RSA BASED PUBLIC KEY CERTIFICATION AUTHORITY

ANANYA GUPTA 2021012
ANANYA GARG 2022068

OBJECTIVE

To understand the RSA based Public Key Certification Authority system, implement it in a programming language, and test its encryption and decryption capabilities.

OVERVIEW

- Each user has a “public-key certificate”, issued by a trusted “certification authority” (or CA). This can be shared with anyone.
- After a certain duration of time, user has to renew the certificate. The certificate has this duration stored as well.
- A receiver must check the certificate, and confirm that received public key is genuine.
- CERTA=[(IDA, PUA, TA, DURA, INFOCA) || ENCPR-CA(Hash(IDA, PUA, TA, DURA, INFOCA))]

RSA ALGORITHM

- Large prime numbers p and q are generated to get n=p*q and ϕ=(p-1)*(q-1).
- This further allowed to generate numbers d and e. Chose e such that: $1 < e < \phi \; and \; gcd(e, \phi) = 1$
- Used keys for encryption and decryption as below:
  - $C = Me(mod\ n)$
  - $M = Cd(mod\ n)$
  - where, M is a plaintext such that M<n and C is the ciphertext.

ENCRYPTION AND DECRYPTION

- We use RSA algorithm for encryption and decryption. Let: **message=(IDA, PUA, TA, DURA, INFOCA)**
- Now using SHA-256, we get the h = hash(message). Concatenate ‘message’ and encrypt(h) using RSA and send them to the client.
- The client decrypts the encrypted part of the certificate and verifies that the hash of the message is equal to the decrypted value.
- After this, the clients exchange messages in between them using gRPC. This also involves encryption and decryption.

AT CA END

- Start listening on the port using gRPC server.
- Connect to the two clients and respond to their messages.
- Do the following for each of the clients:
  - Send CA's public key to the client.
  - Secondly, register the client by adding it in the dictionary of clients.
  - Lastly, it issues the certificate of the other client and sends it to the current client. (i.e. Certificate of B is sent to A and vice versa).
- The CA stops execution after both the clients have been handled.

AT CLIENT END

- The client connects to one of the ports of the server.
- It starts sending and receiving messages using gRPC.
- Depending on the messages:
  - First, it asks for the public key of CA and stores it in an instance of the client class.
  - Second, clients sends the request for registering itself.
  - Lastly, it asks for the Certificate of the other client. (i.e. Certificate B requested by A and vice versa)
- After receiving the certificate, client starts the message conversation with other client on receiving the input, via RSA-based encryption and decryption.