Certification Authority

SIL 765 - Assignment 3

Nichit Bodhak Goel (2017MCS2089)
Shadab Zafar (2017MCS2076)

Project Selection:

A1 = Last 4 digits of entry number of first student = 2089
A2 = Last 4 digits of entry number of second student = 2076

K = A1 + A2 mod 4 = 4165 mod 4 = 1

Project 1: Certification Authority (CA)

Problem statement

You are required to (a) build a CA, and (b) build clients that wish to send messages suitably encrypted with public key of receiver, but only after they know the other client's public key in a secure manner. There are two ways for client A to know the public key of another client, B: (a) get it from CA (which is rarely done), or receive a "certificate" from B itself. (Pl. refer Slide 81 of Lecture 6). We will presently limit the fields in the "certificate" to the following:

CERTA = 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.

To do so, you will need to:

- ensure that clients already (somehow) know the public key of the certification authority,
- CA has the public keys of all the clients, and corresponding to which the clients themselves have their corresponding private keys with themselves,
- messages between CA and clients are encrypted using RSA algorithm (with fixed parameters (n, p, q)) and using CA's private key,
- messages sent/received between clients (once they have each other client's public key) are encrypted using RSA algorithm using the same parameters (n, p, q).

• find a way to generate and encode “current time”.

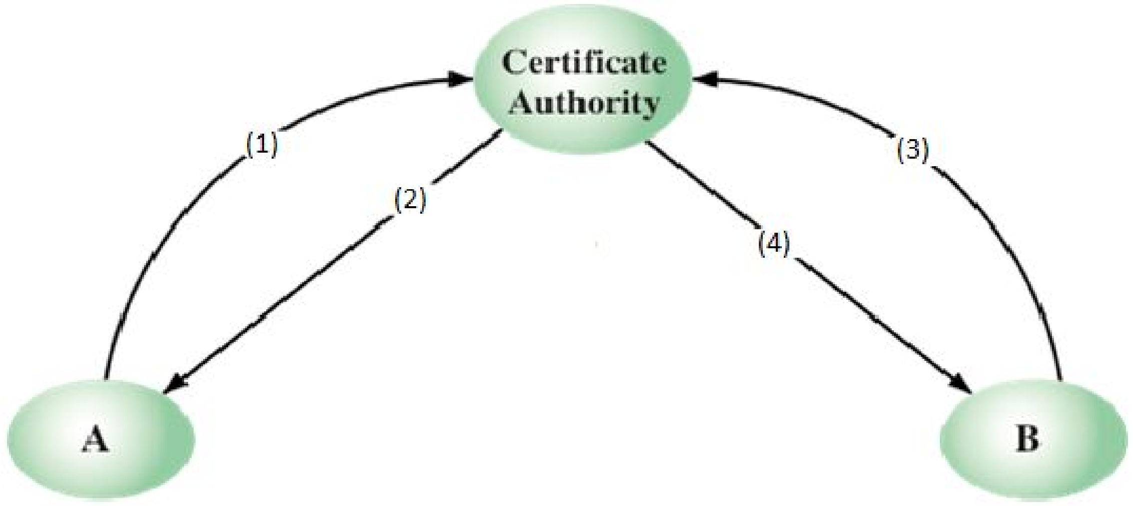
Once the public keys have become known to the clients, the clients should encrypt and send “hello xxx” message using their private keys.

Introduction

Certification Authority is a trusted third party which issues the digital certificates to the users which helps them to prove the ownership of their public key to other users. These certificates are valid for a certain duration of time after which the user have to request for the certificate again. A sender will make its public key known via the certificate and the receiver will verify the certificate thereby confirming the validity of the public key.

Procedure

1.Certificate Creation (Communication between client and CA) :-



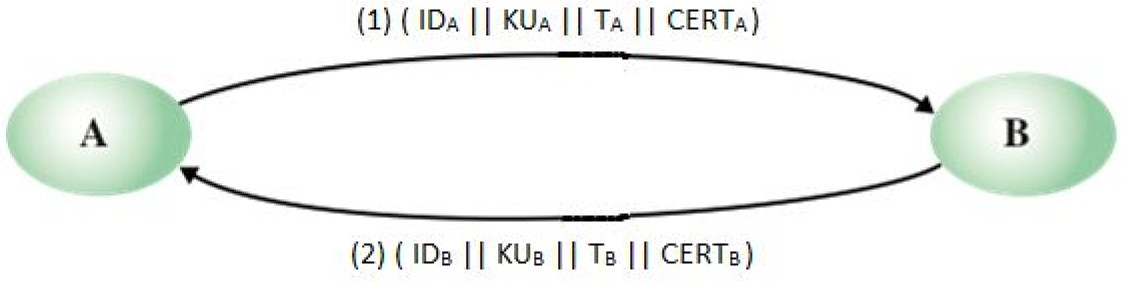
Obtaining Certificates from CA

Client Request:- Client will send its public key (KU_A) and identity (ID_A) encrypted with the public key of the CA (PU_X).
Client Request :- ENC_{PUX} (ID_A).

CA Reply:- CA will send the timestamp (T_A), public key (KU_A), identity (ID_A) and the certificate which will contain the hash of ID of client, public key of client and the timestamp specifying the time of issuance encrypted with the private key of CA i.e.
$CERT_A = ENC_{PRX}(Hash(ID_A, KU_A, T_A))$.
CA reply :- $ID_A || KU_A || T_A || CERT_A$.

2. Communication between clients

I. Exchange of Certificates :-


(1) (IDA || KUA || TA || CERTA)
A
B
(2) (IDB || KUB || TB || CERTB)

I. Exchanging Certificates

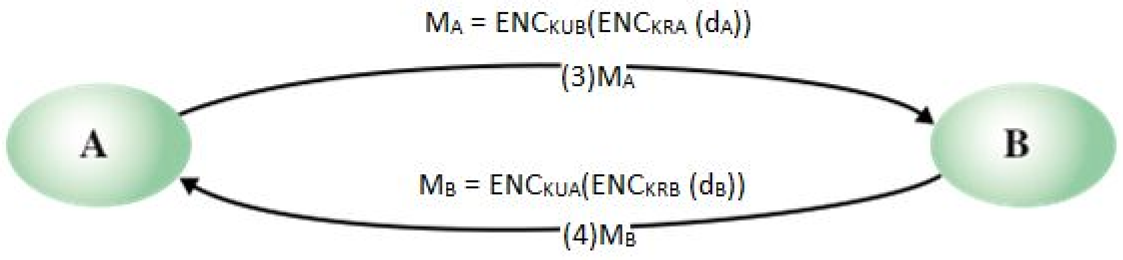
Client A request to Client B :- Client A will send its certificate along with its ID and Public key and timestamp to Client B.
Client A to B :- $ID_A || KU_A || T_A || CERT_A$.

Client B reply to Client A :- Now, Client B will verify the certificate of A by taking hash of (ID_A || KU_A || T_A) and decrypting the $CERT_A$ with the public key of CA (PU_X) and checking if they are equal.
$Hash(ID_A, KU_A, T_A) = DEC_{PUX}(CERT_A)$

After verification, Client B will send its certificate along with its ID and Public key and timestamp to Client A.
Client B to A :- ID_B || KU_B || T_B || $CERT_B$.

Similarly, Client A will verify the certificate of B.
Now, both clients are having public keys of each other with them.

II. Exchanging Messages :-


$M_A = ENC_{KUB}(ENC_{KRA} (d_A))$
(3)M_A
A
B
$M_B = ENC_{KUA}(ENC_{KRB} (d_B))$
(4)M_B

II. Exchanging Messages

Client A to Client B :- Client A will encrypt the data (d_A) first with its own private key (KR_A) to ensure authentication and after the it will encrypt it with the public key (KU_B) of client B to ensure message integrity and confidentiality. Then, Client A will send this message to Client B i.e. $M_A = ENC_{KUB}(ENC_{KRA} (d_A))$.
Client A to B :- M_A

Client B to Client A :- Similarly, Client B will send message to Client A i.e. $M_B = ENC_{KUA}(ENC_{KRB} (d_B))$.
Client B to A :- M_B

On Certification Authority Server

1) Firstly, CA server will be started so that it can accept requests from the clients.
2) When a request is received for the certificate, it will reply with the certificate along with the timestamp to that client.

```
> python3 server.py

> Certification authority now listening on port: 7070

>>> New client connected: ('127.0.0.1', 58666)

> Received request for new certificate:
Shadab|835209960655|1000076001443
> Time of issuing:  2018-04-27 16:45:43.658237
> Certificate:
wp0Pw4PDnFpxwrrCtXTDu8OfS1XCoCfCocOfS8Kbw7R2w5pSw4BBGnMOOlzCisOrw63CpM
Kde8KkwrRkbUfDh2TCr8Kiw5ZvNsKnwodWesKAw5hgw5PDkMOgwpBAB0zDisKxwplcVmkH
wo1Lw6XDnFfCiEfDpMOaw6nCog==

>>> Client disconnected:  ('127.0.0.1', 58666)

>>> New client connected: ('127.0.0.1', 58672)

> Received request for new certificate:
Nichit|927326331365|1000076001443
> Time of issuing:  2018-04-27 16:45:44.739411
> Certificate:
wozCrcKrw7zDpQDDvEjDtMKbNsOkwrByDVtYw5HCoQlew6rDo8KrccOhCUjCgMKKQQw8Ic
O8CsOwJw3DlTRdw6YfJ8OjW1DDjMOvwpvCuW7DrCnCicO1wqrClmB2Jn7Cs1LCpn3DtcK/
w6pxwoPDk8KHQAVMwrXDgMO2
```

```
>>> Client disconnected:  ('127.0.0.1', 58672)
```

On Client 1

1) First, Client 1 will check if its certificate is valid or not. If not, then request the CA for the certificate and get the certificate and timestamp of the certificate from the CA server.
2) Otherwise, listen for the request from the client 2.
3) Once, request is received from the client 2. Verify the digital certificate of the client 2.
4) Then, Client 1 will send its public key , ID, timestamp of certificate and digital certificate to client 2.
5) Receive the message from client 2 and decrypt the message using the private key of client 1 and public key of other client and read the data sent “Hello, Client 1”.
6) Send the “Hello, Client 2” message to client 2 after encrypting it with client 1’s private key and client 2’s public key.

```
My ID:  Shadab
My Public Key:  (835209960655, 1000076001443)

Sending request for a new certificate to CA

Received certificate from CA.

Shadab|835209960655|1000076001443|2018-04-27
16:45:43.658237|wp0Pw4PDnFpxwrrCtXTDu8OfS1XCoCfCocOfS8Kbw7R2w5pSw4BBGn
MOOlzCisOrw63CpMKde8KkwrRkbUfDh2TCr8Kiw5ZvNsKnwodWesKAw5hgw5PDkMOgwpBA
B0zDisKxwplcVmkHwo1Lw6XDnFfCiEfDpMOaw6nCog==

======================================================================
=====================

> I'm now listening on port: 7171
```

```
=================================

>> User ID:  Nichit
>> Public Key:  (927326331365, 1000076001443)
>> Issue Time:  2018-04-27 16:45:44.739411
>> Certificate:
wozCrcKrw7zDpQDDvEjDtMKbNsOkwrByDVtYw5HCoQlew6rDo8KrccOhCUjCgMKKQQw8Ic
O8CsOwJw3DlTRdw6YfJ8OjW1DDjMOvwpvCuW7DrCnCicO1wqrClmB2Jn7Cs1LCpn3DtcK/
w6pxwoPDk8KHQAVMwrXDgMO2

> Certificate is valid for given public key.

=================================

Received msg from client: Hello, Shadab
```

On Client 2

1) First, Client 2 will check if its certificate is valid or not. If not, then request the CA for the certificate and get the certificate and timestamp of the certificate from the CA server.
2) Otherwise, Client 2 will send its public key , ID, timestamp of certificate and digital certificate to the other client.
3) Verify the received digital certificate of Client 1.
4) Send the “Hello, Client 1” message to client 1 after encrypting it with client 2’s private key and client 1’s public key.
5) Receive the message and decrypt the message using the private key of client 1 and public key of client 2 and read the data sent “Hello, Client 2”.

```
My ID:  Nichit
My Public Key:  (927326331365, 1000076001443)
```

```
Sending request for a new certificate to CA

Received certificate from CA.

Nichit|927326331365|1000076001443|2018-04-27
16:45:44.739411|wozCrcKrw7zDpQDDvEjDtMKbNsOkwrByDVtYw5HCoQlew6rDo8Krcc
OhCUjCgMKKQQw8IcO8CsOwJw3DlTRdw6YfJ8OjW1DDjMOvwpvCuW7DrCnCicO1wqrClmB2
Jn7Cs1LCpn3DtcK/w6pxwoPDk8KHQAVMwrXDgMO2

======================================================================
=====================

> Now connecting to client on port: 7171

> Sending my public key & certificate

=================================

> Client has sent its public key & certificate:

>> User ID:  Shadab
>> Public Key:  (835209960655, 1000076001443)
>> Issue Time:  2018-04-27 16:45:43.658237
>> Certificate:
wp0Pw4PDnFpxwrrCtXTDu8OfS1XCoCfCocOfS8Kbw7R2w5pSw4BBGnMOOlzCisOrw63CpM
Kde8KkwrRkbUfDh2TCr8Kiw5ZvNsKnwodWesKAw5hgw5PDkMOgwpBAB0zDisKxwplcVmkH
wo1Lw6XDnFfCiEfDpMOaw6nCog==

> Certificate is valid for given public key.

=================================

Received msg from client: Hello, Nichit
```