2^{nd} Implementation Project

MC889/MO421 – Introduction to Cryptography

Prof. Diego de Freitas Aranha

2015/1

1 Introduction

The objective of this project is to implement an image encryption method to visually compare the statistical properties of different symmetric ciphers.

2 Method

In this wok, you should implement two simple encryption methods: the affine cipher and TEA. The affine cipher is the more general case of a classical substitution cipher which applies a linear transformation to every plaintext element. The encryption function is given by $E_{a,b}(x) = (ax + b) \pmod{m}$ and the decryption function is given by $D_{a,b}(y) = a^{-1}(y - b) \pmod{m}$, with a and m co-prime. The integer pair (a, b) is the shared key. For modern implementations, consider $m = 256$ and that a byte is processed each turn.

The TEA cipher (*Tiny Encryption Algorithm*) is a minimalist block cipher that employs modern design concepts and requires low computational power. The details of implementing the cipher can be found in the document `tea.pdf`.

A possible method for encrypting images is to consider an image as a sequence of bytes that specify the RGB color components of consecutive pixels. At first, any cipher can be implemented over an image by appending null bytes to the image until its length is a multiple of the block size. However, this can create ambiguity during decryption depending on the image format. For simplifying the padding handling, consider for now that the input image always has dimensions aligned with the block size of the underlying cipher.

3 Material

A program is provided in the C programming language to encrypt messages in the PPM format using the polyalphabetic Vigenère cipher. The source code can be found in file `encrypt_image.c`. Use this file to understand the image encryption method and the interface of the functions that read and write images.

To compile this file, execute:

```
$ gcc encrypt_image.c ppm.c -o encrypt_image
```

To use the program in encryption mode:

```
$ ./encrypt_image -e -v input.ppm output.ppm
```

The program will ask for the key length and a sequence of integers specifying the key:

```
Key size: 5
Encryption key: 40 67 23 200 56
```

To use the program in decryption mode:

```
$ ./encrypt_image -d -v output.ppm input.ppm
```

The key shall be provided as in the previous manner.

To convert images to the PPM file format, use *ImageMagick*:

```
$ convert image_file image.ppm
```

4 Objective

The objective is to implement the affine cipher and TEA and observe the statistical properties of the two ciphers from the execution of the program with different images. The modes of operation for TEA are ECB (*Electronic CodeBook*) and another method of your choice. The affine cipher should ask for a pair of integers (a, b) between 0 and 255 and check if a and 256 are co-prime. The TEA cipher should ask for a sequence of 4 32-bit integers as the encryption key, provided in hexadecimal format.

5 Evaluation

The submission must include the program source code, implementing encryption and decryption using the affine and TEA ciphers. You can use any programming language you want, given that instructions for compilation/execution are provided. The submission must be followed by a brief report containing sample images that illustrate the following issues:

- What are are the main differences in perception from images encrypted by Vigenère, affine and TEA ciphers?
- Does using TEA in ECB mode have visible limitations? Provide examples to justify your observations.
- How can the usage of TEA be improved security-wise? Implement an improved method through an additional option in the main program and illustrate the security improvement with examples.
- How can the problem with block size boundaries be solved more elegantly? Implement an improved method through an additional option in the main program and justify the improvement.

Notice that the two last issues can be solved by the same method. The submission deadline is May 28 and the project is individual.