10 points for each question

Question 1 (Problem 1 on page 438) Suppose Alice is sending packets to Bob using IPSec. Suppose Bob's TCP acknowledgement gets lost, and Alice's TCP, assuming the packet was lost, retransmits the packet. Will Bob's IPSec implementation notice that the packet is a duplicate and discard it.

Question 2 (Problem 7 on page 439) Referring to Figure 17-2, assume that A and B are using IPsec in transport mode, and F1 and F2 have established an encrypted tunnel using IPSec. Assume A sends a TCP packet to B. Show the relevant fields of the IP header(s) as given to A's IPsec layer, as transmitted by A, as transmitted by F1, and as received by B.

Question 3 (Problem 1 on page 476) Suppose if Alice's aggressive-mode IKE connection initiate is refused, Alice starts up another aggressive-mode connection initiate with her next (and weaker) choice of Diffie-Hellman group, rather than starting a main-mode exchange telling Bob all her supported Diffie-Hellman groups. What is the vulnerability, given an active attacker? (See 18.5.1 *Aggressive Mode and Main Mode*)

Question 4 (Problem 3 on page 476) Show how someone who knows both Alice's and Bob's public encryption keys (and neither side's private key) can construct an entire IKE exchange based on public encryption keys that appears to be between Alice and Bob.

Question 5 (Problem 4 on page 476) Write out the shortened version of main-mode public signature keys that hides Alice's ID from an active attacker. Explain why the 6-message version descried in 18.5.10.1 *Public Signature Keys*, Main Mode allows parallel computation of g^{ab} mod p, whereas the shortened version does not.

Question 6 (Problem 7 on page 476) Design a protocol in which one side has a public signature key and the other side has a public encryption key.

Question 7 (Problem 8 on page 476) Design a protocol in which authentication is one-way since only one side has a public key. Do the protocol with a public signature key. Now do it with a public encryption key.

Question 8 (Problem 9 on page 476) In the public encryption key case, SKEYID is defined as hash(nonces, cookies). SKEYID is supposed to be something that is not computable except by Alice and Bob. Why can't an eavesdropper or active attacker calculate SKEYID?