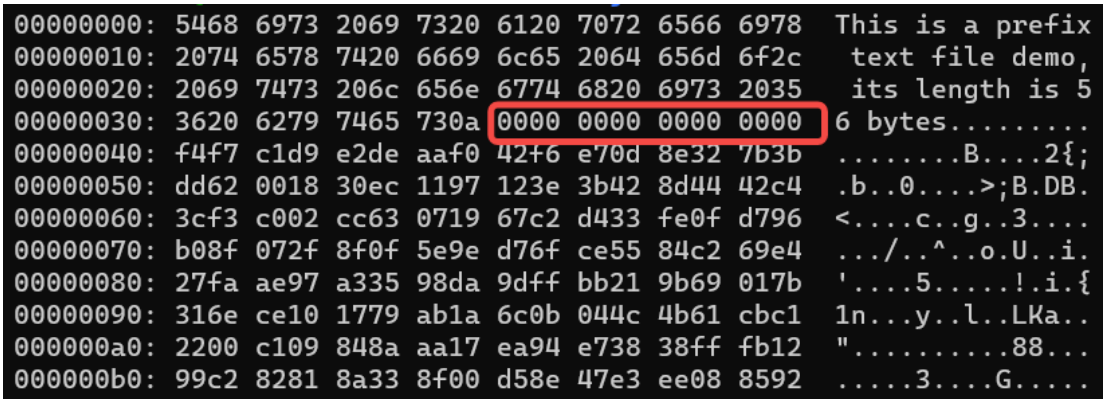
ECE 571: Lab 3

> *Date: 2022-03-14*
>
> *Author: Zhaohui Yang*

Task 1: Generating Two Different Files with the Same MD5 Hash

Question 1 If the length of your prefix file is not multiple of 64, what is going to happen?

When the prefix file is not multiple of 64 (herein it is a 56-byte text file), the plaintext will be padded at the tail before MD5 encryption. Here in this case, 8 bytes `0000 0000 0000 0000` are padded at the tail.



The output files are different when using `diff` command as well as observing their hexadecimal presentations using `xxd` command.

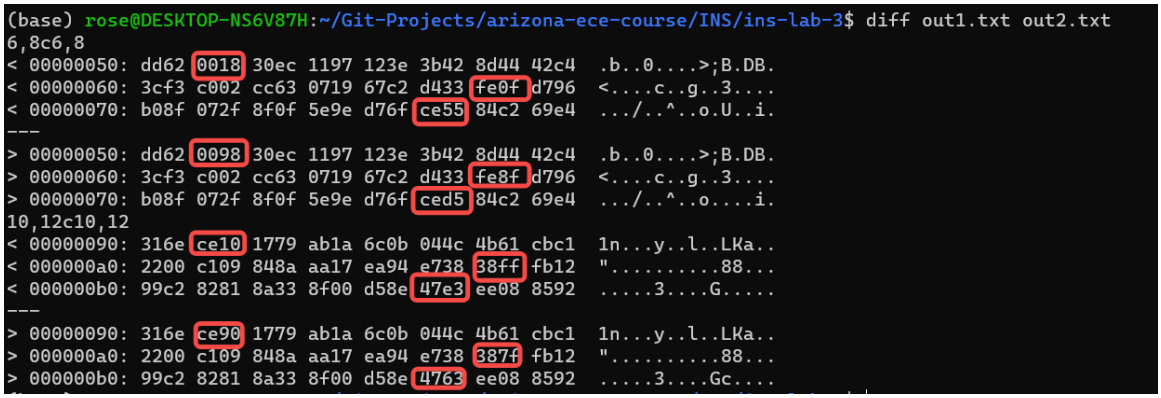
Question 2 Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what happens.

Here in a 64-byte file "prefix-64.txt" was created and two output files "out1-64.bin" and "out2-64.bin" are generated by `md5collgen` command. Since the input file is 64-byte, there is no padding operation in MD5 encryption process, thus the output files are also 192-byte (192=64+128).

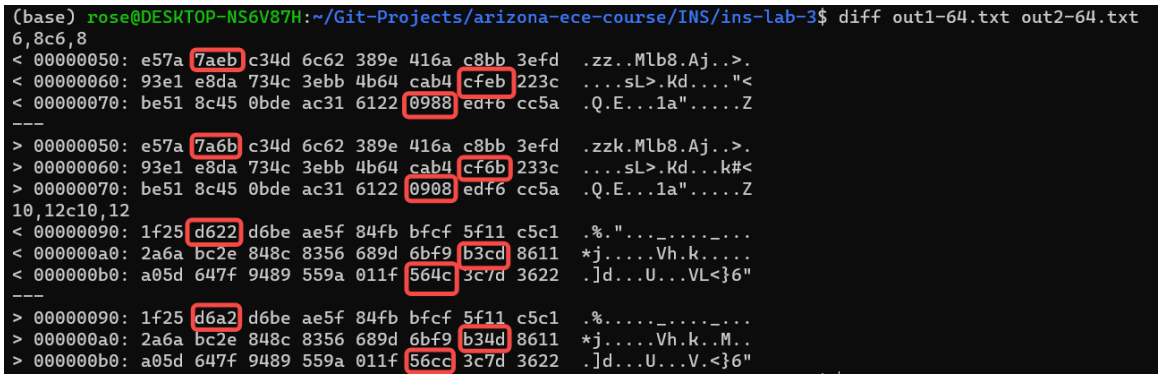
Question 3 Are the data (128 bytes) generated by md5collgen completely different for the two output files? Please identify all the bytes that are different.

They are not completely different, no matter using 56-byte prefix or 64-file prefix file, respectively. The different byte sequences are all noted by red circles in the following figures.

1. *output files of 56-byte prefix file*



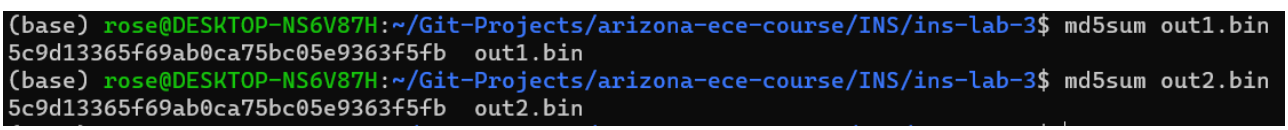
2. *output files of 64-byte prefix file*



In conclusion, there are just several bytes are different between the two encrypted 128-byte digests.

Task 2: Understanding MD5's Property

```
echo "a suffix text file" > suffix.txt
cat out1.bin suffix.txt > modified1
cat out2.bin suffix.txt > modified2
echo "observing MD5 value of files with the same initial MD5 values
after appended with the same suffix file"
md5sum modified1
md5sum modified2
```



```
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ md5sum modified1
aa4d6c3f8bfe7fb4bc7af4d1518fc49b  modified1
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ md5sum modified2
aa4d6c3f8bfe7fb4bc7af4d1518fc49b  modified2
```

Since the MD5 values of files "out1.bin" and "out2.bin" are the same, with the same suffix file "suffix.txt", the MD5 values of these two modified files are exactly the same.

Task 3: Generating Two Executable Files with the Same MD5 Hash

Shell script in this task:

```
gcc -o xyz_arr xyz_arr.c
# .... find the splitting offset of "xyz_arr"
# that is 12352, 12480
head -c 12352 xyz_arr > prefix-xyz
tail -c +12480 xyz_arr > suffix-xyz
md5collgen -p prefix-xyz -o file1 file2
cat file1 suffix-xyz > exe1
cat file2 suffix-xyz > exe2
chmod +x exe1
chmod +x exe2
./exe1 # different output of "./exe2"
./exe2
md5sum exe1 # same value with "exe2"
md5sum exe2
```

Herein I fill in the array `xyz` with 200 the same elements `0x41`. Then compile "xyz_arr.c" to generate the executable file "xyz_arr". By means of some binary-file visualization tools or Linux commands, the splitting offset of file "xyz_arr" is 12352 and 12480.

```
00003000: 0000 0000 0000 0000 0840 0000 0000 0000  .........@......
00003010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003020: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003030: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003040: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003050: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003060: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003070: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003080: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003090: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
000030a0: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
000030b0: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
000030c0: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
000030d0: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
000030e0: 4141 4141 4141 4141 4743 433a 2028 5562  AAAAAAAAGCC: (Ub
```

Then using commands in the above shell script, modified executable files"exe1" and "exe2" are generated. Eventually the two executable files output different results, but they have the same MD5 value.

```
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ ./exe1
41414141414141414141414141414141414141414141414141414141414141414141414181f912195a9db3c29131b81d7de3f8c8a1dfbbef61b12
8e16398e8d30b8e48ee4c3fcd8b49156745eface55b834572d38e6d162c33c1cc58811efed835118a48a205f1093524e375286e74945d
1b17f3859423a2bc34070aa5fa8c7cf35fce79fc86e367978a72e257f69080175fadf9fb75494a3379fdc03c425f414141414141414141
41414141414141414141414141414141414141414141414141414141414141414141414141
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ ./exe2
41414141414141414141414141414141414141414141414141414141414141414141414181f912195a9db3c29131b81d7de3f8c8a1df3bef61b12
8e16398e8d30b8e48ee4c3fcd8b49156745eface855b834572d38e6d162c33c1ccd8811efed835118a48a205f1093524e375286e74945
d1b1ff3859423a2bc34070aa5fa8c7cf35fce79fc86e367978a72e2d7f59080175fadf9fb75494a33797dc03c425f414141414141414141
41414141414141414141414141414141414141414141414141414141414141414141414141
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ md5sum exe1
8ed35dbf09c3b198302ee1d96aa1337a  exe1
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ md5sum exe2
8ed35dbf09c3b198302ee1d96aa1337a  exe2
```