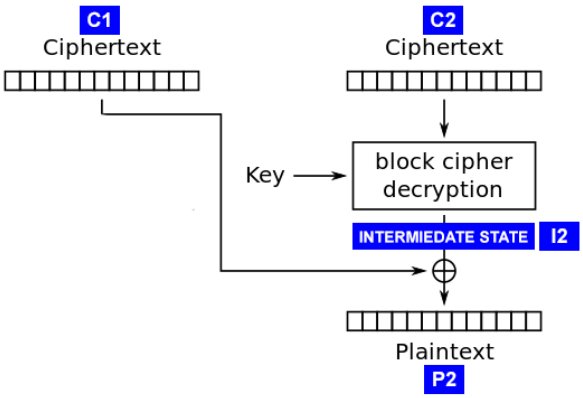
CMPU-315 – Computer Security (S21)

Hannah Gommerstadt

Homework #1 – Due April 12th, 2021

Introduction

In this assignment, you will implement a padding oracle attack. Specifically, you will be decrypting a ciphertext that consists of 3 blocks of 16 characters. The first block is the Initialization Vector (IV) which can be ignored. The second block is the first block of the actual ciphertext (C1). The third block is the second block of the actual ciphertext (C2). Your goal is to decrypt C2. Remember, that due to the IV, you will not be able to decrypt C1. For extra credit, you can extend your attack to work for ciphertexts with any number of blocks. It can be helpful to refer to this diagram:



I have provided an oracle, found in the file `oracle.py`. I have also provided some useful methods in `attack.py`. Your goal is to implement the `attack()` method.

Pseudocode

1. Break ciphertext into blocks.
2. Initialize variables *fakeciphertext* (this is C1') and *plaintext* (this is P2) to be lists of 16 characters.
3. For each character c in the block (16 of them, going backwards):
   (a) For every value v ranging from [0,256):
      i. Set the current character of *fakeciphertext* to v. This is C1'[c].
      ii. Query the oracle by passing it *fakeciphertext* and the block you want to decrypt. This query represents C1'C2.

iii. If the query returns false, continue to next v.

iv. If the query returns true:

A. Compute the intermediate value:
I2[c] = C1'[c] $\oplus$ P2'[c]
Remember that P2'[c] refers to the padding of the fake plaintext. For example, when working on the second to last character, the padding will be 02.

B. Compute the value of the real *plaintext* character:
P2[c] = I2[c] $\oplus$ C1[c].
Note, that C1[c] is the real prior ciphertext block.

C. Now that we have the plaintext character, we need to set everything up for the next iteration. For example, if we just dealt with the second to last character, we need to set both the second to last character and the last character of the *fakeciphertext* to values such that the padding will be 03 (so we can deal with the 3rd to last character).
For every character k from the current character c to the last character:

- Get the value of the real ciphertext. This is C1[k].
- Compute I2[k] = C1[k] $\oplus$ P2[k].
  Note that P2[k] is the real plaintext character that has been decrypted!
- Update *fakeciphertext* by computing C1'[k] = I2[k] $\oplus$ P2'[k].
  Note that P2'[k] is the character in the fake plain text, which should be the expected padding. In our example, this would be 03.

4. Return the *plaintext*.

Hints

- Carefully look at my code in `oracle.py` and `attack.py` - it may give you hints about how to complete the assignment.
- To access the second to last element of a list k, you can do k[-2]. In Python, a string is a list of characters.
- When XOR'ing two values, you want to make sure you are dealing with integers. To convert a charater to an integer, use `ord()`.
- When constructing ciphertexts or plaintexts, you will want to work with characters. Use `chr()` to convert a number to a character.
- Start by decrypting the last character of the plaintext. Once you have that working, work on step C of the pseudocode above.

Submission

Please make your program readable by commenting your code and using good variable names. Once you are ready, submit your project through the course's Moodle page.