

# SIL765 – Assignment 3

## Certification Authority (CA)

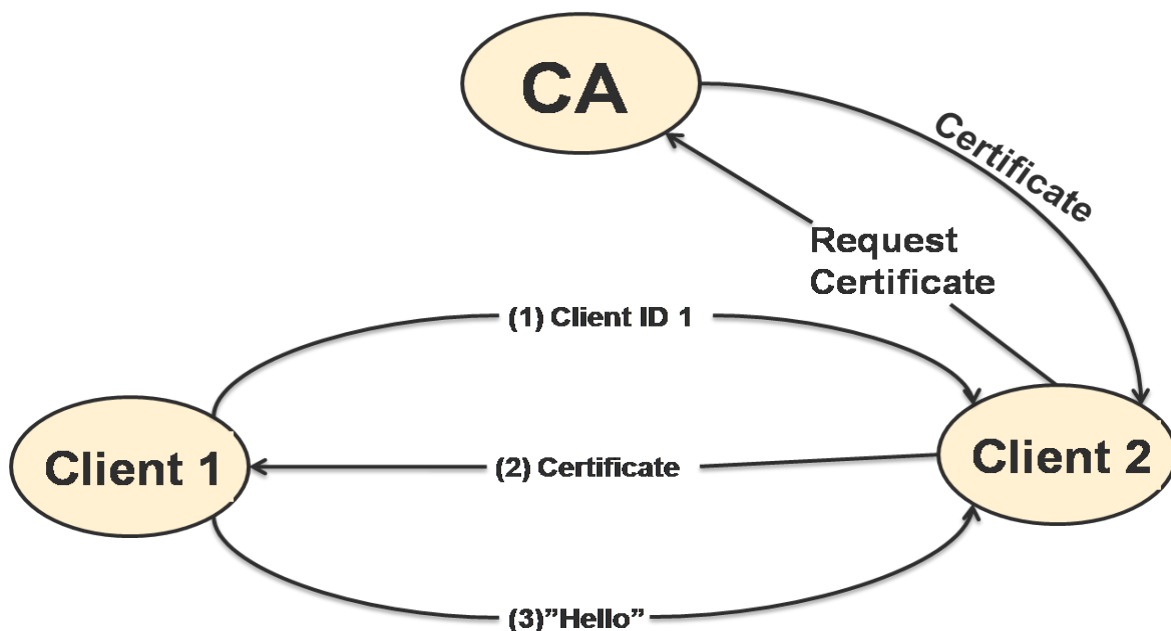
Kushagra Madan : 2014CS10232

Pragnesh Jogadiya : 2014CS10246

### Overview:

In this project, we have built a Certification Authority which has public keys of all the clients and corresponding to which the clients themselves have their private keys. Public key of Certification Authority is known to all the clients. Certification Authority will provide a certificate to clients on request.

2048 bit RSA algorithm has been used for secure communication between CA and client as well as between client and client.



## **Certificate Format:**

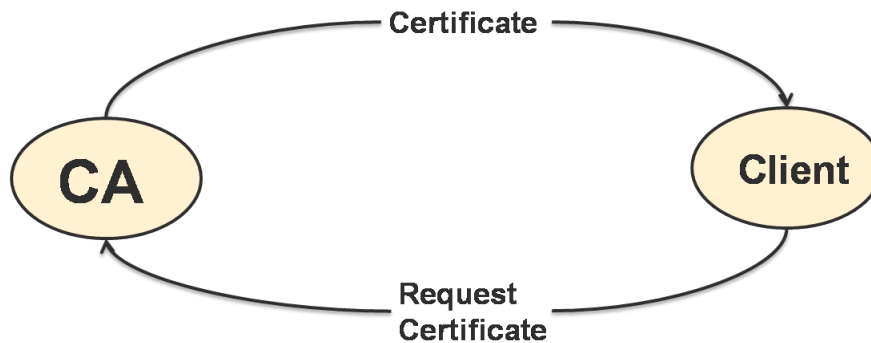
Format of certificate:  $CERT_A = ENC_{PRX}(ID_A, KU_A, T_A)$ , where

- $PR_X$  is private key of certification authority ( $PU_X$  is public key of certification authority)
- $ID_A$  is user ID,
- $KU_A$  is public key of A,
- $T_A$  is time of issuance of certificate.

Certification authority will hash client ID, public key of client and time of issuance of certificate. It will then sign the hash using its private key and it to the client along with the time of issuance of the certificate.

## **Certification Authority:**

Certification authority has public keys of all clients and its public key is known to all clients. On request from client, CA issues certificate encrypted with its private key to client which includes, client ID, public key of client and time of issuance of certificate.



## **Client:**

Client will have three options available.

- (1) Get certificate from certification authority.
- (2) Send hello to a particular client.
- (3) Receive message from client.

## 1. Getting certificate from certification authority.

On selecting the first option, client will request a public key certificate from CA. On receiving the request, CA will send client's certificate encrypted with CA's private key to the client. Client can decrypt it using the public key of CA.

## 2. Send hello to client

On selecting the second option, client will send its client ID to a receiving client. On receiving the client ID, receiver will send its certificate to sender. Sender will verify certificate and after verifying, it will send "hello" message encrypted with its own private key to the receiver.

```
pub_f = open('./server_key.pub', 'rb')
ca_pub_key = pub_f.read()
ca_pub_key_obj = RSA.importKey(ca_pub_key)
recv_hash = hashlib.sha512(recv_id + "_" + str(recv_pub_key) + "_" + recv_time).digest()
recv_sig = eval(recv_sig)
if(ca_pub_key_obj.verify(recv_hash, recv_sig)):
    print("Certificate verified!")
    s.close
    time.sleep(3)
    s = socket.socket()
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.connect((host, port - int(client_id) + int(recv_id)))
    msg = "hello from client " + client_id
    recv_pub_key_obj = RSA.importKey(recv_pub_key)
    emsg = recv_pub_key_obj.encrypt(msg, 'x')[0]
    s.sendall(emsg)
    s.close
    print("Sent message to client " + recv_id + "\n")
    time.sleep(1)
else:
    print("Invalid certificate!\n")
    s.close
    time.sleep(1)
```

Code Snippet in which client is sending "hello" to another client.

### 3. Receive message from client.

On selecting the third option, client will wait to receive message from sender client.

```
elif (int(inp) == 3):
    #receive hello
    s.close
    s = socket.socket()
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind((host, port))
    s.listen(5)
    c, addr = s.accept()
    recv_client_id = c.recv(1024)
    f = open("./client" + client_id + "/certificate" + client_id + ".txt" ,"rb")
    msg = f.read()
    c.send(msg)
    c.close()
    c, addr = s.accept()
    recv_emsg = c.recv(1024)
    f = open("./client" + client_id + "/key" + client_id + '.pem', 'rb')
    priv_key = f.read()
    priv_key_obj = RSA.importKey(priv_key)
    recv_msg = priv_key_obj.decrypt(recv_emsg)
    print("Message received from client " + recv_client_id + ": " + recv_msg + "\n")
    s.close
```

Code Snippet in which client is waiting for sender's message.

## Communication between clients:

For communication between clients, following steps are taken.

- (1) Receiving client will wait for the sender's client ID.
- (2) On receiving the ID, receiver will send its certificate to sender.
- (3) Sender will verify certificate sent by receiver.
- (4) After verification of certificate sent by receiver, sender will send "hello" to receiver.

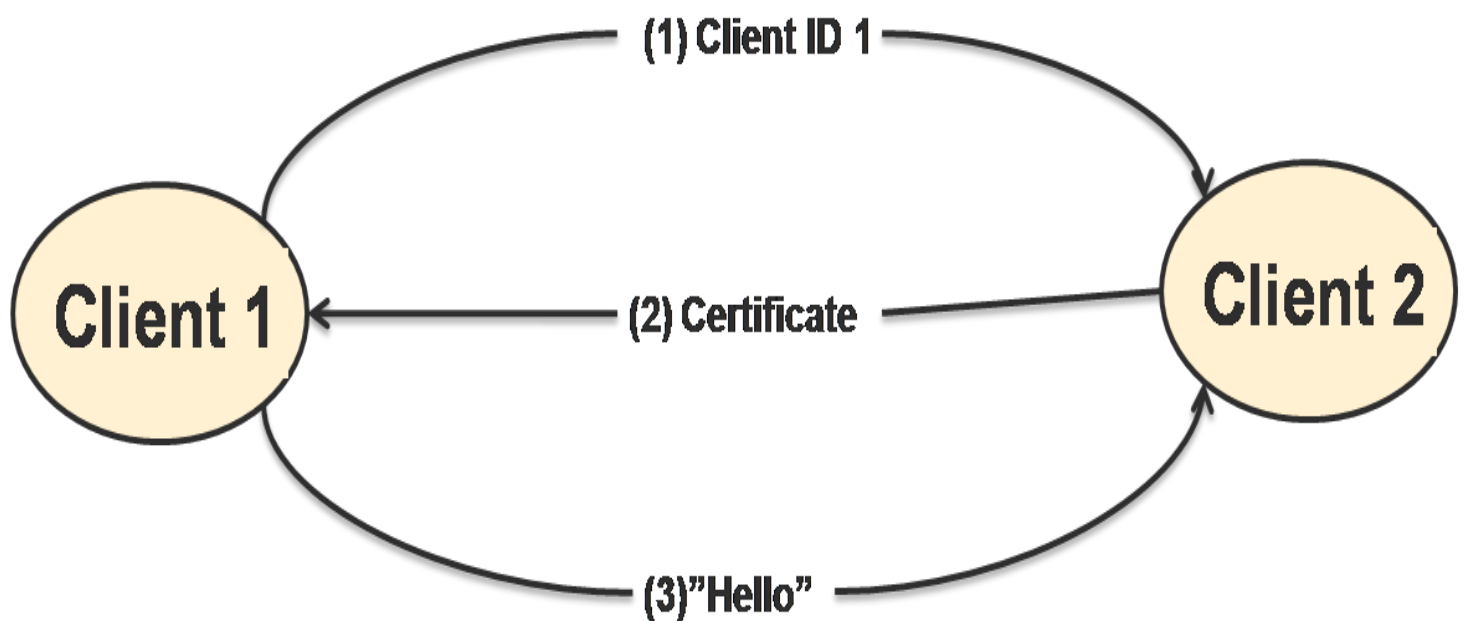
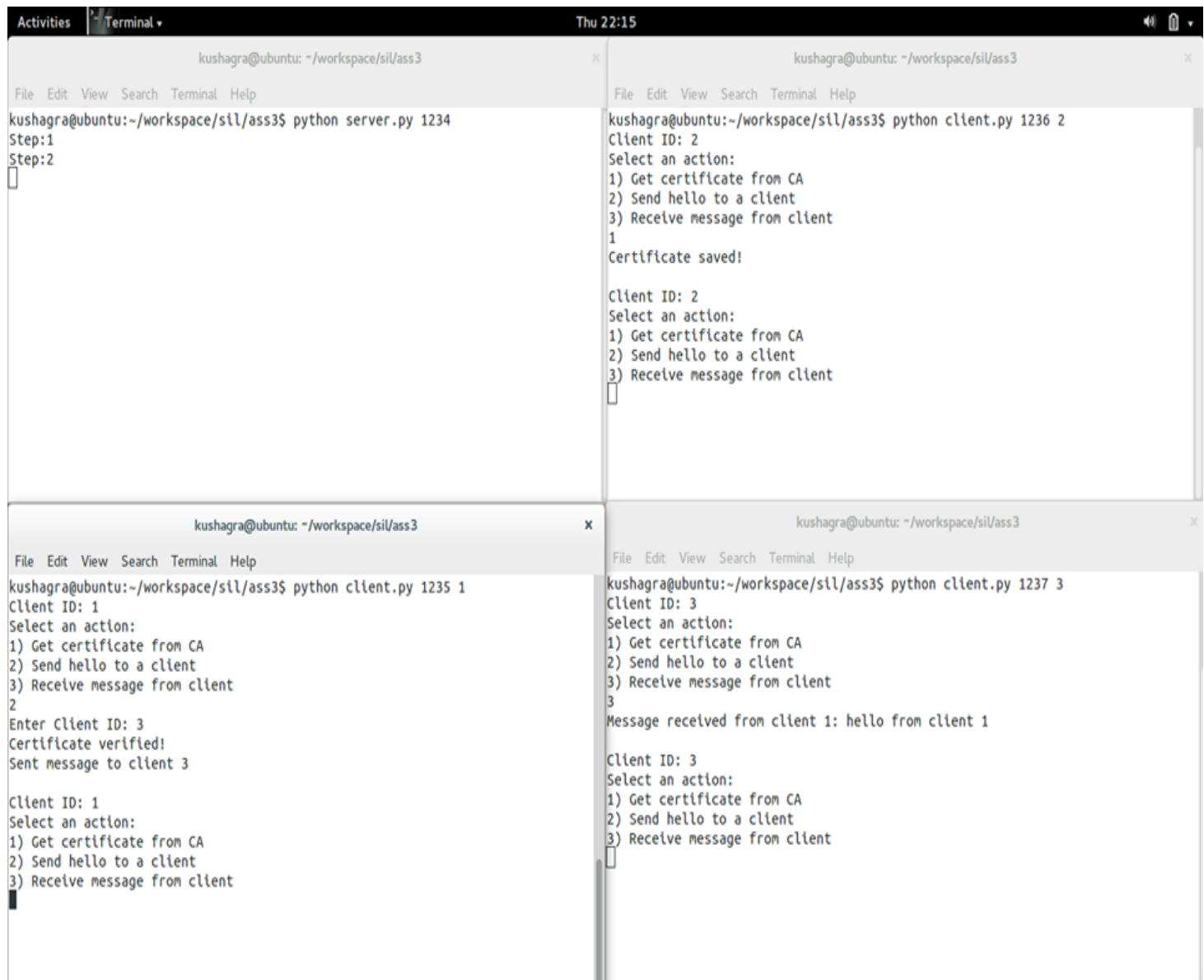


Diagram showing communication between clients.

## Sample Run:



The screenshot shows a terminal window with four panes, each displaying the output of a Python program. The top-left pane shows the server.py program running with client ID 1234. The top-right pane shows the client.py program running with client ID 1236. The bottom-left pane shows the client.py program running with client ID 1235. The bottom-right pane shows the client.py program running with client ID 1237. The program prompts the user to select an action from a list of three options: 1) Get certificate from CA, 2) Send hello to a client, and 3) Receive message from client. The user enters '1' in the top-left pane, '2' in the top-right pane, '2' in the bottom-left pane, and '3' in the bottom-right pane. The program outputs the corresponding action and the result of the action.

```
kushagra@ubuntu: ~/workspace/sil/ass3
File Edit View Search Terminal Help
kushagra@ubuntu:~/workspace/sil/ass3$ python server.py 1234
Step:1
Step:2
█
```

```
kushagra@ubuntu:~/workspace/sil/ass3$ python client.py 1236 2
Client ID: 2
Select an action:
1) Get certificate from CA
2) Send hello to a client
3) Receive message from client
1
Certificate saved!

Client ID: 2
Select an action:
1) Get certificate from CA
2) Send hello to a client
3) Receive message from client
█
```

```
kushagra@ubuntu:~/workspace/sil/ass3$ python client.py 1235 1
Client ID: 1
Select an action:
1) Get certificate from CA
2) Send hello to a client
3) Receive message from client
2
Enter Client ID: 3
Certificate verified!
Sent message to client 3

Client ID: 1
Select an action:
1) Get certificate from CA
2) Send hello to a client
3) Receive message from client
█
```

```
kushagra@ubuntu:~/workspace/sil/ass3$ python client.py 1237 3
Client ID: 3
Select an action:
1) Get certificate from CA
2) Send hello to a client
3) Receive message from client
3
Message received from client 1: hello from client 1

Client ID: 3
Select an action:
1) Get certificate from CA
2) Send hello to a client
3) Receive message from client
█
```