Josh Gribbon
Homework 01
20171006

Problem 1: Shared of forgotten keys?

1. Confidentiality would of course be important because they don't want to leak the key while trying to verify that they match. Integrity would also be important so they can know that their conclusion is valid.
2. **Integrity** is met for an attacker that can eavesdrop, but if the attacker can modify the message then of course the message integrity can't be ensured. **Confidentiality** is not ensured, because an attacker can eavesdrop to get x and y, then xor the two to get the key that Alice and Bob share, if it is indeed the same key.
3. A hash function would help. Alice and Bob could then use their keys to generate the hash, and just send the hashes to each other. As long as the hash is collision resistant then the attacker will not be able to figure out the key. A MAC could then also be used to ensure integrity.

Problem 2: Mind the security definitions

1. MACs provide integrity and "unforgeablity"
2. A replay attack involves an attacker taking a verifiable message and sending it repeatedly in order to mess up the stream.
3. Replay attacks don't break message integrity, because the individual messages sent remain valid. The goal is to mess up the system, not the encryption.

Problem 3: Failing cryptography

1. Eve can use the encrypted messages she had has and compare to what topics where actually covered. Then she can listen to the communication for the next exam and match up the beginnings of the encrypted messages (excluding the MAC) to her previously collected messages (also excluding the MAC) and find out which topic will be on what problem. This also assumes that Bob always asks for problems in order. If they changed the order that they discussed problems then Eve could find patterns over time but would not necessarily be able to figure out all the problems for the next assignment.
2. Eve could do a replay attack and send Alice's previous message to Bob, who would then think there will be more assignments after number two.
3. This is a bad idea because then if one key is leaked they both are. If their encryption key is leaked they can still use their MAC key to establish a new encryption key. If their MAC key is leaked then their encryption is still secure but they can't guarantee integrity or authenticity.

Problem 4: Cloudy solutions

1. Parts:
    1. Data deduplication
    2. This saves storage space by making sure they don't store the same data twice.
    3. The hash function must be collision resistant to reliably confirm or deny that a file has already been uploaded
    4. If files are encrypted then the same file will appear differently for different users, causing the resulting hashes to be different, making it impossible to check if they're already there.
2. Eve could choose a student that she thinks would have the correct answers and try uploading different files with their name and different answer combinations. Then if she sees that a file will not be uploaded she will know the combination submitted by that student. If she could do this efficiently she could try it for all students and find out what they all submitted. With n true/false questions there will be 2^n possible combinations. For 10 questions there are 1024 combinations, but with 100 questions there would be 1.26x10^30 combinations, making this attack unfeasible.