Solutions

ENEE 457

One-Time Pad Class Exercise

1. What happens if we use the same one-time pad to encrypt two messages? What information will be leaked about the messages?

   Solution: Formally, the adversary gets to see c_1 = k \xor m_1 and c_2 = k \xor m_2.

   We discussed two vulnerabilities. The first vulnerability is any issue for any perfectly secret scheme in which the key is used twice. The second vulnerability is specific to OTP.

   1. The adversary can detect whether m_1 = m_2 by checking whether c_1 = c_2. If c_1 = c_2 then it must be the case that m_1 = m_2. This reveals whether the same message was encrypted twice.
   2. The adversary can compute c_1 \xor c_2 = m_1 \xor m_2 \xor k \xor k = m_1 \xor m_2. Thus, the adversary learns the \xor of the two encrypted messages. Practically speaking, once the adversary has this information, the adversary can use frequency analysis (e.g. if it knows that the messages are Englinsh text) to potentially fully recover m_1 and m_2.

2. In a brute-force search, an attacker tries decrypting a ciphertext with each possible key until the correct key is discovered. Does a brute-force search attack work for the one-time pad? Why or why not?

Solution: Brute force search *does not* help in the case of one-time pad (or any perfectly secret encryption scheme). The reason is that if you fix a ciphertext c, it maps to *every message* m under some key. Specifically, if we want c to decrypt to m, we can set k = c \xor m. This means that if one performed brute force search on a particular ciphertext c, one would recover every possible message in the message space. Even if the adversary knows some partial information about the message (e.g. knows that the message is English text, or more generally falls into some set M), it still doesn't help. Because during the brute force search every message in M will be recovered under some key and so no additional messages from M can be eliminated during the brute force search.