Assignment 1

Hyoungshick Kim

September 10, 2017

1 Breaking a Vigenere variant cipher

This is a programming assignment to test your understanding of cryptanalysis for Vigenere cipher.

- Your goal is to implement a code to breake a Vigenere variant where byte-wise XOR is used instead of addition modulo 26.
- You will write a code in the C programming language to print out the key and the plaintext decrypted from a given ciphertext in the input file named `hw1_input.txt`'. Please write the key and the plaintext into the output file named '`hw1_output.txt`'. Your program will read a ciphertext in the input file, find a key which is used for the ciphertext and then print out the original plaintext in the output file. In the output file, the first line and the second line represent the key and the plaintext, respectively.
- For test, the key length will be less than or equal to 10 bytes; the plaintext will be between 1,000 and 5,000 bytes. Please test your code extensively with several inputs, so you are sure it works correctly.
- The following is an example of input and output files:

```
[Input file: hw1_input.txt]
Igomm#bpzqvl!ulsng

[Output file: hw1_output.txt]
0x01 0x02 0x03
Hello crypto world!
```

- Ciphertext was encrypted with the following C program.

```
#include <stdio.h>

#define KEY_LENGTH 3 // Can be anything from 1 to 10

int main() {
        unsigned char ch;
        FILE *fpIn, *fpOut;

        unsigned char key[KEY_LENGTH] = { 0x01, 0x02, 0x03 };
        /* key values can be changed */

        fpIn = fopen("plaintext.txt", "r");
        fpOut = fopen("hw1_input.txt", "wb");
        for (int i = 0; fscanf(fpIn, "%c", &ch) != EOF; ++i) {
                ch ^= key[i % KEY_LENGTH];
                fwrite(&ch, sizeof(ch), 1, fpOut);
        }

        fclose(fpIn);
        fclose(fpOut);
        return 0;
}
```

- You will be judged by (1) the correctness of the results returned by your submitted program, (2) the actual running time of the program and (3) the well-written document to explain your source code and the performance analysis of your algorithm.
- Your code should be written in ANSI C. We will use the GNU compiler (i.e., `gcc`) to compile your source code.
- Please upload your source code (c files), instructions to illustrate how your source code works, document to explain your code and the performance analysis to iCampus. Submit your assignment by midnight, Sunday September 24; **late submissions are allowed with a penalty. Each 24 hours (or part thereof) late will cost you 20%.**
- **Your assignments must be your own original work.** We will use a tool to check for plagiarism in assignments.