1st Project

MC889/MO421 – Introduction to Cryptography
Prof. Diego de Freitas Aranha
2017/1

1 Introduction

This assignment consists in implementing an automated attack against an insecure instance of the One-Time Pad (OTP) cipher, given two challenge ciphertexts using the same key. The attack will be mounted against ciphertexts captured over the network, after an insecure WEP-protected wifi network is successfully breached.

2 Method

The OTP cipher is a stream cipher, where the key material has the same length as the input message, attaining the perfect secrecy property. OTP is extremely hard to use securely in practice, because key material needs to be random and can never be repeated.

For this assignment, we will consider as the alphabet of definition the Latin alphabet and punctuation marks represented in the ASCII encoding. The plaintext messages were written in the English language.

3 Material

To understand how encryption works in this domain, a simple program implementing OTP is provided. The source code of this program is available as `otp.c` and serves as illustration. The program was used to generate the challenge ciphertexts and was compiled as:

```
$ gcc otp.c -o otp
```

The 205-byte key was extracted from the GNU/LINUX `/dev/urandom` pseudo-random number generator as follows:

```
$ dd if=/dev/urandom of=key.txt bs=205 count=1
```

Encryption was performed as follows for the two plaintext messages:

```
$ ./otp plaintext.txt ciphertext.txt key.txt
```

Decryption works in the same way.

4 Access

The challenge ciphertexts `C1` and `C2` are transmitted over an insecure network, “protected” by the terribly weak WEP encryption standard. To capture the ciphertext, you must find a way to join the network `TeiaDoAranha` located in my office. You can use whatever means you find necessary, but please do not attack neighboring access points or devices inside the network.

Hint: The WEP key is somewhat related to current events in Brazil. If you need to inject traffic from a legitimate client to speed up the attack, you can use the MAC address `b8:27:eb:a4:82:f5`, belonging to the Raspberry Pi 3 device transmitting the messages.

5 Objective

The objective is to implement a program capable of recovering the plaintext messages and the encryption key corresponding to the provided challenge ciphertexts. You can use any approach you may find suitable, such as frequency analysis or dictionary-based *cribbing*. Breaking OTP sometimes requires manual intervention. Thus, the program can receive limited manual assistance for directing the automatic method.

The cryptanalisys tool can receive as input the challenge ciphertexts and produce the original plaintext messages and the encryption key. Recovered plaintext should be identical to the original messages.

For verification, the challenge ciphertexts you must capture have the following hash values:

```
$ sha256sum challenge1.txt challenge2.txt
fdb8bb2642cb7c9a1869ab99019e3ee3eaff5160c0cf5c8aec1267742e941eb1  challenge1.txt
44e79e85e1aaa37ba74a4a77a7fb15f2da43bca494f1b6ff3ab3eb493b68c24f  challenge2.txt
```

6 Scoring

The cryptanalysis tool should be submmited in both binary and source code forms by private e-mail until April 17, 2017. Please including compile and usage instructions and annotate your message subject with either `[MC889]` or `[MO421]`. The program should be followed by a short report describing how the ciphertexts were obtained and the results. Graduate students will receive more scrutiny in the evaluation of their reports. A single **bonus point** in the first written test will be assigned to the fastest program able to automatically recover the message in its entirety.