CPE-321
Introduction of Computer Security

One-Time-Pad

Objectives

The objectives for this lab assignment are as follows:

- To implement XOR function for encryption.
- To explore the One Time Pad cipher.
- Apply OTP on an image.
- To implement Two-Time Pad cipher.

Programming Tools: Strong recommendation to use Python and Pycrypto (or some other high-level language with decent crypto support).

Tasks

Please read the following tasks carefully, and finish the lab assignment accordingly:

1. Write a function that XOR's two arbitrary-length bit strings together. To check your work, the ASCII string:

   Darlin dont you go

   XOR with the ASCII string:

   and cut your hair!

   Should produce the hex value:

   250f164c0a1b54441601015259071449154e

   Make sure to indicate an error if the two strings have different length.

2. Implement OTP by XOR the bits of a plaintext with the bits of key to produce a ciphertext. In order to maintain "perfect secrecy" the key must be generated perfectly at random, be as long as the message, and never reused it. Write a program that takes as an input a file to be encrypted and generates a random key (using your system's/dev/random device will be sufficient), XOR the key with the plaintext, and writes the resulting ciphertext to a new file. Within the same program, try to XOR the key with the ciphertext and verify that you get the original plaintext back again.
3. Encrypt an image using OTP. Modify your program from Task 2 to encrypt a windows bitmap (BMP) image. Download the two bitmap images (cp-logo.bmp and mustang.bmp) from the class website. Encrypt both files using the OTP, creating two different ciphertexts. It will be helpful if you name your ciphertexts with a .bmp extension. We will visualize the ciphertexts, tricking an image viewer into thinking thy are images again. To do this, you need to do a little pre-processing to make these files viewable.
4. Implement Two-Time Pad. Modify your program from Task 3 to encrypt both bitmaps images under the same key. Perform the same plaintext header preservation and reattachment as before. View the encrypted images. Both images should appear indistinguishable from random. XOR both encrypted images together, preserving and re-appending one of the headers (either will work).

Questions

1. What is OTP? What are assumptions to achieve perfect secrecy?
2. From the results of Task 3, what do you observe? Are you able to derive any useful information about from either of the encrypted images? What are the causes for what you observe?

3. From the results of Task 4, are you able to derive any useful information about the original plaintexts from the resulting image? What are the causes for what you observe?

In Your Report

Please address the following in your report:

1. Describe each task what you did and what you observed;
2. Include all code that you wrote;
3. Please include any explanations of the surprising or interesting observations you made;
4. Write at a level that demonstrates your technical understanding, and do not shorthand ideas under the assumption that the reader already "knows what you mean." Think of writing as if the audience was a smart colleague who may not have taken this class.
5. Make sure to include any images you have generated;
6. Describe what you did in sufficient detail that it can be reproduced;
7. Please do not use screenshots of the VM to show the commands and code you used, but rather paste (or carefully re-type) these into your report from your terminal. Submit your completed write up to PolyLearn in PDF format.

Hints:

1. BMP images have a 54 byte header that tells the image viewer that the file is a legitimate BMP image. You will need to replace the encrypted BMP header in the ciphertext you created with the valid BMP header from the original plaintext. It will be easiest if you do this programmatically (rather than by hand).
2. Consider the following property of XOR as you do this:
   $(A \oplus K) \oplus (B \oplus K) = A \oplus B$