1. (10 points) Assume that the following packets are all received by a receiver using EMSS.

What should the receiver do to authenticate packet $P_i$? Please don’t include unnecessary steps that do not contribute to the authentication of $P_i$.

Solution:
Simply put, he has to see if there is a directed path from packet $P_i$ to the signature packet. To elaborate,
- Verify the signature packet and get the correct $H(P_{i+1})$ and $H(P_{i+2})$.
- The receiver should authenticate the packet $P_{i+2}$ by calculating its hash and checking whether it equals the hash value received in the signature packet. If it matches, then the user gets the correct $H(P_i)$ from packet $P_{i+2}$.
- Now the receiver can authenticate the packet $P_i$ by calculating the hash of packet $P_i$ and checking whether it equals the hash value received in packet $P_{i+2}$. If they are equal, the packet is authenticated.
- The receiver can authenticate packet $P_i$ even if packet $P_{i+2}$ is lost as it can get the $H(P_i)$ from packet $P_{i+1}$ if $P_{i+1}$ is authenticated from the signature packet in the similar fashion as $P_{i+2}$ was authenticated. In other words, there are two directed paths from the packet that needs to be authenticated to the signature packet.
2. (10 points) The following figure illustrates the idea of the immediate authentication mechanism in the extension to TESLA.

Is it correct to say that once the packet for $M_{j+vd}$ is received, the receiver can immediately authenticate the entire packet? Why?

Solution:

No. Though the data portion of the later packet is authenticated by the hash image in the earlier packet and the disclosed key $K_i$ is authenticated by the earlier key, the other parts in this packet are not authenticated. These include the hash image $H(M_{j+2vd})$ and the MAC. Immediate authentication is achieved.

3. (10 points) TESLA is based on a one-way key chain of limited size. The key chain commitment is initially authenticated using a digital signature. When one key chain is completely used up, the sender will need to generate and use another key chain. For the later key chain, can you use keys in the previous key chain to authenticate its key chain commitments?

(a) (5 points) Describe how exactly this should be done.

Solution:

The sender uses the last few keys in the previous chain to authenticate the commitment of the later chain. The sender includes the new commitment into the last packet. Once it broadcasts the packet, receivers can learn the commitment and authenticate the later packets.

(b) (5 points) Discuss the pros and cons of this approach.

Solution:

Pros: this scheme can reduce expensive digital signature operation.
Cons: The problem is packet losses. If those packets are lost, the later key chain commitment cannot be authenticated.

4. (10 points) TESLA requires loose clock synchronization. In applications where such synchronization is not possible, TESLA cannot be used. Assume that we require a solution that is
not vulnerable to compromised receivers for such applications. Further assume that the sender and each receiver share a pairwise symmetric key.

(a) (5 points) Describe a method for broadcast authentication that only uses symmetric cryptography.

Solution:
The sender should generate MACs per receiver using each symmetric key. The packet has $H(m|k)$ Each packet is related to a different pairwise key of each receiver. The sender broadcasts these MACs along with the packet.

(b) (5 points) Discuss the limitation of your solution.

Solution:
It has a scalability problem. This is because every packet must contain a separate MAC for each receiver. In addition, the sender has a large computation overhead, since it must compute separate MACs for every receiver on every packet it sends.

5. (10 points) Compare the use of hash functions in BiBa signature scheme, BiBa broadcast authentication protocol, MicroMint, and TESLA. Consider both similarities and differences. (Note: You can ignore MicroMint in your answer, since we didn’t cover it in class this semester.)

Solution:
The similarity is that all these schemes make use of the one-way property of hash functions. The differences are: BiBa signature scheme generates random SEALs and look for a 2-way collision of SEALs to form a signature. The security lies in the difficulty of finding k-way collisions for one-way functions. BiBa broadcast authentication protocol combines the use of SEAL hash chains and k-way collisions. Micromint exploits the strong collision free property. By randomly selecting $x$ and computing $h(x) k^2 n$ times to get about $2^n - 1$ k-way collisions, where $n$ is the number of bits in $h(x)$. And TESLA uses key hash chains for authentication.

6. (10 points) Consider group key agreement protocols.

(a) (5 points) Describe the general form of the group key in the GDH protocols.

Solution:
The general form of group key in GDH protocol is $K = a^{N_1 N_2 ... N_n} \mod q$, where $n$ is the number of group members, $a$ is the base of exponent, and $q$ is the order of the group.

(b) (5 points) Describe the general form of the group key in the tree-based group key agreement protocol.

Solution:
The general form of group key for a tree-based group key agreement is \( g^{n_1\cdot n_2} \), where \( n_1 \) (or \( n_2 \)) is either the private key associated with a leaf node, or a key in the form of \( g^{n_1\cdot n_2} \).

7. (10 points) In GDH.1, for the up flow, each member \( M_i, i<n \), receives some intermediate values and forwards another set of values to \( M_{i+1} \). Count the number of values in the message received by \( M_i \). List the set of values received by \( M_6 \).

Solution: [The notations are the same with that used in the paper]

In GDH.1, the message received by \( M_i \) from \( M_{i-1} \) has the following values:
\[ a, a^{N_1}, a^{N_1 N_2}, a^{N_1 N_2 N_3}, \ldots, a^{N_1 N_3 \ldots N_{i-1}} \], the last one being the cardinal value. It receives a total of \((i-1)\) values.

8. (10 points) Consider GDH.2. Denote the members as \( M_i \), and the secret of \( M_i \) as \( N_i \). Assume there are totally 7 members in a group.

(a) (5 points) What is the set of messages \( M_5 \) receives in the upflow stage?

Solution:

In the upflow stage, \( M_5 \) receives \( \{a^{N_1 N_2 N_3 N_4}, a^{N_1 N_2 N_3}, a^{N_1 N_2 N_4}, a^{N_1 N_3 N_4}, a^{N_2 N_3 N_4}\} \), with \( a^{N_1 N_2 N_3 N_4} \) being the cardinal value.

(b) (5 points) What is the message sent to \( M_5 \) in the downflow stage? What is needed by \( M_5 \) in this message?

Solution:

In the downflow stage, the message sent to \( M_5 \) is \( \alpha \) \( \{a^{N_2 N_3 N_4 N_5 N_6 N_7}, a^{N_1 N_3 N_4 N_5 N_6 N_7}, a^{N_1 N_2 N_3 N_4 N_5 N_7}, a^{N_1 N_2 N_3 N_4 N_6 N_7}, a^{N_1 N_2 N_3 N_4 N_6 N_5} \} \). \( M_5 \) will need \( \alpha \) to raise \( N_5 \) in order to get the group key \( \alpha^{N_1 N_2 N_3 N_4 N_5 N_6 N_7} \).

9. (10 points) Consider the Tree-Based Group Key Agreement protocol and the following key tree:
Assume the prime defining the finite field is $p = 11$, and the generator is $g = 2$. Further assume the private key of $M_1$ through $M_6$ are $3, 4, 5, 6, 7, \text{ and } 8$. Compute the blind keys along the path from $M_1$ to the root.

**Solution:**

For node $<3,0>$: $K_{<3,0>} = 3$, 
so $BK_{<3,0>} = g^{K_{<3,0>}} \mod p = 2^3 \mod 11 = 8$ 

For node $<2,0>$: $K_{<3,1>} = 4$, $K_{<2,0>} = g^{K_{<3,0>}K_{<3,1>}} \mod p = 2^{3 \times 4} \mod 11 = 4$, 
so $BK_{<2,0>} = g^{K_{<2,0>}} \mod p = 2^4 \mod 11 = 5$ 

For node $<1,0>$: $K_{<2,1>} = 5$, $K_{<1,0>} = g^{K_{<2,0>}K_{<2,1>}} \mod p = 2^{5 \times 4} \mod 11 = 1$, 
So $BK_{<1,0>} = g^{K_{<1,0>}} \mod p = 2^1 \mod 11 = 2$ 

For node $<0,0>$: $K_{<3,6>} = 7$, $K_{<3,7>} = 8$, $K_{<2,3>} = g^{K_{<3,6>}K_{<3,7>}} \mod p = 2^{7 \times 8} \mod 11 = 9$,  
$K_{<2,2>} = 6$, $K_{<1,1>} = g^{K_{<2,2>}K_{<2,3>}} \mod p = 2^{6 \times 9} \mod 11 = 5$,  
$K_{<0,0>} = g^{K_{<1,0>}K_{<1,1>}} \mod p = 2^{1 \times 5} \mod 11 = 10$,  
So $BK_{<0,0>} = g^{K_{<0,0>}} \mod p = 2^{10} \mod 11 = 1$. 

10. (10 points) Consider the following network configuration, in which Iolus is used.
Assume the GSC is distributing a new group key to the group members using Iolus. How many times will this new key be encrypted and decrypted before A learns the value of the new key? Also describe what entity performs each of the encryption and decryption.

Solution:

GSC encrypts the group key with the subgroup key of G1,
GSI (1-2A) decrypts the message and reencrypts it with the subgroup key of G2A,
GSI (2A-3B) decrypts the message and reencrypts it with the subgroup key of G3B,
A decrypts the message.
Hence 3 encryptions, 3 decryptions.
11. (10 points) Consider the basic LKH.

If \( R_3 \) is removed from the group, what keys must be changed? Give one way to distribute the updated keys to the other group members. (Please note that it’s \( R_3 \), not \( R_5 \), that is removed.)

Solution:

If \( R_3 \) is removed, the keys that need to be changed are: \( K_{34}, K_{14}, K_{18} \). Let the new key be, from \( K_{34} \) to \( K'_{34} \), from \( K_{14} \) to \( K'_{14} \) and from \( K_{18} \) to \( K'_{18} \). If we assume user-oriented key management, messages sent by the Group Key Manager (denoted by \( s \)),

\[
\begin{align*}
s & \rightarrow \{ R_4 \}: (K'_{34}, K'_{14}, K'_{18})_{K4} \\
\rightarrow \{ R_1, R_2 \}: (K'_{14}, K'_{18})_{K12} \\
\rightarrow \{ R_5, R_6, R_7, R_8 \}: (K'_{18})_{K58}
\end{align*}
\]
12. (15 points) You need to read the following paper to answer this question:

Chung Kei Wong, Mohamed Gouda, Simon S. Lam, "Secure group communications using key graphs."

Consider the following key tree.

(a) (5 points) If u_5 is removed from the group, what keys should be changed?

(b) (5 points) Assume key oriented rekeying is used. Describe the messages the group manager needs to send to the group members. Use the following convention to describe each message:

GM \rightarrow \{set of users\}: \{Kx\}Ky, \{Kz\}Kw, ...

(c) (5 points) Assume group oriented rekeying. Redo (b).

Solution:

(a) If u_5 is removed from the group, K_{456}, K_{1-9}, K_{1-c} should be changed.

(b) Let the new key be from K_{456} to K_{46}, from K_{1-9} to K^{'}_{1-9} and from K_{1-c} to K^{'}_{1-c}.

GM \rightarrow \{\{u_1, u_2, u_3\}: \{K^{'}_{1-c}\}K^{'}_{1-9}, \{K^{'}_{1-9}\}K_{123}\}
GM \rightarrow \{u_4\}: \{K^{'}_{1-c}\}K^{'}_{1-9}, \{K^{'}_{1-9}\}K_{46}, \{K_{46}\}K_{4}
GM \rightarrow \{u_6\}: \{K^{'}_{1-c}\}K^{'}_{1-9}, \{K^{'}_{1-9}\}K_{46}, \{K_{46}\}K_{6}

(c) Assume group oriented rekeying. Redo (b).
GM $\rightarrow$ \{u_7, u_8, u_9\}: \{K'_1\}_cK'_{1-9}, \{K'_1\}_{9}\}
GM $\rightarrow$ \{u_a, u_b, u_c\}: \{K'_1\}_cK_{abc}

(c) If we redo for group-oriented rekeying $K_{456}$, $K_{1-9}$, $K_{1-c}$ should be changed. Assume the new keys are $K'_456$, $K'_1\_9$, $K'_1\_c$:

$L_0$ : \{K'_1\}_cK'_{1-9}, \{K'_1\}_cK_{abc}
$L_1$ : \{K'_1\}_9K_{123}, \{K'_1\}_9K'_456, \{K'_1\}_9K_{789}
$L_2$ : \{K'_456\}_4K_4, \{K'_456\}_6K_6

GM $\rightarrow$ \{u_1, u_2, ..., u_9, u_a, u_b, u_c\}: L_0, L_1, L_2