Aiden Pratt
6014 Cryptography HW

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.
In a known-plaintext attack, the attacker has access to both the plaintext and its corresponding ciphertext, allowing them to analyze the relationship between the two and potentially discover the encryption key. When dealing with a block cipher with an 8-bit block size, the situation becomes more vulnerable to attacks due to the limited number of possible block combinations.

To break the encryption with an 8-bit block size using a known-plaintext attack, an attacker might employ a technique called brute-force, taking advantage of the small block size. Since there are only 256 possible block combinations (2^8), an attacker could systematically try each possible key until finding the one that correctly encrypts the known plaintext to the provided ciphertext.

If the same key is used for each block independently, the attacker can leverage the repetition of the key across blocks. They can analyze the patterns in the ciphertext that correspond to the repeated occurrences of the known plaintext, providing additional information to deduce the key.

In the case of expecting a specific word like "BOB" in the plaintext, the attacker can focus on the blocks where this word is likely to appear. By encrypting "BOB" with each potential key and comparing the result to the corresponding ciphertext blocks, they can narrow down the possibilities and increase the efficiency of the attack.

It's important to note that the vulnerability of an 8-bit block cipher to known-plaintext attacks highlights the need for stronger encryption algorithms with larger block sizes in real-world applications to ensure robust security.

Aiden Pratt
6014 Cryptography HW

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.

- **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.**
  The eavesdropper could detect patterns in each block such as the starting and ending of each block. For example, if the same key is being used for each block, if the a value in the beginning position in the first block is the same as the value in the beginning position in the second block, an eavesdropper can find some statistical analysis to uncover what the value of the plain text or key might be.

plaintext:	1101 1010
key:	1100 1100
ciphertext:	0001 0110

  in the first position in both blocks, it is a 0. Because the ciphertext has this repeating 0, then that means the plaintext in both blocks in that first position is a repeating bit.

- **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).**

The attacker could potentially exploit the predictable nature of encrypting each block with the same key in the scheme. The attacker can be aware that it is a block cypher and that the same key is being used for each block. They can manipulate a ciphertext of a block which will correspond to the different plaintext values.

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

A modification to the key after each block is necessary so that a unique key is used in each block. When someone is using a unique key for each block, the encryption becomes far more robust and unpredictable, making attacks more challenging.

Aiden Pratt
6014 Cryptography HW

programming part 1:

Try modifying 1 bit of the ciphertext and then decrypting with the correct passwords. What do you see?

encrypted state of input message: 00000011 11100010 00001010 00111111 10000010 11110111 01010110 10000000

one bit changed cyphertext: 00010011 11100010 00001010 00111111 10000010 11110111 01010110 10000000

flipped the fourth bit and that is seen in the binary print out of the cypher text.

programming part 2:
Verify that encrypting 2 messages using the same keystream is insecure. What do you expect to see if you xor the two encrypted messages?

utilizing the same key for encrypting two different messages is insecure because it violates the one time pad principle. when these encrypted messages are xor'd, it can result in revealing a little bit of details about each of them.