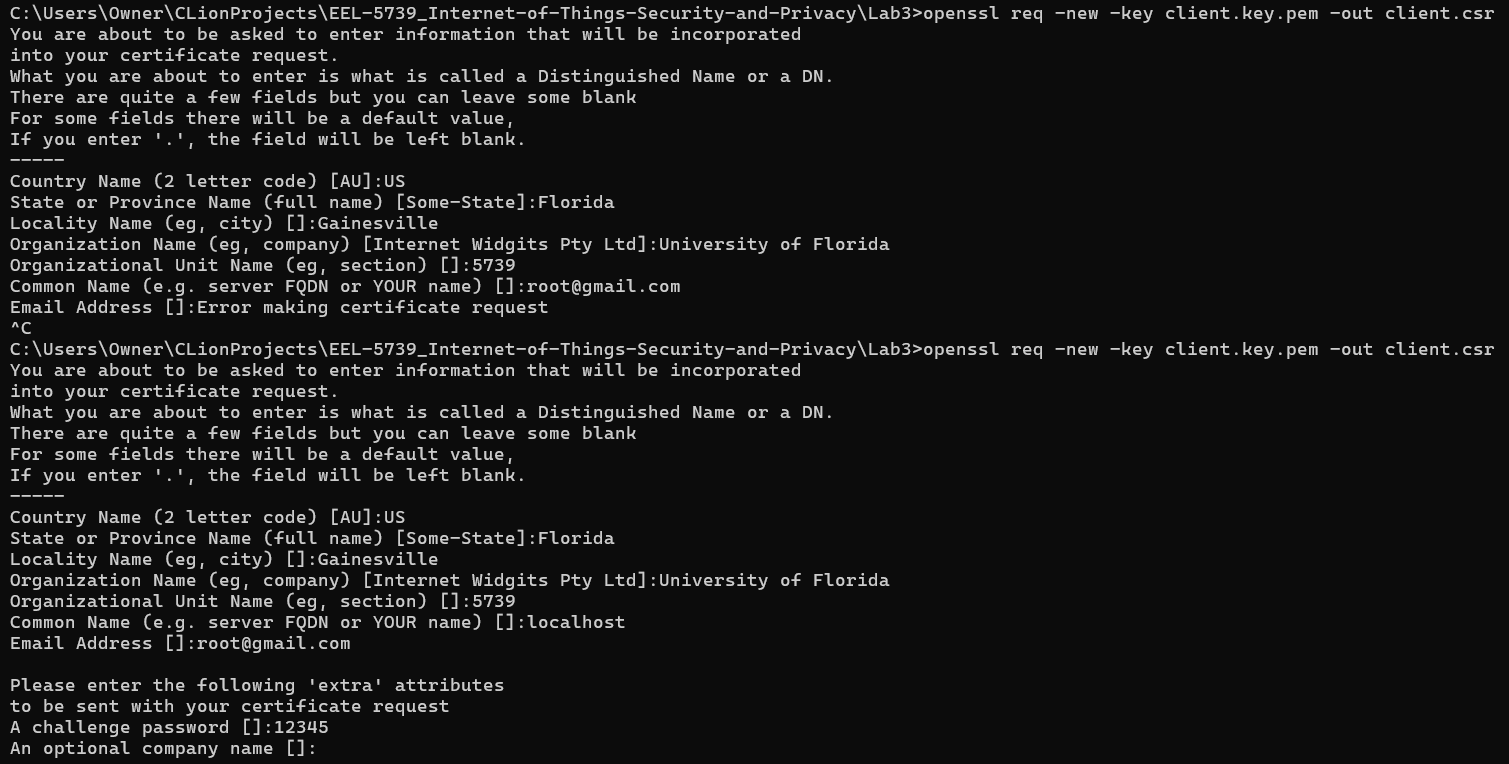


## Create Certificate Signing Request:

openssl req -new -key client.key.pem -out client.csr

Note: each certificate needs some field filled differently in the CSR (I recommend different emails, e.g., root@ca, broker@ca, etc.)

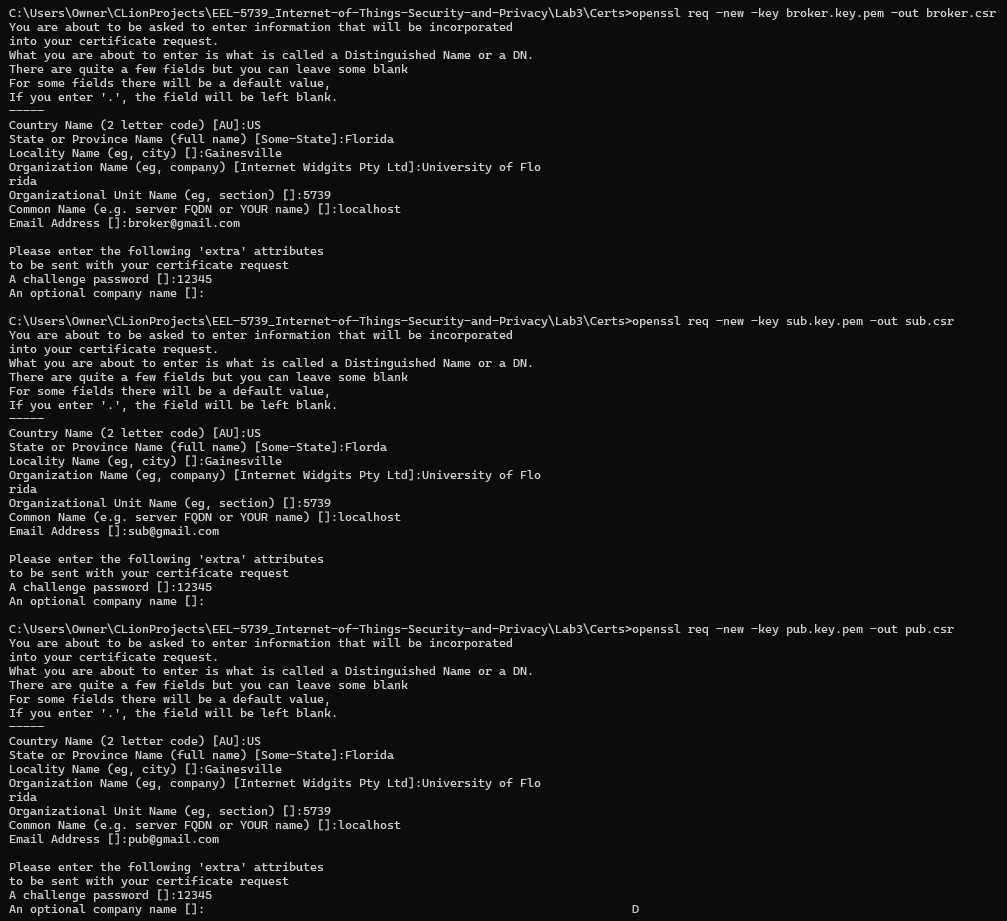
Note: The Common Name (CN) must match your computer’s IP address. I recommend using localhost for Lab 3



openssl req -new -key broker.key.pem -out broker.csr

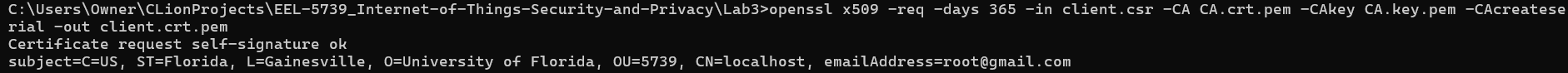
openssl req -new -key sub.key.pem -out sub.csr

openssl req -new -key pub.key.pem -out pub.csr



## Create a Certificate using the CSR:

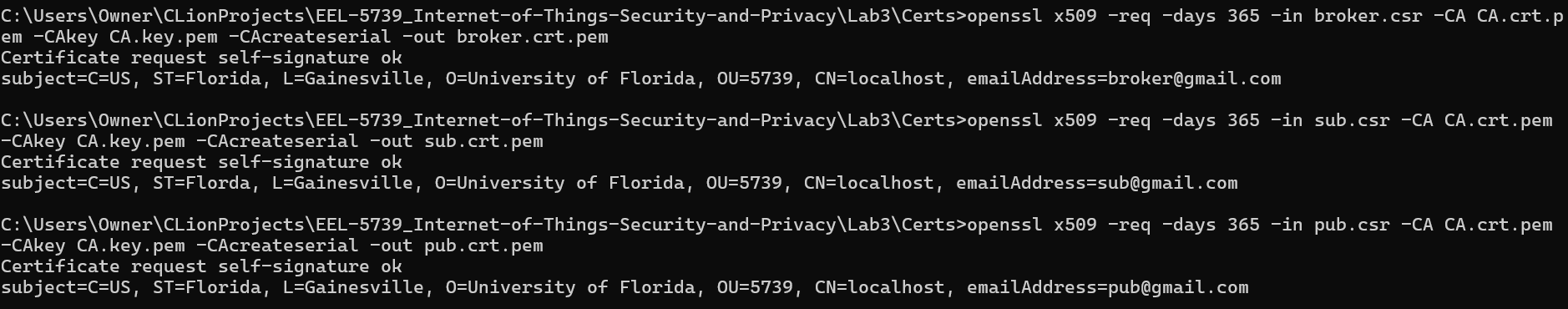
openssl x509 -req -days 365 -in client.csr -CA CA.crt.pem -CAkey CA.key.pem -CAcreateserial -out client.crt.pem



openssl x509 -req -days 365 -in broker.csr -CA CA.crt.pem -CAkey CA.key.pem -CAcreateserial -out broker.crt.pem

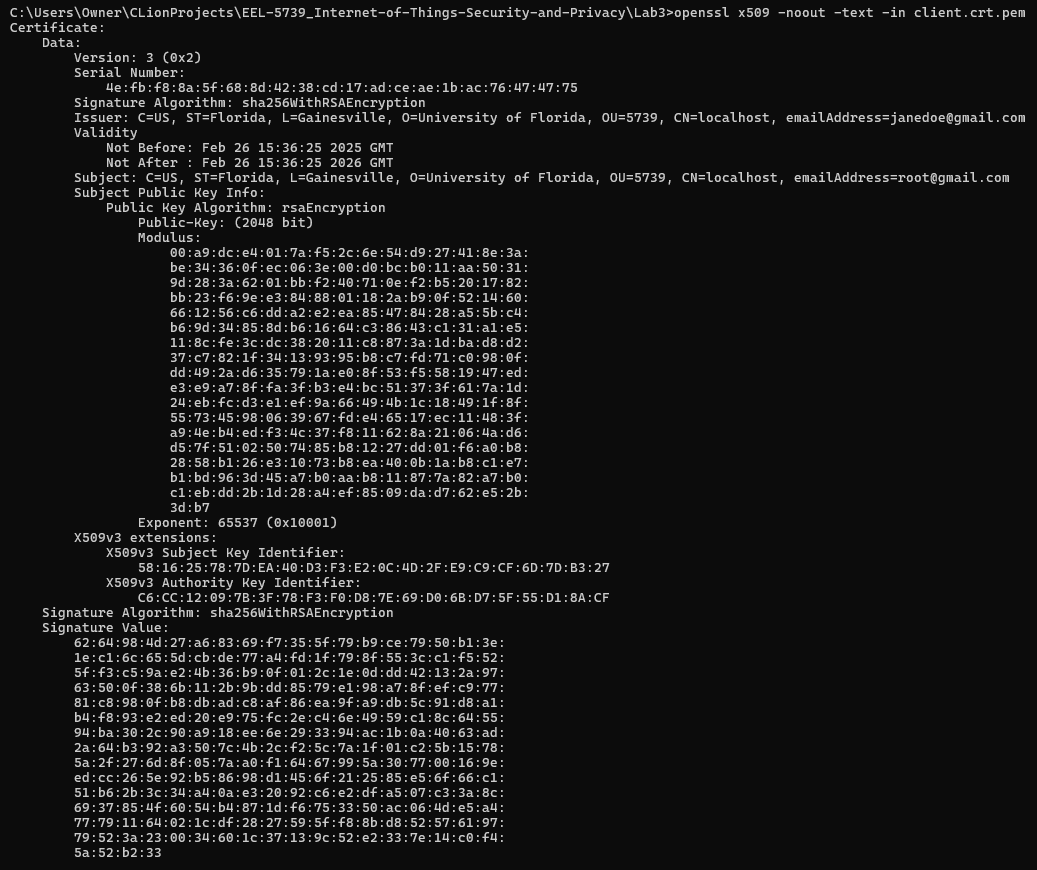
openssl x509 -req -days 365 -in sub.csr -CA CA.crt.pem -CAkey CA.key.pem -CAcreateserial -out sub.crt.pem

openssl x509 -req -days 365 -in pub.csr -CA CA.crt.pem -CAkey CA.key.pem -CAcreateserial -out pub.crt.pem

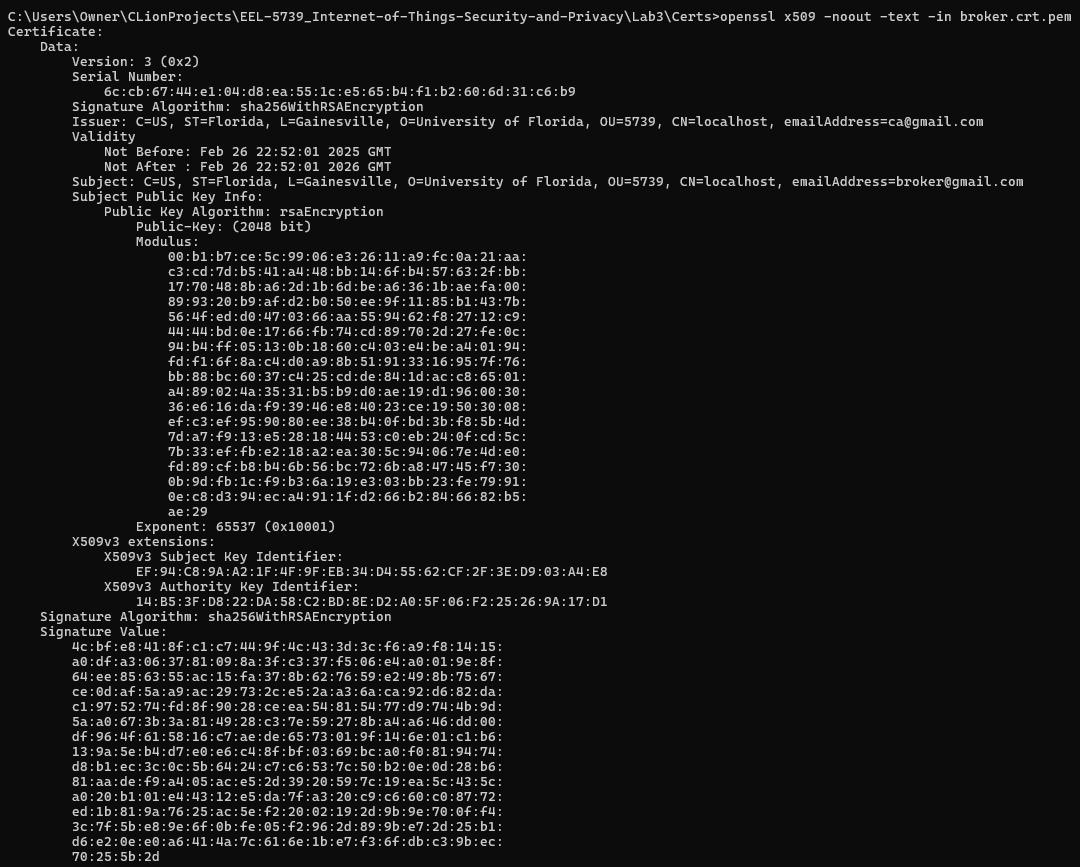


## Verify Certificate:

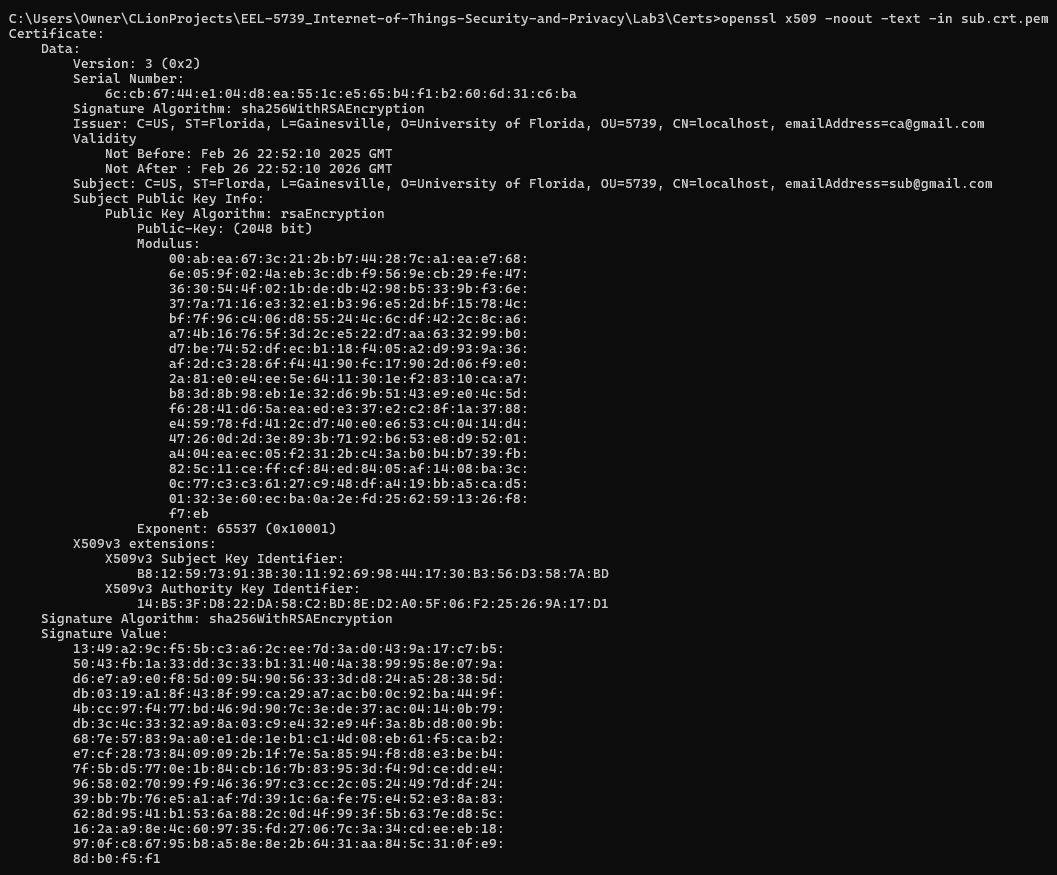
openssl x509 -noout -text -in client.crt.pem



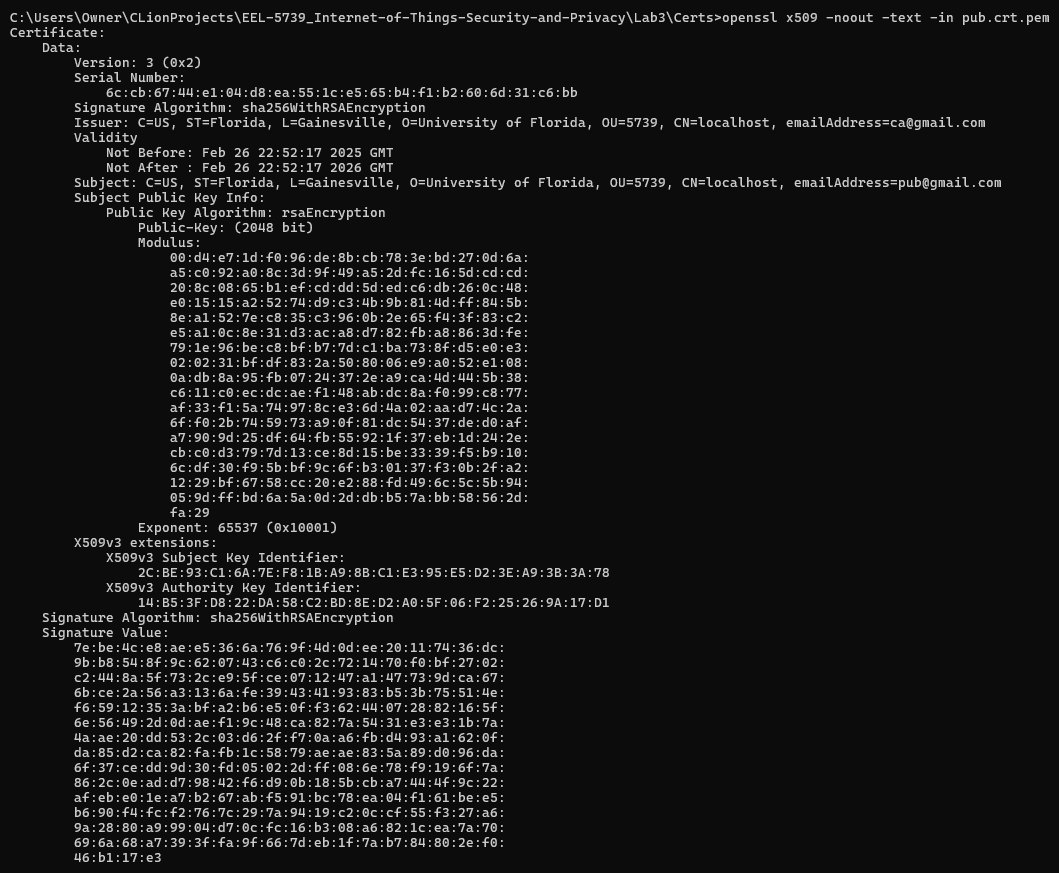
openssl x509 -noout -text -in broker.crt.pem



openssl x509 -noout -text -in sub.crt.pem



openssl x509 -noout -text -in pub.crt.pem



## Verify Certificate matches Key:

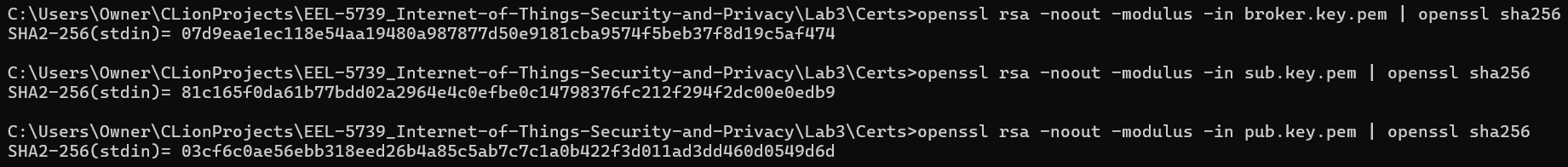
openssl rsa -noout -modulus -in client.key.pem | openssl sha256



openssl rsa -noout -modulus -in broker.key.pem | openssl sha256

openssl rsa -noout -modulus -in sub.key.pem | openssl sha256

openssl rsa -noout -modulus -in pub.key.pem | openssl sha256



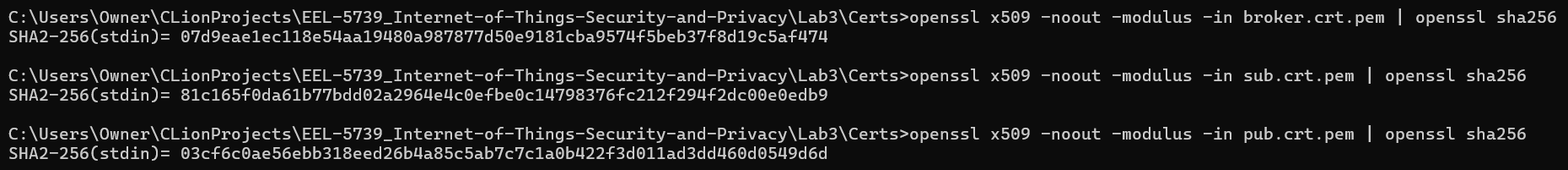
openssl x509 -noout -modulus -in client.crt.pem | openssl sha256



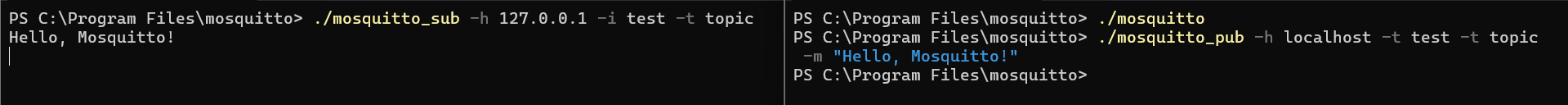
openssl x509 -noout -modulus -in broker.crt.pem | openssl sha256

openssl x509 -noout -modulus -in sub.crt.pem | openssl sha256

openssl x509 -noout -modulus -in pub.crt.pem | openssl sha256



Mosquitto Checking



Mosquito Broker with SSL/TLS

Generate Server Certificates (CA.crt and CA.key)

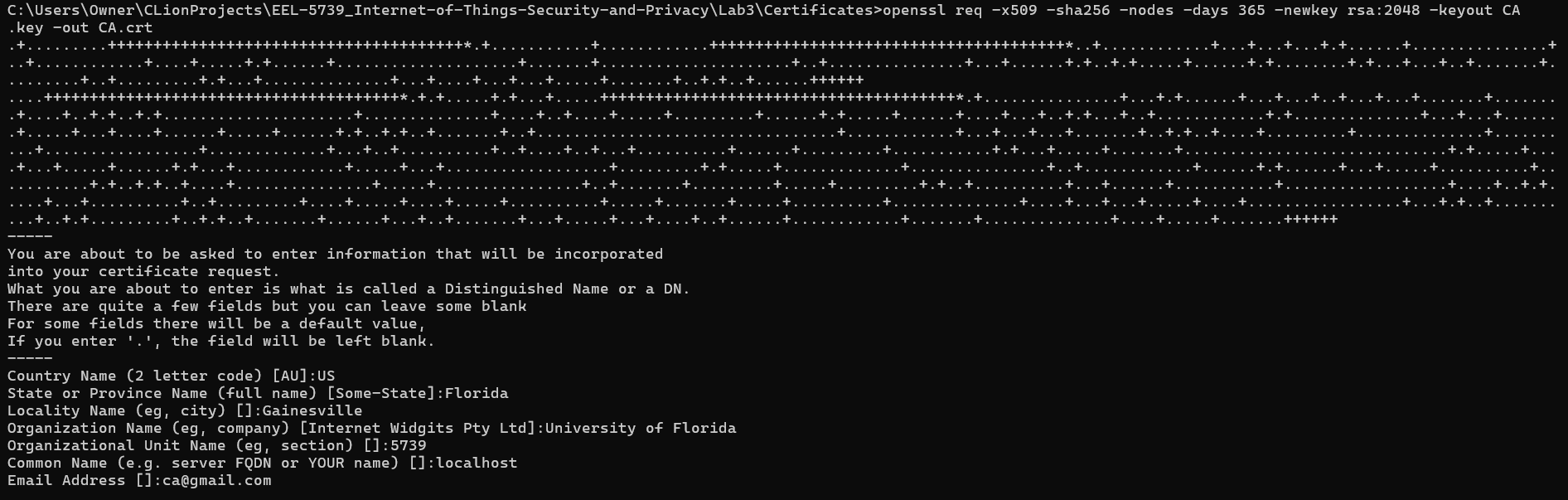
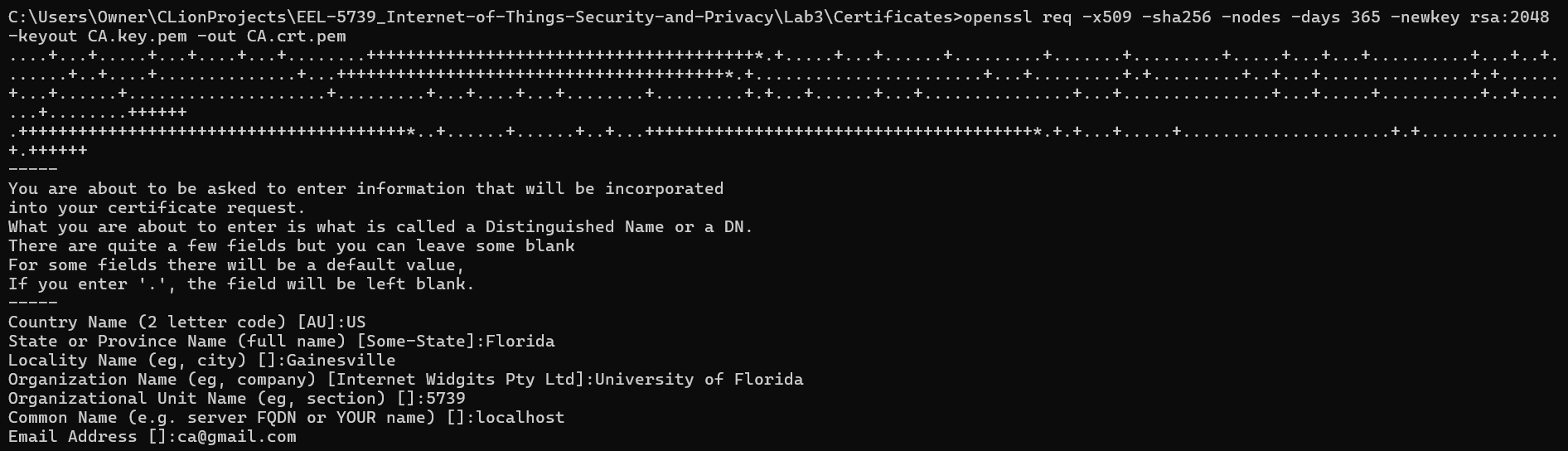


Figure : Needs to be in PEM format

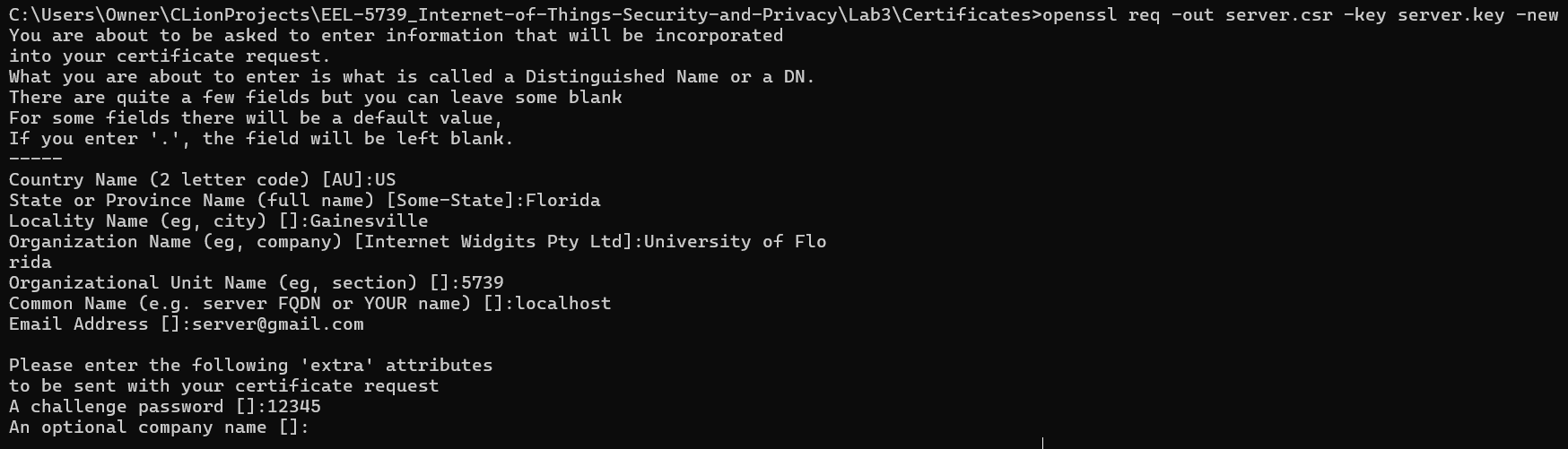


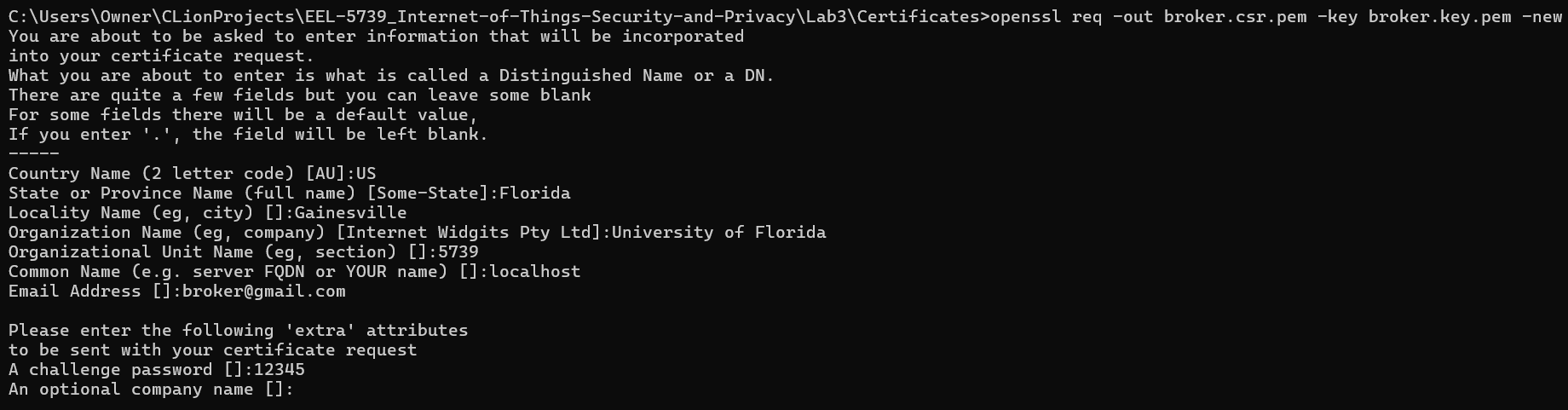
Generate a key for server (server.key)



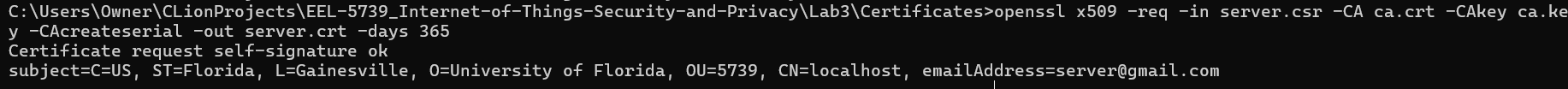


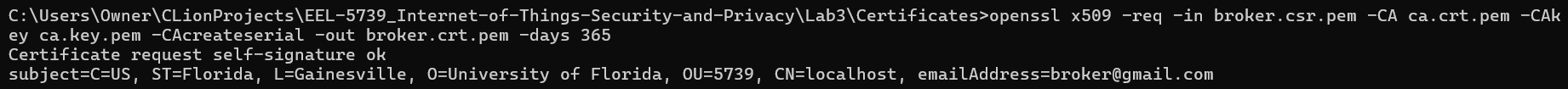
Generate Certificate signing request(CSR)





Send it to CA





openssl genrsa -out broker.key.pem 2048

openssl req -out broker.csr.pem -key broker.key.pem -new

openssl x509 -req -in broker.csr.pem -CA ca.crt.pem -CAkey ca.key.pem -CAcreateserial -out broker.crt.pem -days 365

Run Each of the following in separate terminals:

./mosquitto -c "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab3\mosquitto.conf" -v

./mosquitto\_sub -h 127.0.0.1 -i "testSub" -t "test" -p 8883 --cafile "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab3\Certs\CA.crt.pem" --cert "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab3\Certs\sub.crt.pem" --key "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab3\Certs\sub.key.pem" -d

./mosquitto\_pub -h 127.0.0.1 -p 8883 -i "testPub" -t "test" -m "Hello" --cafile "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab3\Certs\CA.crt.pem" --cert "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab3\Certs\pub.crt.pem" --key "C:\Users\Owner\CLionProjects\EEL-5739\_Internet-of-Things-Security-and-Privacy\Lab3\Certs\pub.key.pem" -d