Vietnam National University, Hanoi | Faculty of Information Technology
University of Engineering and Technology | Monday, December 29, 2014

Duration : 120 minutes | *Open books and notes, no notebooks, no mobile phones*
Class : INT3093 | *No discussion or exchange of documents between students during the exam*

Final Exam Solution
Network Security

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

1. Key distribution and user authentication (2.5 points)

a. (1 point)

TS_1 allows AS to verify that the client's clock is synchronized with that of AS.

In the case where the client's clock is not synchronized with that of AS, for example, suppose that the client's clock is 2 hours later than the AS' time, the lifetime for $Ticket_{tgs}$ is only one hour, and the transmission time of message (2) is 1 minute. Then, the client can never use $Ticket_{tgs}$. For example, at 9 am at the AS, when the AS creates $Ticket_{tgs}$ and sends message (2), the local time at the client is already 11 am. When the client receives message (2), the client's time is 11:01 am, while the ticket's expiration time is 9 am + 1 hour = 10 am. That means when the client receives the ticket, it has already expired.

b. (0.5 point)

As the client C can't decrypt $Ticket_{tgs}$ to get TS_2 and $Lifetime_2$ inside it, AS needs to add these fields outside the ticket so that the client C can learn its expiration time.

c. (1 point)

Message (6) enables the client C to authenticate the service server V. Because the message was encrypted by the session key, C is assured that it could have been created only by V. The inclusion of TS_5 in this message assure C that this is not a replay of an old message. There is nothing wrong if we replace $TS_5 + 1$ with TS_5. It is because the original timestamp TS_5 is also something that C has known in advance so it can also be used by C to verify the authenticity of message (6). $TS_5 + 1$ is used in Kerberos Version 4 simply to mean that message (6) is also an authenticator like $Authenticator_C$ and it is created at a time later than TS_5.

2. Transport-level security (2.5 points)

a. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	?	null	?
MAC	null	?	null	?
Encr. key	null	?	null	?
MAC secret	null	?	null	?

IV	null	?	null	?

b. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	?	null	?
MAC	null	?	null	?
Encr. key	null	?	null	?
MAC secret	null	?	null	?
IV	null	?	null	?

c. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	DES	null	DES
MAC	null	MD5	null	MD5
Encr. key	null	?	null	?
MAC secret	null	?	null	?
IV	null	?	null	?

d. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	DES	null	DES
MAC	null	MD5	null	MD5
Encr. key	null	?	null	?
MAC secret	null	?	null	?
IV	null	?	null	?

e. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	DES	null	DES
MAC	null	MD5	null	MD5
Encr. key	null	3456	null	4567
MAC secret	null	1234	null	2345

IV	null	5678	null	6789

f. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	DES	?	null	DES
MAC	MD5	?	null	MD5
Encr. key	3456	?	null	4567
MAC secret	1234	?	null	2345
IV	5678	?	null	6789

g. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	DES	null	DES
MAC	null	MD5	null	MD5
Encr. key	null	4567	null	3456
MAC secret	null	2345	null	1234
IV	null	6789	null	5678

h. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	DES	DES	?
MAC	null	MD5	MD5	?
Encr. key	null	4567	3456	?
MAC secret	null	2345	1234	?
IV	null	6789	5678	?

i. (0.25 point)

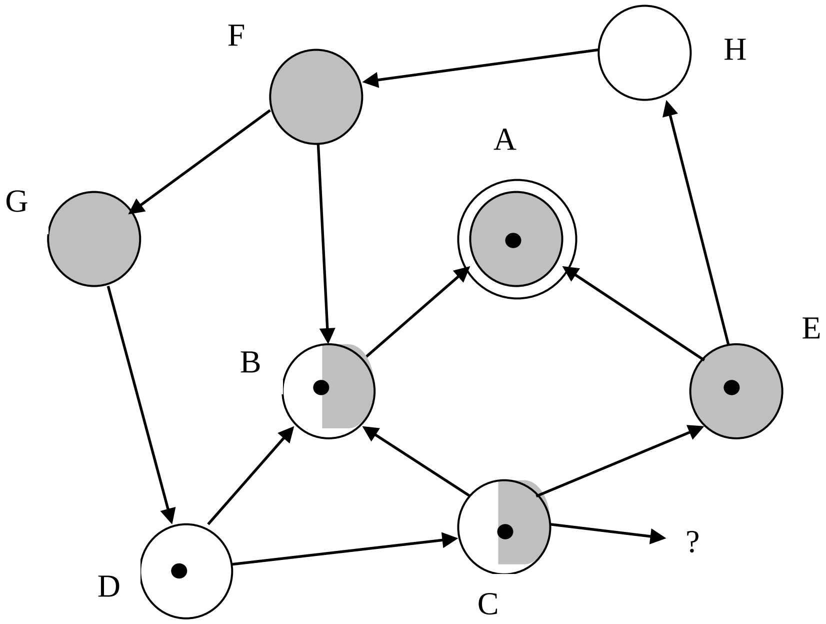
	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	DES	?	DES	?
MAC	MD5	?	MD5	?
Encr. key	4567	?	3456	?
MAC secret	2345	?	1234	?

IV	6789	?	5678	?

j. (0.25 point)

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	DES	?	DES	?
MAC	MD5	?	MD5	?
Encr. key	3456	?	4567	?
MAC secret	1234	?	2345	?
IV	5678	?	6789	?

3. Electronic mail security (2.5 points)



4. IP Security (2.5 points)

a. (0,5 điểm)

The receiver finds that the received packet falls within the replay window, but the packet is not new, so it is discarded.

b. (0,5 điểm)

The receiver finds that the received packet is to the right of the replay window, so the MAC is checked. As the packet is authenticated, the window is advanced so that 457 is the right edge of the window, the correponding slot in the window is marked, and the packet is accepted.

c. (0,5 điểm)

The receiver finds that the received packet falls within the replay window and is new, so the MAC is checked. As the packet is authenticated, the correponding window is marked and the packet is accepted.

d. (0,5 điểm)

The receiver finds that the received packet is to the left of the window, therefore the packet is discarded.

e. (0,5 điểm)

The receiver finds that the received packet falls within the replay window but is not new, so the packet is discarded.

Vietnam National University, Hanoi — Faculty of Information Technology
University of Engineering and Technology — Monday, December 29, 2014

Duration : 120 minutes — *Open books and notes, no notebooks, no mobile phones*
Class : INT3093 — *No discussion or exchange of documents between students during the exam*

Final Exam
Network Security

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

1. Key distribution and user authentication (2.5 points)

Consider the Kerberos Version 4 authentication dialogue.

f. (1 point)

What is the rationale for the field TS_1 in message (1)? Give an example for why we need TS_1.

g. (0.5 point)

What is the rationale for the fields TS_2 and $Lifetime_2$ located outside the $Ticket_{tgs}$ structure in message (2)?

h. (1 point)

What is the rationale for message (6)? Why is TS_5 used in this message? Is there anything wrong if we replace $TS_5 + 1$ with TS_5? Explain why.

2. Transport-level security (2.5 points)

As taught in the classroom, for any given system, whether it is a client or a server, SSL defines a write state and a read state. The write state defines the security information for data that the system sends, and the read state defines the security information for data that the system receives. The matrices below show the systems' initial read and write states.

Client

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	?	null	?
MAC	null	?	null	?
Encr. key	null	?	null	?
MAC secret	null	?	null	?
IV	null	?	null	?

Server

	Write		Read	
	Cnt	Pnd	Cnt	Pnd
Encryption	null	?	null	?
MAC	null	?	null	?
Encr. Key	null	?	null	?
MAC secret	null	?	null	?
IV	null	?	null	?

As the matrices indicate, SSL actually defines two separate read and write states for each system. One of the states is current and the second is pending. Both the client and the server, therefore, maintain a total of four different states: the current write state, the pending write state, the current read state, and the pending read state. The matrices use the abbreviations "Cnt" and "Pnd" for current and pending, respectively. The matrices also show the key elements of a state. They are the encryption algorithm (abbreviated "Encryption"), the message integrity algorithm (abbreviated "MAC" for Message

Authentication Code), the encryption key (abbreviated “Encr. Key”), the MAC secret, and the IV (Initialization Vector).

Suppose that by executing the SSL Handshake protocol, the systems agree to use the Data Encryption Standard (DES) for symmetric encryption and Message Digest 5 (MD5) for message integrity. The client write MAC secret, the server write MAC secret, the client write encryption key, the server write encryption key, the client write IV, and the server write IV are 1234, 2345, 3456, 4567, 5678, and 6789, respectively.

k. (0.25 point)

Write the matrix describing the client’s states after sending the *client_hello* message and before receiving the *server_hello* message.

l. (0.25 point)

Write the matrix describing the server’s states before receiving the *client_hello* message.

m. (0.25 point)

Write the matrix describing the server’s states just after sending the *server_hello* message.

n. (0.25 point)

Write the matrix describing the client’s states just after receiving the *server_hello* message.

o. (0.25 point)

Write the matrix describing the client’s states just before sending the *change_cipher_spec* message.

p. (0.25 point)

Write the matrix describing the client’s states after sending the *change_cipher_spec* message and before sending the *finished* message.

q. (0.25 point)

Write the matrix describing the server’s states just before receiving the *change_cipher_spec* message.

r. (0.25 point)

Write the matrix describing the server’s states after receiving the *change_cipher_spec* message and before receiving the *finished* message.

s. (0.25 point)

Write the matrix describing the server’s states after sending the *change_cipher_spec* message and before sending the *finished* message.

t. (0.25 point)

Write the matrix describing the client’s states after receiving the *change_cipher_spec* message and before receiving the *finished* message.

3. Electronic mail security (2.5 points)

A user A maintains a PGP public key ring with the fields **Public Key**, **User ID**, **Owner Trust**, and **Signatures** as follows:

Public Key	*PU*A	*PU*B	*PU*C	*PU*D	*PU*E	*PU*F	*PU*G	*PU*H
User ID	A	B	C	D	E	F	G	H
Owner Trust	*Ultimate*	*Usually trusted*	*Usually trusted*	*Not trusted*	*Always trusted*	*Always trusted*	*Always trusted*	*Not trusted*
Signatures	-	A	B, E, I	B, C	A, H	G, B	D	F

The **Key Legitimacy** fields are computed on the basis of the attached signatures as follows:

- If the owner is A then the public key is *legitimate*.
- If at least one signature has a signature trust value of *ultimate*, then the public key is *legitimate*.
- Otherwise, PGP computes a weighted sum of the trust values. A weight of 1 is given to signatures that are *always trusted* and ½ to signatures that are *usually trusted*. When the total of weights of the introducers of a **Public Key**/**User ID** combination reaches 1, the public key is considered *legitimate*.
- In all remaining cases, the public key is considered *illegitimate*.

Draw the corresponding PGP trust model.

4. IP Security (2.5 points)

Suppose that the current replay window spans from 200 to 455. The receiver has received all the packets with odd sequence numbers in the replay window but none with even sequence numbers. What will the receiver do with each of the following packets?

a. *(0,5 điểm)*

A non-authenticated packet with sequence number 455.

b. *(0,5 điểm)*

An authenticated packet with sequence number 457.

c. *(0,5 điểm)*

An authenticated packet with sequence number 202.

d. *(0,5 điểm)*

An authenticated packet with sequence number 201.

e. *(0,5 điểm)*

A non-authenticated packet with sequence number 202.

Note that the above packets are received in the sequence that they are ordered.