

NSC Assignment 4

We had to implement Project 2 (i.e., On-the-go verification of Driver's License).

System Overview -

We had implemented a pan-India RTO verification model. It contains 2 files:

client.py: This will be used by the police Officer. The Police Officer will scan the Driving License and get the signed and encrypted DL information. He/she will then send this information to their own local RTO servers for which they will have the access, for verification.

server.py: This is the RTO server. There are multiple servers active at a given time for fault tolerance purposes. Each police officer will have access to their local server only to ensure privacy. Once the local server receives a verification request, there can be 2 cases:

- The local RTO itself issued the DL, and therefore will be able to directly verify.
- The license was signed and issued from a different RTO, so the local RTO might not have the access to the same. So it will contact the central RTO to get the public keys of the signing RTO. Once received, it will also store it with itself for next time verifications. Now, using the obtained public key, the verification can be done easily.
- **For verification:** The encrypted digital certificate received by the local RTO is first decrypted using its own hash function. Now the signature of the document ID is decoded using the public key of the signing authority (self or other RTO). So 2 step verification:
 - Hash Value Verification
 - Signature Verification using Public Key of Signing Authority.
- For signing: Using its own private key, the RTO will sign the DL

Questions -

Q1) What is the information to be supplied by the driver to the police officer? And what information is sought and obtained by the police officer from the transport authority?

A1) The driver needs to supply his driving license, which is digitally signed by any transport authority. The police officer will extract the encrypted and signed certificate along with a unique signature ID, from this QR data on license.

The police officer just needs the access to the Transport Authority servers, i.e. port and ip. He/She then sends the data from license QR over to the transport server. The transport server will internally verify the license with the signing authority, and return if it is valid or not.

Q2) Would you need a central server that has the correct and complete information on all drivers and the licenses issued to them?

A2) No, in case of servers of different cities being active and managing their data only. Each RTO has their own signing methods and therefore can separately verify them using public and private keys. For example: Kerala RTO will have the information of kerala licenses only. However, if we want to allow cross-modal information between servers, then we would need a centralized server. In this case, if a certain license is not obtained from that region's RTO then it will redirect it to the national RTO which will redirect it to the region that the particular license belongs to.

Q3) In what way are digital signatures relevant?

A3) Digital signatures help in authentication, integrity and non-repudiation of original information of license holders, by the traffic police. It helps in authenticating the license holder that his/her license is valid and its integrity is maintained. Additionally, it also helps in non-repudiation thereby making it an essential key of the verification process. Thereby meaning that if a digital signature is verified then no one can repudiate its integrity or its ownership.

Q4) Does one need to ensure that information is kept confidential? Or not altered during 2-way communication?

A4) No. Confidentiality is not required over here. However, the information can not be altered during the 2-way communication. For the same reason, RSA encryptions are being used.

Q5) Which of these, viz. confidentiality, authentication, integrity and non-repudiation is/are relevant?

A5) In the case of digital signatures authentication, integrity and non-repudiation are relevant. However, due to the insensitive nature of the information being shared, confidentiality is not relevant here.

Bonus Question -

Question) Is date and time of communication important? If so, how can that be obtained from a well-known server in a secure manner?

Answer) Yes. This is important as it will help in preventing replay attacks. The information is encoded using the well known RSA algorithm and then sent over a secure channel. Additionally a hash value is also sent of the original message so that the integrity of the received message can be verified. This is how we obtain information from a well-known server in a secure manner.

Once the message is received, the server/client will verify the time taken, and if it is within limits (say 1sec) then it will proceed accordingly, else drop the packet, and deny access.

The screenshot displays a Python application running in a terminal window. The terminal output shows the following sequence of events:

```

KERALA RTO is listening on port: 4000
Received message: {'req_type': 1, 'rto': 'KERALA RTO', 'encoded_doc': [3, 4, 11, 7, 8, 17, 19, 14, 3, 11, 0, 0, 12, 11, 4, 4, 13, 28, 30, 26, 33], 'signature': 81, 'signingAuthority': 'DELHI RTO', 'timestamp': [28, 26, 28, 30, 26, 30, 28, 27, 31, 27, 31, 33, 32, 35, 30, 35, 31, 35]} from ('127.0.0.1', 49944)
Requesting public key of DELHI RTO from INDIAN RTO
Request: {'req_type': 2, 'rto': 'KERALA RTO', 'requesting_rto': 'DELHI RTO', 'timestamp': [28, 26, 28, 30, 26, 30, 28, 27, 31, 27, 31, 33, 33, 26, 28, 34]} sent to 127.0.0.1:4000

```

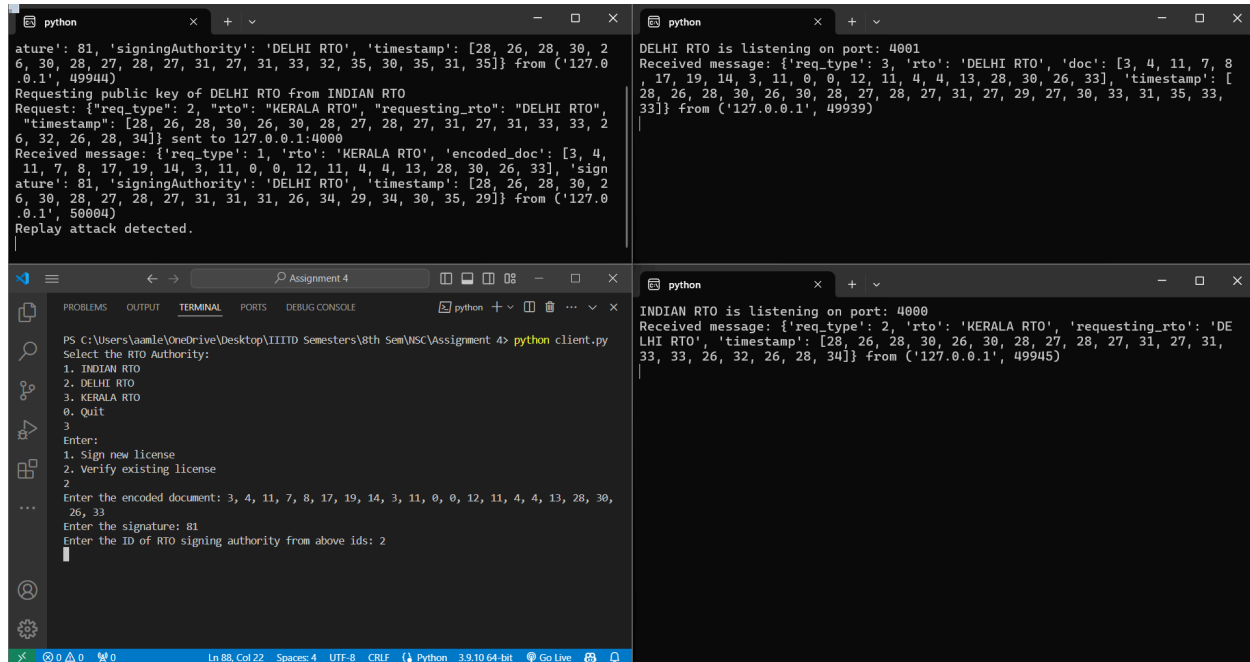
The GUI window, titled "Assignment 4", contains a menu with the following options:

- Signed document: [3, 4, 11, 7, 8, 17, 19, 14, 3, 11, 0, 0, 12, 11, 4, 4, 13, 28, 30, 26, 33]
- Signature: 81
- Signing Authority: DELHI RTO
- Select the RTO Authority:
 1. INDIAN RTO
 2. DELHI RTO
 3. KERALA RTO
 0. Quit
- Enter:
 1. Sign new license
 2. Verify existing license
- Enter the encoded document: 3, 4, 11, 7, 8, 17, 19, 14, 3, 11, 0, 0, 12, 11, 4, 4, 13, 28, 30, 26, 33
- Enter the signature: 81
- Enter the ID of RTO signing authority from above ids: 2
- Verification Status: True
- Select the RTO Authority:
 1. INDIAN RTO

The bottom status bar of the application shows the following information:

- Ln 50, Col 13
- Spaces: 4
- UTF-8
- CR/LF
- Python 3.9.10 64-bit
- Go Live
- ENG IN
- 09:52 PM
- 21-04-2024

Testing



```
python
ature': 81, 'signingAuthority': 'DELHI RTO', 'timestamp': [28, 26, 28, 30, 2
6, 30, 28, 27, 28, 27, 31, 27, 31, 33, 32, 35, 30, 35, 31, 35]} from ('127.0
.0.1', 49944)
Requesting public key of DELHI RTO from INDIAN RTO
Request: {'req_type': 2, 'rto': 'KERALA RTO', 'requesting_rto': 'DELHI RTO',
'timestamp': [28, 26, 28, 30, 26, 30, 28, 27, 28, 27, 31, 27, 31, 33, 33, 2
6, 32, 26, 28, 34]} sent to 127.0.0.1:4000
Received message: {'req_type': 1, 'rto': 'KERALA RTO', 'encoded_doc': [3, 4,
11, 7, 8, 17, 19, 14, 3, 11, 0, 0, 12, 11, 4, 4, 13, 28, 30, 26, 33], 'sign
ature': 81, 'signingAuthority': 'DELHI RTO', 'timestamp': [28, 26, 28, 30, 2
6, 30, 28, 27, 28, 27, 31, 31, 31, 26, 34, 29, 34, 30, 35, 29]} from ('127.0
.0.1', 50004)
Replay attack detected.
```

```
python
DELHI RTO is listening on port: 4001
Received message: {'req_type': 3, 'rto': 'DELHI RTO', 'doc': [3, 4, 11, 7, 8
, 17, 19, 14, 3, 11, 0, 0, 12, 11, 4, 4, 13, 28, 30, 26, 33], 'timestamp': [
28, 26, 28, 30, 26, 30, 28, 27, 28, 27, 31, 27, 29, 27, 30, 33, 31, 35, 33,
33]} from ('127.0.0.1', 49939)
```

```
python
INDIAN RTO is listening on port: 4000
Received message: {'req_type': 2, 'rto': 'KERALA RTO', 'requesting_rto': 'DE
LHI RTO', 'timestamp': [28, 26, 28, 30, 26, 30, 28, 27, 28, 27, 31, 27, 31,
33, 33, 26, 32, 26, 28, 34]} from ('127.0.0.1', 49945)
```

```
PS C:\Users\aaamle\OneDrive\Desktop\IIITD Semesters\8th Sem\MSC\Assignment 4> python client.py
Select the RTO Authority:
1. INDIAN RTO
2. DELHI RTO
3. KERALA RTO
0. Quit
3
Enter:
1. Sign new license
2. Verify existing license
2
Enter the encoded document: 3, 4, 11, 7, 8, 17, 19, 14, 3, 11, 0, 0, 12, 11, 4, 4, 13, 28, 30,
26, 33
Enter the signature: 81
Enter the ID of RTO signing authority from above ids: 2
```

Replay Attack