Introduction to Cryptography, Spring 2021

Homework 2: DES Programming

Due: 2021/3/23 (Wednesday)

1. This homework is about the implementation of the DES core function, which encrypts a block of plaintext to a block of ciphertext with a key of 64 bits (with parity bits).
   a. Input format: the input is an ordered pair of keys and plaintexts in characters, such as "12345678 Advanced". Each character is interpreted as its ASCII code, e.g., 'A' = 41 (Hex)
   b. Output format: 16 hex characters, such as AE184796707E59FB, which is the ciphertext of the above key and plaintext.
   c. You can use the following key-plaintext-ciphertext tuple as a test sample for correctness: 12345678 Advanced AE184796707E59FB
   d. Use C or C++ to write your code.
2. Submission to E3 with two files.
   a. The source code file with name: DES.c or DES.cpp.
      i. The output file "out.txt" that contains 5 lines of ciphertexts for the ordered pairs of key and plaintext (one pair per line) from the file "DES-Key-Plaintext.txt".
      ii. One line of time (in milliseconds) for the running time of each DES encryption.
3. On-site test
   a. Test site: to be announced. You need to go to the computer room for the on-site test at specified time.
   b. TA will ask you to modify your DES program for a modified specification MDES.
   c. You need to show the MDES ciphertext for the ordered pair of key and plaintext, which will be given on site.
   d. You need to show the running time for the above encryption.
4. Grade evaluation
   a. If you fail the on-site test, you fail this homework.
   b. Correctness of out.txt.
   c. Performance of your program by averaging 1000 times of one encryption.
5. TA will run a plagiarism checker on your programs to check plagiarism. So, write your own code, do not copy from others or anywhere.
6. You can use the following code to compute the running time of a function

```
#include <time.h>

clock_t start, end;

double cpu_time_used;

start = clock();

... /* Do the work. */

end = clock();
```

```
    cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
```