Lab 6: Block Cipher

50.020 Security

Hand-out: March 15
Hand-in: March 22, 9pm

1 Objective

By the end of this lab, you should be able to:

- Implement the ultra-lightweight PRESENT block cipher

2 Background

- Read on PRESENT: an Ultra-lightweight Block Cipher (Section 3):
  - `http://yannickseurin.free.fr/pubs/Bogdanov_et_al07_CHES.pdf`
  - `http://www.emsec.rub.de/media/crypto/veroeffentlichungen/2011/01/29/present_ches2007_slides.pdf`
- Read Netpbm image format:
  - `http://en.wikipedia.org/wiki/Netpbm_format`

3 Implementing PRESENT

- Use Python script `present.py` as a template to implement PRESENT block cipher with 80-bit key and 64-bit block length.
- You need to implement both the encryption and decryption portion.
- Check your result using Appendix 1 of the above paper. Note that `present.py` provides those test cases at the end of the file.

4 Implementing ECB Mode

- We are going to encrypt an image (`Tux.ppm`). Check whether you can see the image by typing:

  ```
  $ display Tux.ppm
  ```

  Install ImageMagick if you cannot run display.

```
$ sudo apt-get install imagemagick
```

- The above implementation is for a plaintext of 64-bit length. In this section, you need to extend your code so that it can work with plaintext larger than 64-bit. Use Electronic Codebook Mode (ECB) for this purpose.
- Use your ECB mode block cipher to encrypt the file `Tux.ppm`. Decrypt the file and see whether you can still see the same image. Use `ecb.py` and run the file as the following.

```
$ python ecb.py -i [input filename] -o [output filename] -k [key filename]
-m [mode]
```

5 Limitation of ECB Mode

- ECB mode reveals some side-channel information about the plaintext pattern in the ciphertext. In this section, we will try to learn information about the plaintext image from its ECB ciphertext.
- Download `letter.e`. This is a secret image, encrypted in ECB mode, with a secret key. Your task is to recover the plaintext of the original image. The image is stored in PBM format and has its header information stored in the file `header.pbm` (that is known to the attacker, i.e. you).
- Write a Python script to extract the image pattern from `letter.e`. You can use `extract.py` as a starting point.
  - *Hint: You can ignore the first few characters which represent the header as it has already been given. The plaintext PBM image is black and white, i.e. the format only has two values, either 0 or 1. You can assume that there is no space between the data values.*

```
$ python extract.py -i [input filename] -o [output filename] -hh [known header file]
```

6 Hand-In

- Submit your present.py, ecb.py, and extract.py together with the decrypted letter.e
- Make sure to put your username into the header of the submitted file.