SIL Network & System Security; Assignment No. 2, due THU Feb 28, 2019

Listed below, you will find brief description of 2 projects, numbered 0 through 1. In groups of 2, you are required to pick one (see algorithm below), complete that project and submit a report (with a working system) on or before THU Feb 28. The outcome will be evaluated by me.

The algorithm to pick a project: you are required to pick project numbered 0, 1 as determined by k = A1+A2 mod 2, where

A1 = last_4_digits_of_entry_no_of_first_student, and A2 = last_4_digits_of_entry_no_of_second_student.

The submission will consist of three parts:

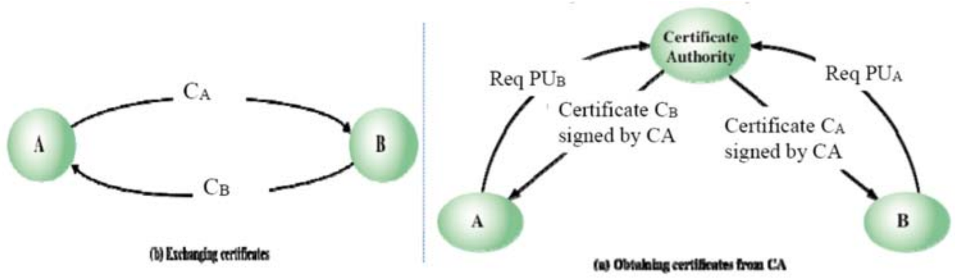
1. 2 to 4 page document describing the system you have designed (including all assumptions you have made),
2. the code as a separate file, and
3. 5 to 8 slides that you will use to present your work.

Project no. 0: Certification Authority (CA)

You are required to (a) build a CA, and (b) build clients that wish to confidentially 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) Receive a "certificate" from B itself, or
(b) Get it from CA (which is rarely done).



C_A
A
B
C_B
Certificate Authority
Req PU_B
Req PU_A
Certificate C_B signed by CA
Certificate C_A signed by CA
A
B

(b) Exchanging certificates

(a) Obtaining certificates from CA

We will presently limit the fields in the "certificate" to the following:

$\text{CERT}_A = \text{ENC}_{PR\text{-}CA}$ (ID_A, PU_A, T_A, ~~DUR_A, $INFO_{CA}$~~)

where

- PR-CA is private key of certification authority (PU-CA is public key of certification authority)
- ID_A is user ID,
- PU_A is public key of A,
- T_A is time of issuance of certificate.

To do so, you will need to:

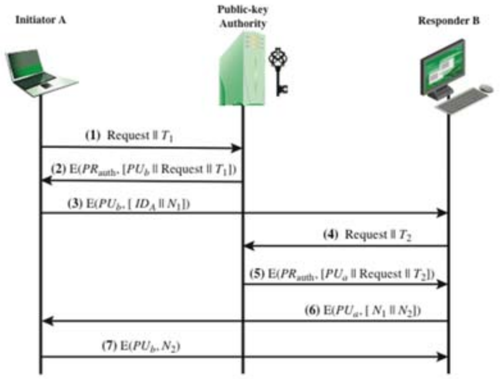
- Decide you will use method (b) to obtain each other's public key,
- Assume:
  1. that clients already (somehow) know the public key of the certification authority,
  2. that the clients have their corresponding private keys with themselves, and
  3. that CA has the public keys of all the clients,
- Messages from CA to clients are encrypted using RSA algorithm and CA's private key,
- Encrypted messages are sent/received between clients once they have each other client's public key, and finally
- Find a way to generate and encode "current time".

Use the above to ensure client A can send 3 messages to B, viz Hello 1, Hello 2, and Hello 3. Client B in turn responds with ACK 1, ACK 2, and ACK 3 to messages received from A.

Project no. 1: Public Key Distribution Authority (PKDA)

You are required to (a) build a PKDA, and (b) build clients that wish to confidentially send messages suitably encrypted with public key of receiver, but only after they know the other client's public key in a secure manner.

You are required to (a) build a, and (b) build clients that wish to send messages suitably encrypted with public key of receiver but of course only after they know each other's public key in a secure manner. Specifically use the scheme described below.


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)

To do so, you will need to:

- Assume:
  4. that clients already (somehow) know the public key of the distribution authority, PKDA,
  5. that the clients have their corresponding private keys with themselves, and
  6. that PKDA has the public keys of all the clients,
- Messages between PKDA and clients are encrypted using RSA algorithm and PKDA's private key,
- Encrypted messages are sent/received between clients once they have each other client's public key, and finally
- Find a way to generate and encode "current time" and "nonces".

Use the above to ensure client A can send 3 messages to B, viz Hi 1, Hi 2, and Hi 3. Client B in turn responds with Got-it 1, Got-it 2, etc. to messages received from A.