**USC UPSTATE**

**CSCI 455: Computer Security**

**Spring 2019**

**Homework Assignment 5**

**Problem 1**

In class we described the *hash-then-sign* method for signing arbitrarily long messages, given a block signature scheme for a fixed length and a hash function that maps an arbitrarily long message to a hash value of that fixed length. Explain why the hash function must be *collision-resistant*. That is, if the hash function is not collision resistant, how can an attacker possibly achieve a successful forgery?

Answer: The hash function must be collision-resistant so that no two arbitrarily long messages can be mapped to the exact same hash. The attacker could then find a message that maps to the exact same hash as the original message, making it seem as it’s truly the original.

**Problem 2**

1. Suppose that an *active* attacker intercepts a ciphertext that is produced by a one-time pad. Show how the attacker can modify the ciphertext so that decryption of the modified ciphertext on the recipient’s side yields a message that flips the last bit and preserves all the other bits of the original message. Explain why the attack works. (**Note**: This question shows that encryption does *not* provide integrity; in particular one-time pad encryption and any stream cipher does *not* provide data integrity.)

Answer: An attacker can XOR the last bit of the ciphertext. This works bc the one-time pad just changes the bits based on the number of the current place. So XORing it would just change it again, but when decrypted the messaged is changed.

1. Describe how to prevent the above attack and explain why your method works.

Answer: One can sign the file after encryption. Doing this will allow the recipient the ability to check and see if the file has been altered in anyway once the sender has sent it.