Lab 2: Cipher Breaking

50.020 Security

Hand-out: February 7
Hand-in: February 14, 9pm

1 Objectives

- Write Python code to decrypt ciphertext using a Substitution Cipher and One-Time Pad.
- Write a small TCP client to communicate with remote challenge server API
- Break a substitution cipher using either brute force (good luck!) or frequency analysis
- Manipulate OTP ciphertext to a target plaintext string

2 TCP client

2.1 Overview

- In this excercise, a remote TCP API will provide a set of challenges for you. You are supposed to break the encryption and recover the plaintexts.
- The server is running on `157.230.47.126:1337`
- The API will always provide binary data or a plain ASCII status/error/success message.
- Based on the provided python requests skeleton code, connect to the API, receive the ciphertext, break the cipher, and resubmit the plaintext as "solution". You should get a positive feedback message.

2.2 Note:

- Consider the full range of 256 extended-ASCII values in your decryption implementation. Your decrypted plaintext should consist only of string.printable characters.
- The API will send a mix of ASCII characters (printable and non-printable) and binary. In particular, binary is only sent in the ciphertext part of the OTP challenge.

3 Attack Implementation

3.1 Part I: Frequency Analysis

- Extend the python script to call option 1 "Substitution". Can you find the substition key used using brute force attack? How many different combinations would you have to try?
- Send the solution plaintext a verify it is correct (reach >80% similarity)
- The substitution is operating on all string.printable values (symmetric mapping). The key changes per each API call.
- If you don't think brute forcing will work, try other methods discussed in lecture
- Hint: the source is in English, the letter case does not affect the verification.

3.2 Part II: OTP messages Integrity

- Extend the python script to call option 2 "OTP". You will obtain a ciphertext again, encrypted with an OTP.
- Try simply returning the ciphertext to the API, you should see a message about the decrypted and parsed content of the message.
- Implement a solution to manipulate the ciphertext to change the message decrypted by the API. In particular, use the techniques discussed in class to change the student ID to your own student ID, and the points reported to 4.

4 Hand-in

- Submit one script which does both API calls, and submits the correct solutions. You may hard-code parameters for the substitution cipher analysis if required.
- Make sure to put your username into the header of the submitted file.