18-731
Final Exam

13 December 2012

Name:

Andrew user id:

Scores:

Problem 0 (10 points):

Problem 1 (20 points):

Problem 2 (10 points):

Problem 3 (25 points):

Problem 4 (15 points):

Problem 5 (25 points):

Problem 6 (10 points):

Problem 7 (20 points):

Total (135 points):

Problem 0: Warm up questions (10 Points)

1. (2 points) With a simple example of signing a bit 0 or 1, explain how one-time signatures provide the same properties as a regular signature. *1 point for mentioning about private*

   and public keys, and 1 point for mentioning the non-repudiation property. Only those who generate the secret value S_0 for bit 0 or S_1 for bit 1 can help verify the bit based on the public key that can be derived P_0 or P_1 for bit 0 and bit 1 respectively.

2. (4 points) Please circle the correct answer

   (a) a) An attacker that obtains a web server's certificate can use the certificate to establish an SSL connection with a client such that the client believes he or she is connected to the legitimate server. TRUE FALSE *False. The attacker needs to compromise keys.*

   (b) b) The use of digital signatures provides non-repudiation of a message and ensures the authenticity of a message, whereas MACs only ensure the authenticity of a message. TRUE FALSE *True*

3. (1 point) Consider a CBC mode encryption with key K, where IV = H(plaintext). Is this a secure encryption? Briefly state why or why not. (Hint: recall that the IV is sent with the ciphertext.)

4. (1 point) We discussed NV RAM write exhaustion attacks in class. Based on the research project presentation of team "while 1==1", was a NV RAM write exhaustion attack successful against the Infineon TPM chip?

5. (1 point) Based on the research project presentation of team "Drop Database", what is an RFID worm?

6. (1 point) Based on the research project presentations of the groups that analyzed the security of SafeSlinger, mention one vulnerability that was identified.

7. (1 Bonus point) Consider the 160-bit long string f = 0xFFFF...FFFF. If someone could identify the 160-bit long string x, such that SHA-1(f || x) = 0, what would the impact be on trustworthy computing? (4 Bonus points) What is the probability that x exists?

8. (2 Bonus points) Where in Pittsburgh should one go to tattoo their most exciting 18731 course materials?

Problem 1: Ad-hoc Network Key Agreement (20 Points)

A vehicular manufacturer wants to provide a secure way of allowing their customers to communicate data between the car (A) and the customer's smartphone (B). The main challenge, however, is that A and B do not share any prior secret. The manufacturer tries to come up with different solutions shown below. They come up with different ideas of leveraging an Out-Of-Band (OOB) channels to authenticate the Diffie-Hellman key exchange messages.[1] They choose *light* in a closed glove compartment as the OOB channel. The driver puts his smartphone inside the glove compartment, and closes it, and the car transmits messages to the smartphone via the light source inside the glove compartment. The smartphone reads the messages via its ambient light sensor. This unidirectional OOB channel provides both secrecy and authenticity. The messages sent via the wireless channel are labeled "via BT" for wireless (e.g., Bluetooth), and the messages sent via the OOB channel are labeled "via light". In all the protocols shown below, the user presses the "Cancel" button on the car to abort the protocol.

1) (6 points) Is the following protocol secure? Please justify your answer, and provide a fix to the protocol if it is broken.

   $A \rightarrow$ (via light) B : a short (20 bit) symmetric key, K_s
   $A \rightarrow$ (via BT) $B : g^a || MAC_{K_s}(g^a)$
   B computes $K = (g^a)^b$
   $B \rightarrow$ (via BT) $A : g^b || MAC_{K_s}(g^b)$
   A computes $K = (g^b)^a$

   Insecure. The attacker can perform a bruteforce attack to find K_s.

[1] An OOB channel is a channel different from the wireless channel, which a wireless attacker cannot influence. In this example, light signals inside a car's glove compartment is such an OOB channel, as a wireless attacker cannot eavesdrop on or create light signals inside the glove compartment.

2) (8 points) Assume that B's public key, K_B, is publicly known. Hence both A and the attacker, M, knows authentic K_B.

$A \rightarrow$ (via light) $B : c_A = [H(g^a)]_{20}$
$A \rightarrow$ (via BT) $B : g^a$
B verifies if the hash of the received g^a matches c_A; aborts if unmatched
B computes $K = (g^a)^b$
$B \rightarrow$ (via BT) $A : \{g^a || g^b\}_{K_B{}^{-1}}$
A verifies the signature, and computes $K = (g^b)^a$

a) (6 points)Please describe how an attacker, M, would try to impersonate A to B (launching a MitM attack)? *The attacker tries to impersonate A to B and launches a MitM by jamming the transmission of g^a so that B does not receive it and only the attacker receives the value. The attacker transmits g^m to B, where the $[H(g^m)]_{20} == [H(g^a)]_{20}$.*

b) (2 points)What is the computational complexity of the attack mentioned in (a)?

3) (6 points) In this problem, we leverage an OOB channel using *sound*. The car and the phone exchange sound signals (e.g., short beeps). The messages sent via this OOB channel are labeled "via sound". Is this protocol secure? If it is secure, what does the user need to do to ensure that the protocol is secure? If its not secure, please provide a fix to the protocol. *This is secure because both devices performs authentication using the OOB channel. (as long as the user verifies that the sound emission is from the legitimate devices.) This bidirectional OOB channel provides authenticity because the driver checks to see if only these two intended devices are emitting the sound signals, and aborts otherwise.*

A picks a random 20 bit n_A
$A \rightarrow$ (via BT) $B : H(g^a||n_A)$
B picks a random 20 bit n_B
$B \rightarrow$ (via BT) $A : H(g^b||n_B))$
$A \rightarrow$ (via BT) $B : g^a||n_A$
$B \rightarrow$ (via BT) $A : g^b||n_B$
$A \rightarrow$ (via sound) $B : S_A = n_A \oplus n_B$
B computes $S_B = n_B \oplus n_A$, and verifies if $S_A \stackrel{?}{=} S_B$;
If passes, computes $K = (g^a)^b$, else aborts
$B \rightarrow$ (via sound) $A : S_B$
A verifies if $S_A \stackrel{?}{=} S_B$;
If passes, computes $K = (g^b)^a$, else aborts

Problem 2: Message Authentication Code (10 Points)

Having learned much about network security this semester, you are hired as a security consultant for a company running a web server. The company authenticates messages from its clients by executing the following MAC verification code upon receiving the message and MAC tag in the following manner (We assume that the client and the web server securely exchanged a 128 bit AES key, K, for MAC computation.) The *byteWiseCmp()* function performs a byte-by-byte comparison.

```
verifyMAC(key, msg, tag){
    return byteWiseCmp( MAC(key,msg), tag);
}

byteWiseCmp(a,b){
    for (i=0 to size-1) {
        if (a[i] != b[i]) //a[i] refers to i^th byte of a
            return -1;
    }
    return 0;
}
```

Do you think this approach is secure against attacks? Please justify your answer. If it is vulnerable to an attack, please provide a mitigation. (Hint: Is this secure to all types of attacks we have seen in class? (e.g., side-channel attacks))

No it is not secure, because of a timing attack.

Sol1: To provide a defense against the attack, the server needs to perform MAC verification in a way that does not allow the attacker to know which byte is being compared, as shown below.

```
verifyMAC(key, msg, tag){
    tag_s=MAC(key,msg);
    return byteWiseComp(MAC(key, tag_s),MAC(key, tag));
}
```

Sol2:

```
byteWiseCmp(a,b,size){
    retval=0
    for (i=0 to size-1) {
        if (a[i] != b[i]) //a[i] refers to i^th byte of a
            retval=-1;
    }
    return retval;
}
```

Problem 3: Secure Ad-hoc networking (25 Points)

1) Consider four nodes A, B, C and D in an ad-hoc network. Node A wants to establish a route to node D and broadcasts a route request message $RReq_{AD}$ which has a unique sequence number in it. The route reply for $RReq_{AD}$ follows the route D$\rightarrow$ C $\rightarrow$ B $\rightarrow$ A. The route reply message $RRep_{AD}$ is constructed at each hop in the fashion below starting from node D. The K_A^{-1} is the private key of node A and H_{DC} is the hash of the message node D sends to node C. For example, in the first message below, $H_{DC} = H(RReq_{AD}, B, C, D, \{D\}_{K_D^{-1}})$, and
$H_{CB} = H(RReq_{AD}, B, C, D, \{C\}_{K_C^{-1}}, \{D\}_{K_D^{-1}}, H_{DC})$ and so on.

D$\rightarrow$ C: $RReq_{AD}$, A, B, C, D, $\{D\}_{K_D^{-1}}, H_{DC}$
C$\rightarrow$ B: $RReq_{AD}$, A, B, C, D, $\{C\}_{K_C^{-1}}, \{D\}_{K_D^{-1}}, H_{DC}, H_{CB}$
B$\rightarrow$ A: $RReq_{AD}$, A, B, C, D, $\{B\}_{K_B^{-1}}, \{C\}_{K_C^{-1}}, \{D\}_{K_D^{-1}}, H_{DC}, H_{CB}, H_{BA}$

Node A on receiving the message from node B, sees the route reply to its route request $RReq_{AD}$ message to reach node D. Node A attempts to verify the signatures in the same sequence as the nodes in clear text in the route reply message and computes the hash of the message for each intermediate hop. If the signatures and the hashes are verified then it accepts the sequence of nodes in the clear text as the route to node D.

a) (7 points) Does this construction of route reply message actually verify the route to a destination node? Explain why it does verify the path or explain why it does not with an example attack. *No. Node C could be malicious and replay a signature of D of its own ID and compute the hash of the message and convince node B of receiving the signature from node D. Hence, node B will then respond to node A with the final message as in the figure. So, node D need not even exist at the time when node C constructs its message to node B.*

b) (8 points) Would this scheme help verify the path to node D if it had signed $\{H_R, D\}_{K_D^{-1}}$ instead of $\{D\}_{K_D^{-1}}$ and keeping rest of the message construction the same as the route reply message from node D to node C. $H_R = H(RReq_{AD})$. Explain why or why not briefly.

D$\rightarrow$ C: $RReq_{AD}$,A, B , C, D, $\{H_R, D\}_{K_D^{-1}}, H_{DC}$

C→ B: $RReq_{AD}$, A, B, C, D, $\{H_R, C\}_{K_C^{-1}}, \{H_R, D\}_{K_D^{-1}}, H_{DC}, H_{CB}$

B→ A: $RReq_{AD}$, A, B, C, D, $\{H_R, B\}_{K_B^{-1}}, \{H_R, C\}_{K_C^{-1}}, \{H_R, D\}_{K_D^{-1}}, H_{DC}, H_{CB}, H_{BA}$

No. Node C could drop the signatures of other intermediate nodes between itself and node D and compute a new hash because it is not possible to know who computed the hash. Hence A would not know of intermediate nodes being removed on the path.

2) Nodes in an ad-hoc network depend on each other for successfully executing functions such as routing, forwarding etc. Suppose a reputation system called "honeybee" is in place where nodes decide to punish their neighbors if they are found misbehaving (by launching blackhole, rushing attack etc.). The "honeybee" protocol allows a node to kill another node through a sting message, but it dies itself after the sting. The sting message is a RSA based signature which contains the ID of the node being attacked by a sting and the ID of the node inflicting the sting. The sting message once sent by a node is flooded through the entire network.

   a) (2.5 points) What are the incentives and drawbacks if any for the malicious node to launch a slandering attack? *Incentive is to kill a legitimate node but drawback is the malicious node would get killed too.*

   b) (2.5 points) What is the incentive if any for the malicious node to launch a framing attack (a malicious nodes misbehaves that makes a legitimate node think that another legitimate node is misbehaving)? *Malicious node survives, but the legitimate node stings another legitimate node and kills itself. Hence, the malicious node could eventually take down a lot of legitimate nodes without itself being killed.*

   c) (2.5 points) Suppose the network nodes use bloom filters, then can the attacker send sting messages to every other node in the network or sting every other node? Justify your answer with reasoning. *Bloom filter helps eliminates duplicate packets in the network and hence duplicate sting messages will not propagate. The malicious node cannot survive itself if it tries to sting every other node.*

   c) (2.5 points) If the sting message were to be efficiently and quickly verified, suggest a scheme to improve honeybee's efficiency. *One-time signatures*

Problem 4: Probabilistic counting (15 points)

A router connected to an organization wants to count the flows from individual hosts. The router generates a secret random key K. It also maintains the flow ID for each IP address of a host in the organization. It computes $H(K||SrcIP||DestIP)$ which outputs a value uniformly between 0 and 1. Use these hash outputs, to answer the questions below.

1) (5 points) How does one use this scheme in a router to detect a malicious host (with maximum number of flows)? *Router stores the smallest hash value v and computes the number of flows as $\frac{1}{v}$*

2) (5 points) Does this counting method lead to accurately framing a host as being malicious all the time? Explain in 1-2 sentences why or why not? *No, an attacker controlling one host could bias the estimation, leading to false positives*

3) (5 points) Suggest and explain a simple fix to make this counting scheme more robust and also show how to estimate the flows with your suggestion. *Instead of picking the smallest hash value, pick the k^{th} smallest hash output and the estimated number of flows will be $\frac{k}{v}$*

Problem 5: TCP Ack storm DoS attack (25 Points)

In the lectures, we discussed several TCP/IP attacks (e.g., TCP SYN flooding, TCP hijacking and TCP poisoning). TCP Ack storm DoS attack is another TCP/IP attack that exploits a subtle design flaw in the TCP specification. This attack is presented and evaluated by Abramov and Herzberg (2012). In the TCP specification, upon receiving a packet with the acknowledgement number field that is larger than the one that the receiving client sent, the receiving client drops the packet and resends the last sent acknowledged packet to the other side of the TCP connection. Figure 1 illustrates one scenario of this attack.

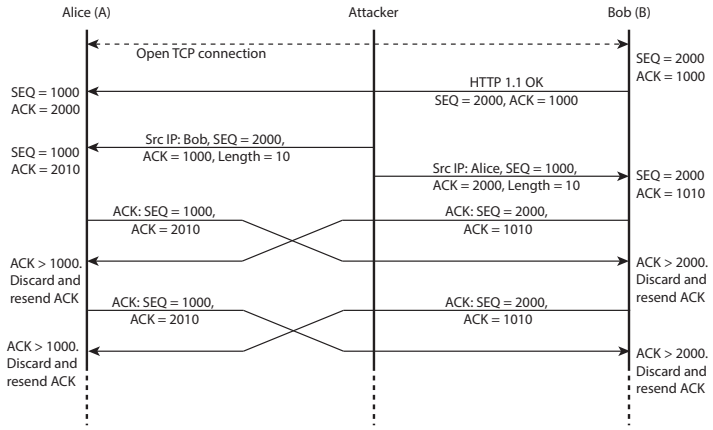

Figure 1: The Two-Packets TCP Ack-storm attack.

In Figure 1, the initial value of A's sequence number is 1000 (A.SEQ = 1000) and B's sequence number is 2000 (B.SEQ = 2000). The attack is conducted as follows:

1. Attacker sends A and B packets of length 10, each impersonating the opposite side.
2. Upon receiving the packet, A advances A.ACK to be 2010, and sends an Ack to B. B advances B.ACK to be 1010 and sends an Ack to A.
3. When B receives a packet with A.ACK = 2010, when B.SEQ = 2000, B discards the packet and resends A the Ack in which B.ACK = 1010 $>$ A.SEQ. A does the same, as it receives a packet from B with B.ACK = 1010.
4. Both A and B receive packets with the Ack number bigger than their SEQ. The behavior in step 3 is performed again.
5. The loop continues when both parties keep receiving packets with an Ack larger their sequence numbers, until when both packets are dropped, or when one side reaches a timeout and ends the connection by RST packet (the time out in apache servers is 225 s).

1) (5 points) The two-packets Ack storm attack described in Figure 1 causes a limited number of packets sent between A and B, and consumes a limited amount of network resources to the target. Please provide a way to enable the attacker to increase the bandwidth consumption in a connection session between A and B.

The attaker can inject additional acknowledgment packets into the TCP streem, identical to the ones sent back and forth by A and B.

2) (5 points) Can we use this attack to attack the TCP-based BGP session between two BGP routers? Please justify your answer.

No. It is because the attacker cannot eavedrop packets between two BGP routers, so the attacker has to guess the sequence number to successfully launch the attack.

3) (5 points) Can SSL prevent this attack? Please justify your answer.

No. SSL is above TCP.

4) (5 points) A TA plans to install a firewall to filter the packets caused by this attack. How to design the firewall?

The firewall filters TCP duplicate Acks.

5) (5 points) Is this attack practical? If it is not, please justify your answer. If it is practical, please discuss the impact of this attack (e.g., the severity of this attack).

Yes. Ack storm DoS attacks are practical. In fact, they are easy to deploy in large scale, especially considering the widespread availability of open wireless networks, allowing an attacker easy WitM abilities to thousands of connections. Storm attacks can be launched against the access network, e.g. blocking address to proxy web server, against web sites, or against the Internet backbone

Problem 6: TPM and Privacy CA (10 Points)

In the class, we discussed the Trusted Platform Module (TPM) and the attestation protocol. For the attestation purpose, each TPM holds an asymmetric key pair called Endorsement Key (K_{EK},K_{EK}^{-1}), which is generated and certified at the TPM manufacturer. The K_{EK}^{-1} never leaves the TPM chip. Because K_{EK} is unique to each TPM, if the K_{EK} is used in attestation directly, the uniqueness enables a third party to identify a particular TPM, raising privacy issues for the platform owner. To achieve platform attestation while preserving privacy, a privacy certificate authority solution (PCAS) is designed. In PCAS, the TPM generates a short-term asymmetric key pair called Attestation Identity Key (K_{AIK},K_{AIK}^{-1}) for attestation. A privacy CA verifies the K_{AIK} provided by a TPM and issues a certificate on the K_{AIK} as long as the TPM is in possession of a valid K_{EK}. However the certificate provided by the privacy CA only provides the statement that the K_{AIK} belongs to a genuine TPM, but does not reveal which TPM. The K_{AIK},K_{AIK}^{-1} pair is then used for attestation. The privacy CA records a link between the K_{EK} and the K_{AIK}. A TA plans to build a privacy CA server at CMU and designs a protocol. The privacy CA server issues a valid certificate on the K_{AIK} provided by a TPM. The protocol is shown in Figure 2.

TPM		$Host$		$PrivacyCA$

K_{EK}^{-1}
Load (K_{AIK}, K_{AIK}^{-1})
	$\xleftarrow{K_{AIK}}$	K_{EK}, K_{CA}	$\xrightarrow{K_{AIK}, K_{EK}}$	K_{EK}, K_{CA}^{-1}
				if(K_{EK} is not valid)
				return reject
				else
				generate a nonce c

$c\|\|K_{AIK} = \{a\}_{K_{EK}^{-1}}$
if(K_{AIK} is not known)
return abort
else
release c
	$\xleftarrow{a}$		$\xleftarrow{a}$	a= $\{c\|\|K_{AIK}\}_{K_{EK}}$

	$\xrightarrow{c}$		$\xrightarrow{c}$	if (c is not valid)
				return reject
				else
				generate a nonce k
				$d = \{k\|\|K_{AIK}\}_{K_{EK}}$
				$cer = \{K_{AIK}\}_{K_{CA}^{-1}}$
				$record(K_{EK}, K_{AIK}, cer)$

$k\|\|K_{AIK} = \{d\}_{K_{EK}^{-1}}$
if(K_{AIK} is not known)
return abort
else
release k
	$\xleftarrow{d}$		$\xleftarrow{b, d}$	$b = \{cer\}_k$

	$\xrightarrow{k}$	$cer = \{b\}_k$		

Figure 2: Privacy CA protocol.

1) (4 points) Assuming the TPM is not compromised, is the protocol secure? Please justify your answer. If it is not secure, please describe the attack (please note that the TPM EK cannot be used for signature).

The proposal is secure. TPM verifies if the AIK is a legitimate AIK.

1) (6 points) The TA learned that the TPM may be compromised by the attacker. The TA thinks that when the TPM is compromised, the protocol is not secure any more because a compromised TPM may claim it has an AIK that belongs to another TPM and obtain the certificate on the AIK from the privacy CA. Is the TA correct? If the TA is incorrect, please justify your answer. If the TA is correct, please briefly describe why this attack is possible, and assuming that the privacy CA does not know if the TPM is compromised, please help the TA to improve the protocol to prevent this attack.

TA is correct. Solutions to fix it: (1) privacy CA requires the TPM to sign a nonce using K_{AIK}^{-1}; (2) TPM signs K_{EK} using K_{AIK}^{-1}.

Problem 7: Secure Broadcast Communication (20 Points)

A number of schemes for Broadcast distribution have been covered. Not satisfied with either of them, Harry Q. Bovik comes up with a scheme of his own, called Time-Structure-Tree (TST), described as follows.

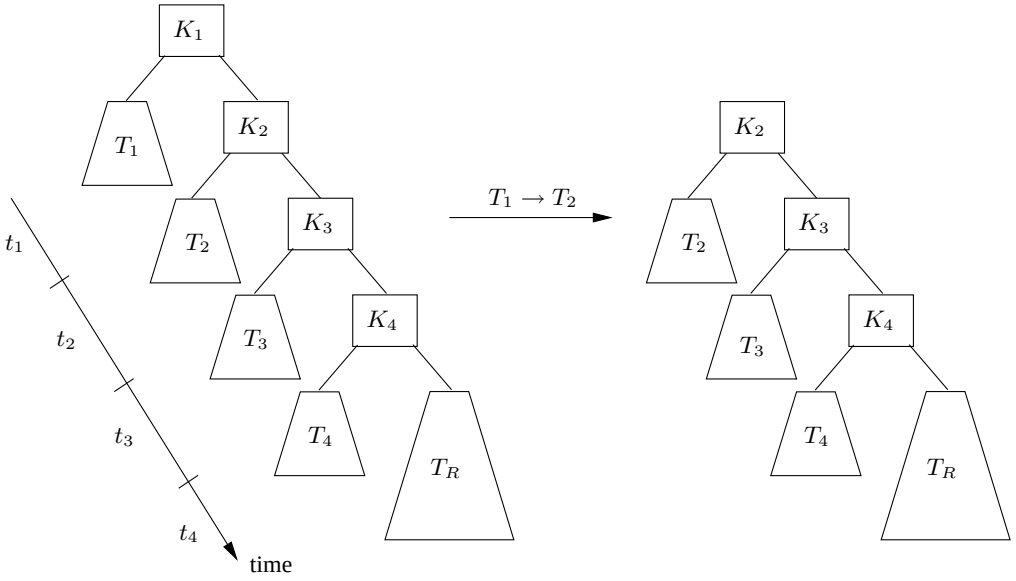

Figure 3: This figure shows the evolution of the time-structured tree (TST) protocol.

The goal of the TST protocol is to extend the LKH key tree protocol to provide the property that when a member joins for a pre-determined time period, no key update message needs to be broadcast when that member leaves.

Figure 3 depicts the TST protocol. Each subtree T_1, T_2, T_3, T_4 and T_R stands for a regular LKH key tree. Each subtree T_i corresponds to a time interval t_i. As time progresses, the key server removes the subtree T_i and current root node K_i after time period t_i is over. With this mechanism, all the members in subtree T_i automatically leave the group at the end of time period t_i, without requiring any broadcast message. The figure shows the change of the key tree as time passes from period t_1 to period t_2.

For example, if a member Alice only joined for a single time period, she would be placed in the LKH tree within subtree T_1. Member Bob wants to stay until the end of time period 2, so he would be placed in subtree T_2. During that time period t_1, key K_1 would be used to encrypt the broadcast messages. At the end of time period 1, the key server will simply cut off subtree T_1 and key K_2 will become the new root key, as shown in the diagram on the right in Figure 3.

a) (2 points) Assuming each subtree T_i contains 1024 members, how many keys will Alice receive from the key server if she joins during time period T_1 and wants to stay only until the end of time period T_1?

11 keys, full credit is given for 10 and 12 as well.

b) (2 points) Assuming each subtree T_i contains 1024 members, how many keys will Bob receive from the key server if he joins during time period T_1 and wants to stay until the end of time period T_2?

12 keys, full credit is given if the number is one higher than the response in a).

c) (2 points) Assuming each subtree T_i contains 1024 members, more generally, how many keys will Carol receive from the key server if she joins during time period T_1 and wants to stay until the end of time period T_i?

10+i

d) (3 points) Is this scheme scalable with respect to the number of members in the group? Briefly justify your answer.

Yes, the heights of the subtrees is still logarithmic in the number of member in the tree.

e) (3 points) Is this scheme scalable with respect to the duration of time a member joins the group? Briefly justify your answer.

No, the number of keys received is linear in the duration of the membership.

f) (4 points) A scheme with a similar property we've seen in class is MARKS. Recall that MARKS did not support member eviction. Can the TST scheme support member eviction? Briefly argue why or why not.

Yes, eviction is supported because TST is equivalent to LKH.

g) (4 points) Is the scheme secure (i.e., does it offer forward and backward secrecy)? Briefly argue why or why not.

Yes, because TST is essentially equivalent to LKH, and LKH offers secure forward and backward secrecy.