

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.

```
00000000: 5468 6973 2069 7320 6120 7072 6566 6978 This is a prefix
00000010: 2074 6578 7420 6669 6c65 2064 656d 6f2c text file demo,
00000020: 2069 7473 206c 656e 6774 6820 6973 2035 its length is 5
00000030: 3620 6279 7465 730a 0000 0000 0000 0000 6 bytes.....
00000040: f4f7 c1d9 e2de aaf0 42f6 e70d 8e32 7b3b .....B....2{;
00000050: dd62 0018 30ec 1197 123e 3b42 8d44 42c4 .b..0....>;B.DB.
00000060: 3cf3 c002 cc63 0719 67c2 d433 fe0f d796 <....c..g..3....
00000070: b08f 072f 8f0f 5e9e d76f ce55 84c2 69e4 .../..^..o.U..i.
00000080: 27fa ae97 a335 98da 9dff bb21 9b69 017b '....5.....!.i.{
00000090: 316e ce10 1779 ab1a 6c0b 044c 4b61 cbc1 1n...y..l..LKa..
000000a0: 2200 c109 848a aa17 ea94 e738 38ff fb12 ".....88...
000000b0: 99c2 8281 8a33 8f00 d58e 47e3 ee08 8592 .....3....G.....
```

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

```
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ diff out1.txt out2.txt
6,8c6,8
< 00000050: dd62 0018 30ec 1197 123e 3b42 8d44 42c4 .b..0...>;B.DB.
< 00000060: 3cf3 c002 cc63 0719 67c2 d433 fe0f d796 <....c..g..3....
< 00000070: b08f 072f 8f0f 5e9e d76f ce55 84c2 69e4 .../..^..o.U..i.
---
> 00000050: dd62 0098 30ec 1197 123e 3b42 8d44 42c4 .b..0...>;B.DB.
> 00000060: 3cf3 c002 cc63 0719 67c2 d433 fe8f d796 <....c..g..3....
> 00000070: b08f 072f 8f0f 5e9e d76f ced5 84c2 69e4 .../..^..o....i.
10,12c10,12
< 00000090: 316e ce10 1779 ab1a 6c0b 044c 4b61 cbc1 1n...y..l..LKa..
< 000000a0: 2200 c109 848a aa17 ea94 e738 38ff fb12 ".....88...
< 000000b0: 99c2 8281 8a33 8f00 d58e 47e3 ee08 8592 .....3....G....
---
> 00000090: 316e ce90 1779 ab1a 6c0b 044c 4b61 cbc1 1n...y..l..LKa..
> 000000a0: 2200 c109 848a aa17 ea94 e738 387f fb12 ".....88...
> 000000b0: 99c2 8281 8a33 8f00 d58e 4763 ee08 8592 .....3....Gc....
```

2. output files of 64-byte prefix file

```
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ diff out1-64.txt out2-64.txt
6,8c6,8
< 00000050: e57a 7aeb c34d 6c62 389e 416a c8bb 3efd .zz..Mlb8.Aj..>.
< 00000060: 93e1 e8da 734c 3ebb 4b64 cab4 cfeb 223c ...sL>.Kd...."<
< 00000070: be51 8c45 0bde ac31 6122 0988 ed76 cc5a .Q.E...1a".....Z
---
> 00000050: e57a 7a6b c34d 6c62 389e 416a c8bb 3efd .zzk.Mlb8.Aj..>.
> 00000060: 93e1 e8da 734c 3ebb 4b64 cab4 cf6b 223c ...sL>.Kd...k#<
> 00000070: be51 8c45 0bde ac31 6122 0908 edf6 cc5a .Q.E...1a".....Z
10,12c10,12
< 00000090: 1f25 d622 d6be ae5f 84fb bfcf 5f11 c5c1 .%. ".....
< 000000a0: 2a6a bc2e 848c 8356 689d 6bf9 b3cd 8611 *j.....Vh.k.....
< 000000b0: a05d 647f 9489 559a 011f 564c 3c7d 3622 .]d...U...VL<}6"
---
> 00000090: 1f25 d6a2 d6be ae5f 84fb bfcf 5f11 c5c1 .%. ".....
> 000000a0: 2a6a bc2e 848c 8356 689d 6bf9 b34d 8611 *j.....Vh.k..M...
> 000000b0: a05d 647f 9489 559a 011f 56cc 3c7d 3622 .]d...U...V.<}6"
```

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 out1.bin
5c9d13365f69ab0ca75bc05e9363f5fb out1.bin
(base) rose@DESKTOP-NS6V87H:~/Git-Projects/arizona-ece-course/INS/ins-lab-3$ md5sum out2.bin
5c9d13365f69ab0ca75bc05e9363f5fb out2.bin
```

```
(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 0000 0340 0000 0000 0000 .....@.....
00003010: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00003020: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
00003030: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
00003040: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
00003050: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
00003060: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
00003070: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
00003080: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
00003090: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
000030a0: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
000030b0: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
000030c0: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
000030d0: 4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAAAA
000030e0: 4141 4141 4141 4141 4743 433a 2028 5562 AAAAAAAGCC: (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.

[illegible]