CSE 350/550 Network Security; Assignment No. 4, due Sunday April 19, 2020

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 to pick a project), complete that project and submit a report (with a working system) on or before Sunday April 19. The outcome will be evaluated by me, and the TAs.

The algorithm to pick a project:

> you are required to pick project numbered 0, 1, or 2 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. a 2 to 4 page document describing the system you have designed,
2. the code as a separate file, and
3. 5 to 8 slides that you will use to present your work.

Project no. 0: Implementation of Kerberos

You are required to (a) build an AS, a TGS, an application server (indeed a web server), and (b) build one or more clients that wish to connect to the application server using the services of the AS and TGS (Please see my slides on Kerberos).

To do so, you will need to:

- ensure that AS and TGS have the required information concerning the server(s) and the clients,
- AS, TGS, the servers and clients are able to generate/interpret ticketsand thus provide services to clients as when sought.

Note, access to the server (in this case a web server) requires authentication.

Once a session key has been established between a client and the web application server, the client can download the home page of the web server.

Project no. 2: Diffie-Hellman key generation and exchange

You are required to build clients A and B that wish to send messages from B to A (only B to A) but encrypted using Elgamal Cryptosystem, but only after having exchanged messages that finally result in computation of one-time keysusing the Diffie Hellman protocol. (Pl. refer to relevant slides fromLecture 8 Part 1.)

The Elgamal Cryptosystem uses the parameters (q, a). These are known to each other using "other means".

To ensure that man-in-the-middle attack is not possible, the two clients A and B send initial messages to each other by adding to each message a MAC (or an "integrity check") that itself is based on HMAC algorithm and uses a shared "secret".

To do so, you will need to:

- ensure that clients already (somehow) know the shared "secret" to be used with HMAC,
- assume that Diffie-Hellman parameters (q, a) are already fixed, and known to them,
- (once they know how to generate or compute one-time passwords) messages sent from B to A are encrypted using Elgamal cryptosystem with parameters, (q, a).

Once the one-time keys have been computed (or can be computed on the fly), B should encrypt and send messages "hello 1", "hello 2", "hello 3".