Ira A. Fulton Schools of Engineering
Arizona State University

CSE 543: Information Assurance and Security

Esper cipher

Purpose

The purpose of this assignment is to familiarize you more with Python (or the programming language you choose) and writing a simple decryption function for a custom encryption routine. You will learn how to read encryption code, deriving the corresponding decryption function (which is the inverse function of the encryption function), implementing the decryption function, and applying the decryption function on a given input string.

Objectives

Students will be able to:

- Read the encryption implementation of the Esper cipher.
- Implement an individual decryption function for an Esper cipher.
- Deal with files and bytes programmatically.
- Decrypt a given ciphertext that is encrypted using an Esper cipher and its corresponding key.

Technology Requirements

The reference encryption code is provided as a Python script; however, you may implement your solutions in any programming language that you choose.

Note: The course team will not be able to help you if you choose any language that is not Python, Java, or C#; therefore, to create the best learning experience, Python is strongly recommended.

Project Description

Esper cipher is a custom cipher. Unlike Caesar cipher that works on uppercase English alphabet only, Esper cipher works on *bytes*. You will not find its implementation anywhere else. The only reference you have is `esper.py` that implements the Esper cipher.

For this project, you are provided three files:

1. `ciphertext.txt` - A ciphertext encrypted with an Esper cipher.
2. esperkey.txt` - The key used for the Esper cipher. It is the stdout output when running `esper.py` to encrypt the plaintext.
3. `esper.py` - The reference implementation of the Esper cipher with only the encryption feature.

You will need to write some code so you can plug in the key (as specified in `esperkey.txt`) and decrypt the ciphertext in `ciphertext.txt`. You will only submit the plain text `plaintext.txt`. Do not include the ciphertext as part of your submission.

If your decryption works correctly, you will see multiple meaningful sentences in English as the plaintext. Otherwise please go back and fix your implementation.

Directions

This list includes recommended steps to solve this problem:

1. Read `esper.py` to understand the encryption algorithm.
2. Write down the encryption algorithm on a piece of paper.
3. Invert the encryption algorithm to get the decryption algorithm.
4. Implement the decryption algorithm in the programming language that you like. Python is recommended.
5. Use the implemented decryption algorithm to decrypt the ciphertext in `ciphertext.txt`.

Evaluation

You will get 100 points if your submitted plain text is fully correct. Otherwise, your points will be 0. Partial credit will not be granted for this project.

Submission Directions for Project Deliverables

In the submission space in the course, you will only submit the plain text as a `plaintext.txt` file. The file must be in plaintext. This assignment is autograded, so you will not receive credit if your plain text file is not a plaintext file, such as a Word document. Do not include your name, your ID number, or the ciphertext as part of your submission .You have unlimited submissions.