Crypto Homework 1: Blocks and Streams
Due 02.28.2024
Samantha Pope
CS6014

<u>Question 1: A block cipher with an 8-bit block size is very easy to break with a known-plaintext attack (assuming each block is just encrypted independently with the same key). Describe how you would do so.</u>

First, I would find out as many pairs as I could using the plaintext document and the ciphertext. Since the block size is 8 bits, each block can be one of 256 different values. I would then create a substitution map for my pairs that says which plaintext value matches with the cipher value. Then I would analyze the pattern of how the plaintext is transformed into the cipher texts. I would use some pattern-recognition - like looking at how bits or groups of bits are changed between the ciphertext and plaintext. Then I would use my substitution map to decrypt known ciphertext to make sure we are decrypting correctly. Then this could be used to decrypt ciphertexts that use the same key, since I have the “formula” or key now deciphered.

<u>Question 2: Assume you're sending a long message using a block cipher (like AES) with the following scheme: split the message into block-sized chunks, then encrypt each with the same key. Basically Alice sends Bob AES(m1, k), AES(m2, k), AES(m3, k), etc.</u>

<u>Part A (3 points): Even if they can't decrypt blocks, what information can an eavesdropper discern from this scheme? Hint: Imagine that Alice is sending a table of data where each cell is exactly one block of data.</u>

They can discern how much data is being sent across between the two parties by the number of blocks sent. They can also gauge the order that the data should be organized into, if they are sending blocks of information serially. They can also look at repeating blocks (since they are using the same key, likely data will have repeats) which could help the eavesdropper understand the order of the data and more about how it is constructed.

<u>Part B (4 points): Things are actually even worse! A malicious attacker can actually CHANGE the message that Bob receives from Alice (slightly). How? This is particularly bad if the attacker knows the structure of the data being sent (like in part A).</u>

The attacker could capture an encrypted block and send it multiple times. They could also reorder the blocks and how they are sent so it messes up the message between the two parties. If they learn the format of how the blocks are organized, for example the key always sends 111 to signal that is the last block being sent, the attacker could send that “ending message” and terminate the conversation early.

Part C (3 points): How could you modify the scheme to mitigate/prevent these types of attack?

I would make a formula so that we could send the messages out of order. This would make it harder for the attacker to find patterns in what we sent. This would work for some information sharing, but if future commands being sent are reliant on previous commands, it may prove difficult.

I could also use authenticated encryption (like AES) to ensure that the attacker is not as easily able to send things without us detecting it was sent by a malicious source.

I could also make the blocks chained (using something like Cipher Block Chaining or Counter) where the encryption of the block depends on the one sent before it. This makes it harder for the attacker to alter the messages or scramble them, as they are now linked.

Question 3 and 4:

My code for both Question 3 and 4 is in the nested folder called “BadBlockCipher/Rc4Cipher”