Project # 4

1. Cache Collision Timing Attack

The goal of this exercise is to reproduce the results of the Bonneau paper on the AES implementation of `OpenSSL 0.9.7`. For this you need to follow the following steps:

1. Make sure you are in a Lunix system with superuser rights. If you don't have linux, you can for example install Ubuntu live on an empty USB flash drive and boot from that.
   `http://www.ubuntu.com/download/desktop/create-a-usb-stick-on-windows`

2. Install OpenSSL 0.9.7. You can download it at:
   `http://users.wpi.edu/~teisenbarth/stuff/openssl-0.9.7a.tar.gz`

   After downloading and unpacking it can be installed as follows:

   ```
   ./config --prefix=/PATH_TO_DESTINATION

   sudo make

   sudo make install
   ```

3. Disable ASLR:
   ```
   echo 0 | sudo tee /proc/sys/kernel/randomize_va_space
   ```
   To re-enable it after the homework, substitute the 0 with a 2

4. Write a code that performs about 5 million encryptions and measures the execution time of each. Set the key to be all-zero, since . Make sure to flush the *entire* table T_4 before each encryption using the `clflush` command. For this you need to determine the address of the table T_4 using `gdb`:

5. Compile the code with the library
   ```
   gcc -o code code.c -I /PATH_TO_DESTINATION/include
             -L /PATH_TO_DESTINATION/lib -lcrypto
   ```

   - start gdb:
     `gdb fnr_template`
   - set break point in code:
     `break AES_encrypt`
   - execute until break point
     `run`
   - find position of fourth t-table:
     `p &Te4`

It is recommended to only write about 500k measurements (ciphertext-timing pairs) into one file to easier convert them to matlab or your analysis tool of choice.

Next, analyze the measurements generated on your CPU to recover the last round key using Bonneau's attack described in class. Templates for both generating the measurements are provided. Please submit your measurement code, your analysis code, the correct last round key, the number of measurements used by you, and, for one byte pair of your choice, the average AES execution time (y-axis) per differential (x-axis) in a plot.

2. Flush and Reload Attack

In this part the flush and reload technique will be used in OpenSSL 0.9.7 library to extract the last round key of AES encryption as done in the Irazoqui paper. All steps which are done in the first question should be repeated again for this part. One of the differences from the first question is the measuring the time for reloading instead of encryption. The other difference is flushing one cache line of the T_4 table is sufficient.

As before, a template for getting the measurements is provided. Please submit your measurement code, your analysis code, the correct last round key, the number of measurements used by you, and, for one byte pair of your choice, the average miss rate (y-axis) per key guess (x-axis) in a plot.

Good Luck and Have Fun!