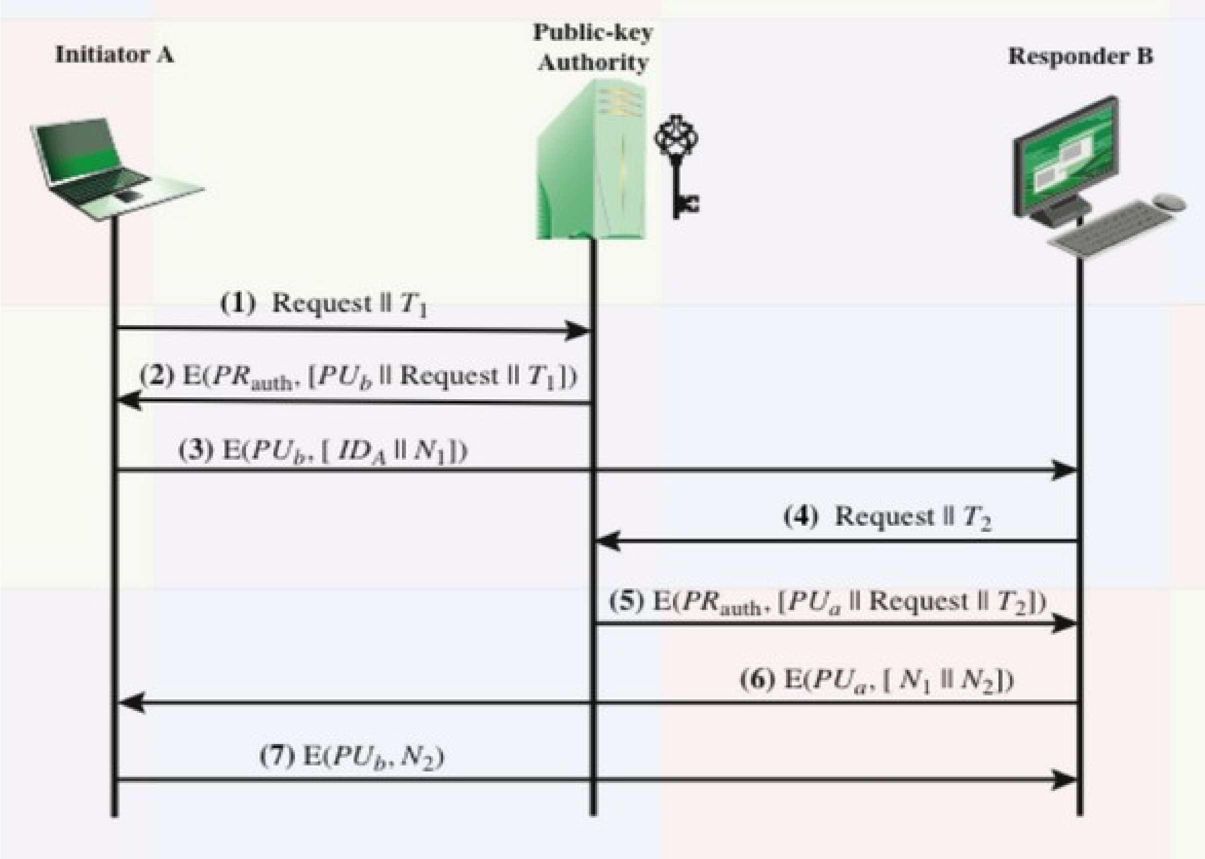
NETWORK SECURITY
Programming Assignment 3

Rhythm Gupta 2018082
Manavjeet Singh 2018295

In this assignment, we have implemented a Public Key Distribution Authority and 2 clients viz A and B such that PKDA will deliver the public key of A to B and public key of B to A. Then A will send messages to B and B will send acknowledgement for each message.


Initiator A
Public-key Authority
Responder B
(1) Request $\|$ T_1
(2) $E(PR_{auth}, [PU_b \| Request \| T_1])$
(3) $E(PU_b, [ID_A \| N_1])$
(4) Request $\|$ T_2
(5) $E(PR_{auth}, [PU_a \| Request \| T_2])$
(6) $E(PU_a, [N_1 \| N_2])$
(7) $E(PU_b, N_2)$

Assumptions:

1. Public key of PKDA is known to A and B.

2. A , B and PKDA know their private keys respectively.
3. PKDA has public keys of A and B.

Steps involved in the process:

1. Initiator A sends a request to PKDA. This request will contain ID of B, time T1 (equal to current time).
2.
   1. PKDA will look into its records and find the public key corresponding to the ID it has received in the request.
   2. It will create a response consisting of the public key of B, request it received containing time T1. This response will be encrypted with the private key of PKDA and sent back to A.
3.
   1. A will decrypt the response using the public key of PKDA.
   2. It will then check that the response satisfies the time constraint.
   3. If it satisfies that, it will extract the public key of B from the response and initiate a connection with B by preparing a message containing its ID.
   4. This message will be encrypted with the public key of B and sent to B.
4.
   1. B will decrypt the message with its private key.
   2. Upon decryption it will extract the ID of A and prepare a request to be sent to PKDA asking for a public key corresponding to A’s ID alongwith current time.
5.
   1. PKDA will extract the public key of A and prepare a response by appending this public key to request and time.
   2. It will encrypt this response with its private key and send it to B.
6.
   1. B will decrypt the response from PKDA with its public key and get the Public Key of A.
   2. It can now reply to messages of A by encrypting replies with the public key of A.

Implementation Details

Libraries Used:

RSA

Pickle
Socket

We have used the Socket library for implementing socket programming between A, B and PKDA.
We have used the RSA library for the purpose of encryption and decryption. Since the message lengths in RSA have to be smaller than key size, we have taken the size of public and private key of PKDA to be 2048 bytes while that of A and B to be 1024 bytes.
Public key objects in RSA libraries have e, n as attributes. So while sending, we have used pickle for serializing and deserializing public keys.
Key information already available with PKDA, A, B is available in pkda_keyinfo, a_keyinfo, b_keyinfo respectively.

How to use:

1. Install Dependencies:

```
pip3 install rsa
pip3 install pickle
```

2. Run following scripts

```
python3 pkda.py
python3 a.py
python3 b.py
```

For generating new set of keys, run genkey.py

File Structure of Project

pkda.py	-> Code for PKDA
a.py -	> Code for initiator A
b.py	-> Code for responder B
genkey.py	-> Script for key generation