Duration : 90 minutes
Class : INT3307E
Open books and notes, no notebooks, no mobile phones
No discussion or exchange of documents between students during the exam

Final Exam
Network Security

(3 problems, 3 pages, point values given in parentheses, 10 maximum)

1. Secret key distribution and user authentication (3 points)

Consider the following handshake protocol for session establishment:

1. A → B: $E(PU_b, [N_1 \| ID_a])$
2. B → A: $E(PU_a, [N_1 \| N_2])$
3. A → B: $E(PU_b, N_2)$
4. A → B: $E(PU_b, E(PR_a, K_s))$

Here N_1 and N_2 are nonces chosen by A and B, PU_a and PU_b are A's and B's public keys, respectively, ID_a is A's identifier, PR_a is A's private key, K_s is a secret key generated by A uniquely for each session.

a. (2 points)

Which of the following functionalities does the above protocol provide?

- Allowing A and B to negotiate cryptographic algorithms: If yes then who is the proposer and who is the responder?
- Allowing A and B to share a secret key: If yes then what algorithm can be used?
- Allowing A to authenticate B: If yes then by using which message(s) and why?
- Allowing B to authenticate A: If yes then by using which message(s) and why?

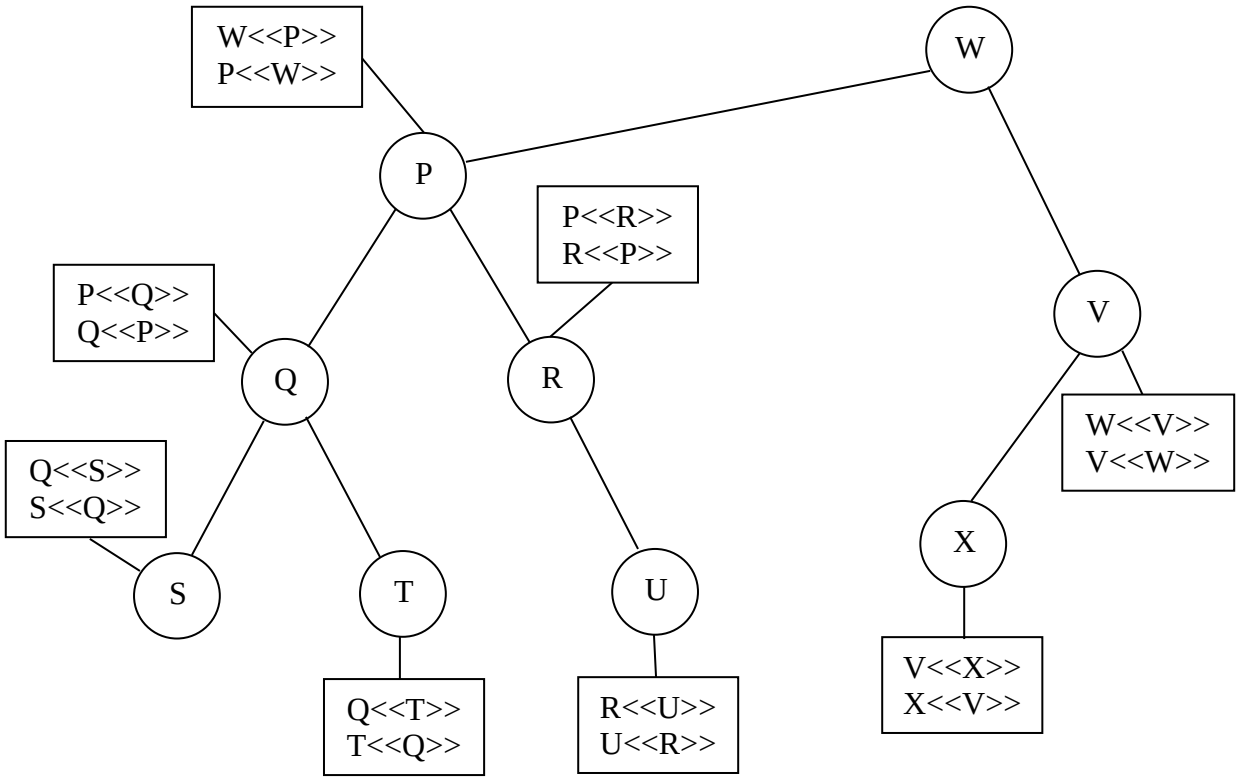
b. (1 point)

Explain the rationale for $E(PR_a, K_s)$ in the fourth message (How can the protocol be attacked if K_s is only encrypted with PU_b? Why can the attack be prevented if K_s is encrypted twice, once with PR_a and then again with PU_b?)

2. X.509 certificates (3 points)

Consider the X.509 hierarchy in the next page.

Suppose that user A has obtained a certificate from certification authority S and user B has obtained a certificate from certificate authority X. Give the chain of certificates that allows A to verify that the certificate of B issued by X is valid. Explain how the verification is proceeded.



3. Transport-level security (4 points)

Consider the TLS Handshake Protocol. Suppose that the RSA key exchange method is used.

a. (1 point)

Draw the most secure exchange of messages expected for this scenario.

b. (2 points)

Describe the parameters associated with each situation-dependent message and with the *client_key_exchange* message.

c. (1 point)

During which of the following times is there a change in the security parameters (including the current encryption algorithm, the pending encryption algorithm, the current hash function, the pending hash function, the current client write encryption key, the pending client write encryption key, the current server write encryption key, the pending server write encryption key, the current client write MAC secret, the pending client write MAC secret, the current server write MAC secret, the pending server write MAC secret, the current client write IV, the pending client write IV, the current server write IV, the pending server write IV) at either the client or server?

- At the client before sending the client_hello message
- At the server before receiving the client_hello message
- At the client after sending the client_hello message and before receiving the server_hello message

- At the server after receiving the client_hello message and before receiving the change_cipher_spec message from the client
- At the client after receiving the server_hello message and before sending the change_cipher_spec message to the server
- At the server after sending the server_hello message and before receiving the change_cipher_spec message from the client
- At the client after sending the change_cipher_spec message and before sending the finished message to the server
- At the server after receiving the change_cipher_spec message and before receiving the finished message from the client
- At the client after sending the finished message to the server and before receiving the change_cipher_spec message from the server
- At the server after receiving the finished message from the client and before sending the change_cipher_spec message to the client
- At the client after receiving the change_cipher_spec message and before receiving the finished message from the server
- At the server after sending the change_cipher_spec message and before sending the finished message to the client
- At the server after sending the finished message to the client
- At the client after receiving the finished message from the server