

姓名：蔡佩蓉 學號：109511286

## MD5 Collision Attack Lab (Lab4)

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

#### Questions:

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

```
seed@VM: ~/Desktop
[05/16/24]seed@VM:~/Desktop$ echo "MD5 Collision Attack Lab Task 1" > prefix.txt
[05/16/24]seed@VM:~/Desktop$ md5collgen -p prefix.txt -o out1.bin out2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1.bin' and 'out2.bin'
Using prefixfile: 'prefix.txt'
Using initial value: 2ec4f2fabd5e01c34fc8ee7bcad4338a

Generating first block: .....
.....
Generating second block: S11.....
Running time: 78.5462 s
[05/16/24]seed@VM:~/Desktop$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[05/16/24]seed@VM:~/Desktop$ md5sum out1.bin
965c2e2bff496325a3e75f4ce3796158 out1.bin
[05/16/24]seed@VM:~/Desktop$ md5sum out2.bin
965c2e2bff496325a3e75f4ce3796158 out2.bin
[05/16/24]seed@VM:~/Desktop$
```

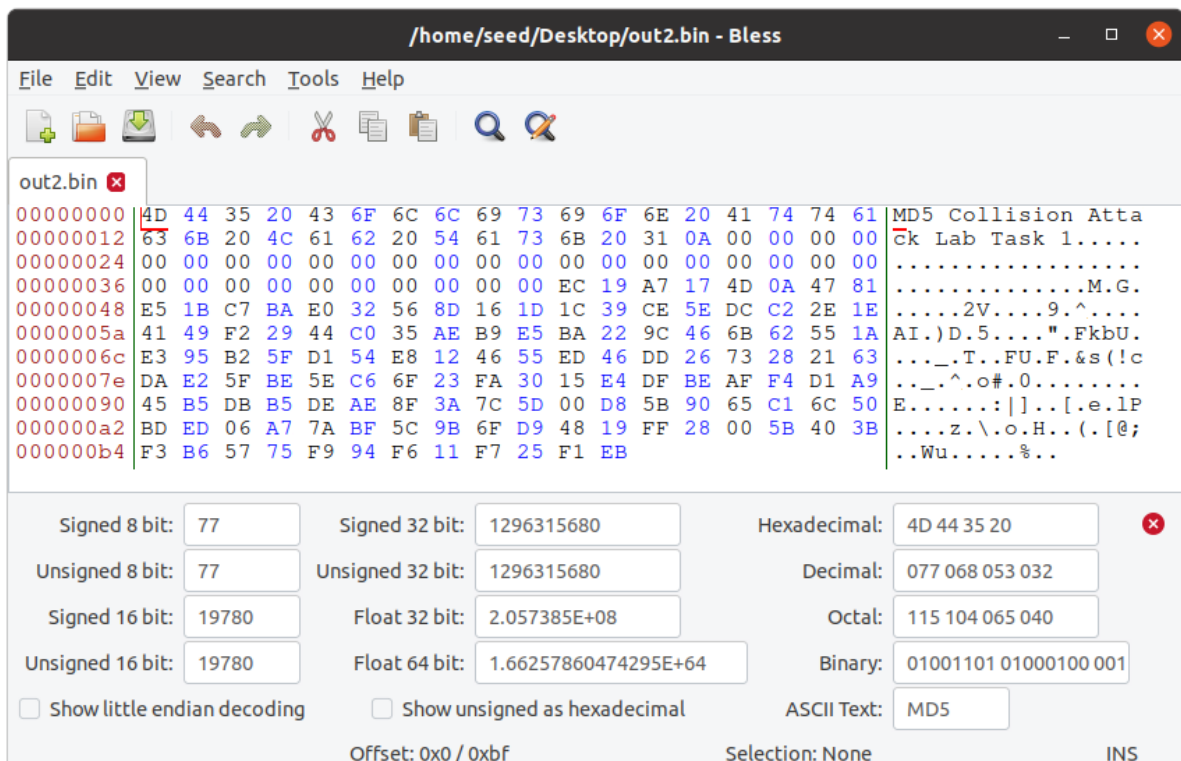
/home/seed/Desktop/out1.bin - Bless

File Edit View Search Tools Help

out1.bin x

00000000	4D 44 35 20 43 6F 6C 6C 69 73 69 6F 6E 20 41 74 74 61	MD5 Collision Atta
00000012	63 6B 20 4C 61 62 20 54 61 73 6B 20 31 0A 00 00 00	ck Lab Task 1.....
00000024	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000036	00 00 00 00 00 00 00 00 00 00 EC 19 A7 17 4D 0A 47 81	.....M.G.
00000048	E5 1B C7 BA E0 32 56 8D 16 1D 1C B9 CE 5E DC C2 2E 1E	.....2V.....^.....
0000005a	41 49 F2 29 44 C0 35 AE B9 E5 BA 22 9C 46 6B 62 55 1A	AI.)D.5....".FkbU.
0000006c	E3 15 B2 5F D1 54 E8 12 46 55 ED 46 DD 26 73 A8 21 63	...T..FU.F.&s.!c
0000007e	DA E2 5F BE 5E C6 6F 23 FA 30 15 E4 DF BE AF F4 D1 A9	...^..o#.0.....
00000090	45 B5 DB 35 DE AE 8F 3A 7C 5D 00 D8 5B 90 65 C1 6C 50	E..5....: ]..[.e.lP
000000a2	BD ED 06 A7 7A BF 5C 9B 6F D9 48 99 FF 28 00 5B 40 3B	....z.\.o.H..([@;
000000b4	F3 B6 57 75 F9 94 F6 91 F7 25 F1 EB	..Wu.....%..

Signed 8 bit:	77	Signed 32 bit:	1296315680	Hexadecimal:	4D 44 35 20
Unsigned 8 bit:	77	Unsigned 32 bit:	1296315680	Decimal:	077 068 053 032
Signed 16 bit:	19780	Float 32 bit:	2.057385E+08	Octal:	115 104 065 040
Unsigned 16 bit:	19780	Float 64 bit:	1.66257860474295E+64	Binary:	01001101 01000100 001
<input type="checkbox"/> Show little endian decoding		<input type="checkbox"/> Show unsigned as hexadecimal		ASCII Text: MD5	
Offset: 0x0 / 0xbf				Selection: None INS	



If the length of the prefix file is not a multiple of 64 bytes, the md5collgen tool will add padding to the prefix (padded with zeros). MD5 operates on 512-bit (64-byte) blocks, so the tool ensures that the prefix (plus padding) aligns to a 64-byte boundary before appending the collision blocks.

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

```
seed@VM: ~/Desktop
[05/16/24]seed@VM:~/Desktop$ echo "MD5 Collision Attack Lab Task 1 MD5 Collision Attack Lab Task 1" > prefix64.txt
[05/16/24]seed@VM:~/Desktop$ md5collgen -p prefix64.txt -o out1_64.bin out2_64.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1_64.bin' and 'out2_64.bin'
Using prefixfile: 'prefix64.txt'
Using initial value: 90a77ba13e4ba412473710293110f4fe

Generating first block: .....
Generating second block: S01..
Running time: 2.39742 s
[05/16/24]seed@VM:~/Desktop$ diff out1_64.bin out2_64.bin
Binary files out1_64.bin and out2_64.bin differ
[05/16/24]seed@VM:~/Desktop$ md5sum out1_64.bin
a6372a4f720ade04909b33b368506118 out1_64.bin
[05/16/24]seed@VM:~/Desktop$ md5sum out2_64.bin
a6372a4f720ade04909b33b368506118 out2_64.bin
[05/16/24]seed@VM:~/Desktop$
```

**/home/seed/Desktop/out1\_64.bin - Bless**

File Edit View Search Tools Help

out1\_64.bin

```

00000000 4D 44 35 20 43 6F 6C 6C 69 73 69 6F 6E 20 41 74 74 61 MD5 Collision Atta
00000012 63 6B 20 4C 61 62 20 54 61 73 6B 20 31 20 4D 44 35 20 ck Lab Task 1 MD5
00000024 43 6F 6C 6C 69 73 69 6F 6E 20 41 74 74 61 63 6B 20 4C Collision Attack L
00000036 61 62 20 54 61 73 6B 20 31 0A DC E2 DE 53 A0 3B E9 D8 ab Task 1....S;...
00000048 4A CC 72 3F A2 6F B3 0D E4 F4 6D 3E 73 37 EF 01 31 D2 J.r?.o.....m>s7..l.
0000005a 41 ED 6B 10 A5 1D 04 DC 07 C3 C8 70 16 02 87 C2 53 23 A.k.....p....S#
0000006c 05 64 6D 73 F4 0D D8 C9 56 31 5D 18 D9 64 FD A1 A6 FB .dms....Vl]..d....
0000007e DD 3A F2 3F 9C 4A 74 75 EC D1 0D 1B CF 00 C1 C0 05 48 ...?.Jtu.....H
00000090 08 DC CC 77 05 E6 35 EB C9 1A 06 C8 EC C7 DD 6F 7E 4D ...w..5.....o~M
000000a2 9E 95 F4 0F 08 14 98 4B 3D D7 9D 80 41 AE 11 8A 3A B2 .....K=...A.....
000000b4 3D E7 19 30 EF A1 20 29 67 89 78 DE =..0... )g.x.

```

Signed 8 bit: 77      Signed 32 bit: 1296315680      Hexadecimal: 4D 44 35 20  
 Unsigned 8 bit: 77      Unsigned 32 bit: 1296315680      Decimal: 077 068 053 032  
 Signed 16 bit: 19780      Float 32 bit: 2.057385E+08      Octal: 115 104 065 040  
 Unsigned 16 bit: 19780      Float 64 bit: 1.66257860474295E+64      Binary: 01001101 01000100 001  
☐ Show little endian decoding      ☐ Show unsigned as hexadecimal      ASCII Text: MD5  
 Offset: 0x0 / 0xbf      Selection: None      INS

**/home/seed/Desktop/out2\_64.bin - Bless**

File Edit View Search Tools Help

out2\_64.bin

```

00000000 4D 44 35 20 43 6F 6C 6C 69 73 69 6F 6E 20 41 74 74 61 MD5 Collision Atta
00000012 63 6B 20 4C 61 62 20 54 61 73 6B 20 31 20 4D 44 35 20 ck Lab Task 1 MD5
00000024 43 6F 6C 6C 69 73 69 6F 6E 20 41 74 74 61 63 6B 20 4C Collision Attack L
00000036 61 62 20 54 61 73 6B 20 31 0A DC E2 DE 53 A0 3B E9 D8 ab Task 1....S;...
00000048 4A CC 72 3F A2 6F B3 0D E4 F4 6D BE 73 37 EF 01 31 D2 J.r?.o.....m.s7..l.
0000005a 41 ED 6B 10 A5 1D 04 DC 07 C3 C8 70 16 02 87 C2 53 23 A.k.....p....S#
0000006c 05 E4 6D 73 F4 0D D8 C9 56 31 5D 18 D9 64 FD 21 A6 FB ..ms....Vl]..d.!...
0000007e DD 3A F2 3F 9C 4A 74 75 EC D1 0D 1B CF 00 C1 C0 05 48 ...?.Jtu.....H
00000090 08 DC CC F7 05 E6 35 EB C9 1A 06 C8 EC C7 DD 6F 7E 4D .....5.....o~M
000000a2 9E 95 F4 0F 08 14 98 4B 3D D7 9D 00 41 AE 11 8A 3A B2 .....K=...A.....
000000b4 3D E7 19 30 EF A1 20 A9 67 89 78 DE =..0... .g.x.

```

Signed 8 bit: 77      Signed 32 bit: 1296315680      Hexadecimal: 4D 44 35 20  
 Unsigned 8 bit: 77      Unsigned 32 bit: 1296315680      Decimal: 077 068 053 032  
 Signed 16 bit: 19780      Float 32 bit: 2.057385E+08      Octal: 115 104 065 040  
 Unsigned 16 bit: 19780      Float 64 bit: 1.66257860474295E+64      Binary: 01001101 01000100 001  
☐ Show little endian decoding      ☐ Show unsigned as hexadecimal      ASCII Text: MD5  
 Offset: 0x0 / 0xbf      Selection: None      INS

The md5collgen tool will not need to add any padding before appending the collision blocks (no zero padding). The two output files will still share the same 64-byte prefix, followed by their respective collision blocks.

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

```
seed@VM: ~/Desktop
[05/16/24]seed@VM:~/Desktop$ md5collgen -p prefix.txt -o out1.bin out2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1.bin' and 'out2.bin'
Using prefixfile: 'prefix.txt'
Using initial value: 2ec4f2fabd5e01c34fc8ee7bcad4338a

Generating first block: .....
Generating second block: S10.....
Running time: 50.0171 s
[05/16/24]seed@VM:~/Desktop$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[05/16/24]seed@VM:~/Desktop$ md5sum out1.bin
08a350112d31c10909535856ald2c918 out1.bin
[05/16/24]seed@VM:~/Desktop$ md5sum out2.bin
08a350112d31c10909535856ald2c918 out2.bin
[05/16/24]seed@VM:~/Desktop$
```

```
seed@VM: ~/Desktop
[05/16/24]seed@VM:~/Desktop$ hexdump -C out1.bin > out1.hex
[05/16/24]seed@VM:~/Desktop$ hexdump -C out2.bin > out2.hex
[05/16/24]seed@VM:~/Desktop$ diff out1.hex out2.hex
6,8c6,8
< 00000050 f9 49 77 88 4a e9 6d 91 1f 62 3a c1 d0 88 b9 62 |.Iw.J.m..b:...b|
< 00000060 59 ce ba f4 6c a1 f1 c7 8b 44 cf b9 aa 1f 01 f6 |Y...l...D.....|
< 00000070 14 a8 d7 d6 6e ec 6c 3b 69 e8 28 ca 35 7d 35 ae |....n.l;i.(5}5.|
---
> 00000050 f9 49 77 08 4a e9 6d 91 1f 62 3a c1 d0 88 b9 62 |.Iw.J.m..b:...b|
> 00000060 59 ce ba f4 6c a1 f1 c7 8b 44 cf b9 aa 9f 01 f6 |Y...l...D.....|
> 00000070 14 a8 d7 d6 6e ec 6c 3b 69 e8 28 4a 35 7d 35 ae |....n.l;i.(5}5.|
10,12c10,12
< 00000090 4b 13 c6 73 70 5e 13 b2 29 c5 10 68 b7 5f 88 d7 |K..5p^..).h...|
< 000000a0 cb 4d 32 f8 9f c7 82 5e f7 01 c8 73 52 fc e9 01 |.M2....^...sR...|
< 000000b0 9d 4b 34 cb 0e d7 14 b9 ae e3 f5 cc 1d b5 b2 9c |.K4......0....|
---
> 00000090 4b 13 c6 f3 70 5e 13 b2 29 c5 10 68 b7 5f 88 d7 |K..5p^..).h...|
> 000000a0 cb 4d 32 f8 9f c7 82 5e f7 01 c8 73 52 7c e9 01 |.M2....^...sR...|
> 000000b0 9d 4b 34 cb 0e d7 14 b9 ae e3 f5 4c 1d b5 b2 9c |.K4......0....|
[05/16/24]seed@VM:~/Desktop$
```

```
seed@VM: ~/Desktop
[05/16/24]seed@VM:~/Desktop$ hexdump -C out1_64.bin > out1_64.hex
[05/16/24]seed@VM:~/Desktop$ hexdump -C out2_64.bin > out2_64.hex
[05/16/24]seed@VM:~/Desktop$ diff out1_64.hex out2_64.hex
6,8c6,8
< 00000050 e4 f4 6d 3e 73 37 ef 01 31 d2 41 ed 6b 10 a5 1d |...57...1.A.k...|
< 00000060 04 dc 07 c3 c8 70 16 02 87 c2 53 23 05 64 6d 73 |....p....S#.0ms|
< 00000070 f4 0d d8 c9 56 31 5d 18 d9 64 fd a1 a6 fb dd 3a |...V1]..d.0....|
---
> 00000050 e4 f4 6d be 73 37 ef 01 31 d2 41 ed 6b 10 a5 1d |...57...1.A.k...|
> 00000060 04 dc 07 c3 c8 70 16 02 87 c2 53 23 05 e4 6d 73 |....p....S#.0ms|
> 00000070 f4 0d d8 c9 56 31 5d 18 d9 64 fd 21 a6 fb dd 3a |...V1]..d.0....|
10,12c10,12
< 00000090 08 dc cc 77 05 e6 35 eb c9 1a 06 c8 ec c7 dd 6f |...5.....o|
< 000000a0 7e 4d 9e 95 f4 0f 08 14 98 4b 3d d7 9d 80 41 ae |~M.....K=...A.|
< 000000b0 11 8a 3a b2 3d e7 19 30 ef a1 20 29 67 89 78 de |...=.0..0g.x.|
---
> 00000090 08 dc cc f7 05 e6 35 eb c9 1a 06 c8 ec c7 dd 6f |...5.....o|
> 000000a0 7e 4d 9e 95 f4 0f 08 14 98 4b 3d d7 9d 00 41 ae |~M.....K=...A.|
> 000000b0 11 8a 3a b2 3d e7 19 30 ef a1 20 a9 67 89 78 de |...=.0..0g.x.|
[05/16/24]seed@VM:~/Desktop$
```

No, not all bytes are different (the different bytes are boxed above).

## Task 2: Understanding MD5's Property

A terminal window titled 'seed@VM: ~/Desktop' showing the execution of the 'md5collgen' tool. The tool generates two files, 'p1' and 'p2', which are MD5 collisions. The user then uses 'diff' to confirm they differ, and 'md5sum' to verify they have the same hash. Finally, the user concatenates each file with the string 's' and verifies that the resulting concatenated files also have the same MD5 hash.

```
[05/16/24]seed@VM:~/Desktop$ md5collgen -p prefix.txt -o p1 p2
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'p1' and 'p2'
Using prefixfile: 'prefix.txt'
Using initial value: 2ec4f2fabd5e01c34fc8ee7bcad4338a

Generating first block: .....
Generating second block: S10.....
Running time: 16.6723 s
[05/16/24]seed@VM:~/Desktop$ diff p1 p2
Binary files p1 and p2 differ
[05/16/24]seed@VM:~/Desktop$ md5sum p1
480c66ab77e248c5b3940d36d46d628e  p1
[05/16/24]seed@VM:~/Desktop$ md5sum p2
480c66ab77e248c5b3940d36d46d628e  p2
[05/16/24]seed@VM:~/Desktop$ echo "abcde" > s
[05/16/24]seed@VM:~/Desktop$ cat p1 s > o1
[05/16/24]seed@VM:~/Desktop$ cat p2 s > o2
[05/16/24]seed@VM:~/Desktop$ md5sum o1
933f1188d55c50a2c3fac05a1d12c0b3  o1
[05/16/24]seed@VM:~/Desktop$ md5sum o2
933f1188d55c50a2c3fac05a1d12c0b3  o2
[05/16/24]seed@VM:~/Desktop$
```

Property of MD5: appending the same suffix to two inputs that have the same MD5 hash will result in the same hash for the concatenated outputs.

This is a fundamental property of many cryptographic hash functions and is crucial for understanding certain types of hash collisions and vulnerabilities.



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

```
seed@VM: ~/Desktop
[05/16/24] seed@VM:~/Desktop$ gcc -o Task3 Task3.c
[05/16/24] seed@VM:~/Desktop$ bless Task3
Gtk-Message: 10:57:04.548: Failed to load module "canberra-gtk-module"
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find file "/home/seed/.config/bless/export_patterns"
```

/home/seed/Desktop/Task3 - Bless

---

**File Edit View Search Tools Help**

Task3 ✖

00002f76 00002f88 00002f9a 00002fac 00002fbe 00002fd0 00002fe2 00002ff4 00003006 00003018 0000302a 0000303c 0000304e 00003060 00003072 00003084 00003096 000030a8 000030ba 000030cc 000030de 000030ff 00003102 00003114 00003126 00003138 0000314a 0000315c 0000316e	<pre> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 00 00 00 00 c0 3d 00 00 00 00 00 00 00 00 00 00 00 00 .....=..... 00 00 00 00 00 00 00 00 00 00 30 10 00 00 00 00 00 00 .....0..... 40 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 @..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 00 00 08 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...@..... 00 00 00 00 00 00 00 00 41 41 41 41 41 41 41 41 41 41 .....AAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAGCC: (Ub 75 6E 74 75 20 39 2E 33 2E 30 2D 31 37 75 62 75 6E 74 untu 9.3.0~17ubunt 75 31 7E 32 30 2E 30 34 29 20 39 2E 33 2E 30 00 00 u1~20.04) 9.3.0... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 01 00 ..... 18 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... 00 00 03 00 02 00 38 03 00 00 00 00 00 00 00 00 00 00 .....8..... 00 00 00 00 00 00 00 00 03 00 03 00 58 03 00 00 00 00 .....X..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04 00 .....           </pre>	
--	---	--

Signed 8 bit:	<div>71</div>	Signed 32 bit:	<div>1195590458</div>	Hexadecimal:	<div>47 43 43 3A</div>	<span style="color: red;">✖</span>
Unsigned 8 bit:	<div>71</div>	Unsigned 32 bit:	<div>1195590458</div>	Decimal:	<div>071 067 067 058</div>	
Signed 16 bit:	<div>18243</div>	Float 32 bit:	<div>49987.23</div>	Octal:	<div>107 103 103 072</div>	
Unsigned 16 bit:	<div>18243</div>	Float 64 bit:	<div>2.00034333882626E+35</div>	Binary:	<div>01000111 01000011 010</div>	

☐ Show little endian decoding    
 ☐ Show unsigned as hexadecimal    
 ASCII Text: 

GCC:

Offset: 0x30e8 / 0x4257                  Selection: 0x3020 to 0x30e7 (0xc8 bytes) INS

Length of prefix needs to be multiple of 64 bytes, 12320 (0x3020) is not multiple of 64, hence use 12352 (0x3040) [prefix = first 12352 bytes]. There is a 128-byte region, hence  $12352 + 128 = 12480$  (0x30C0) as suffix [from 12481<sup>st</sup> byte to the end].

```
seed@VM: ~/Desktop
[05/16/24]seed@VM:~/Desktop$ head -c 12352 Task3 > prefix
[05/16/24]seed@VM:~/Desktop$ md5collgen -p prefix -o a b
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'a' and 'b'
Using prefixfile: 'prefix'
Using initial value: aeefe77439646539fe6db56554d46ae83

Generating first block: ...
Generating second block: S10.....
Running time: 1.58959 s
[05/16/24]seed@VM:~/Desktop$ tail -c +12481 Task3 > suffix
[05/16/24]seed@VM:~/Desktop$ cat a suffix > m
[05/16/24]seed@VM:~/Desktop$ cat b suffix > n
[05/16/24]seed@VM:~/Desktop$ chmod +x m n
[05/16/24]seed@VM:~/Desktop$ md5sum m
5487062a4869a5c7bc1523cb565eec78  m
[05/16/24]seed@VM:~/Desktop$ md5sum n
5487062a4869a5c7bc1523cb565eec78  n
[05/16/24]seed@VM:~/Desktop$
```

```
seed@VM: ~/Desktop
[05/24/24]seed@VM:~/Desktop$ echo $(./m) | md5sum;echo $(./n) | md5sum
4f3fdd9d18c1f9a01f6efd6c864c2e27  -
4a49680c53d1c507ec35c2a29b9c732f  -
[05/24/24]seed@VM:~/Desktop$ md5sum m; md5sum n
5487062a4869a5c7bc1523cb565eec78  m
5487062a4869a5c7bc1523cb565eec78  n
[05/24/24]seed@VM:~/Desktop$
```

We observed that both have different outcomes (different content), but same hash value.

#### Task 4: Making the Two Programs Behave Differently

```
seed@VM: ~/Desktop
[05/24/24] seed@VM:~/Desktop$ gcc Task4.c -o Task4
[05/24/24] seed@VM:~/Desktop$ head -c 12352 Task4 > prefix
[05/24/24] seed@VM:~/Desktop$ md5collgen -p prefix -o p q
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'p' and 'q'
Using prefixfile: 'prefix'
Using initial value: 3287cc4fd721c212b60f5b11195c375d

Generating first block: .....
Generating second block: W.....
Running time: 2.91007 s
[05/24/24] seed@VM:~/Desktop$ tail -c +12745 Task4 > suffix
[05/24/24] seed@VM:~/Desktop$ tail -c 160 p > middle
[05/24/24] seed@VM:~/Desktop$ python3 -c "print('\x00'*40)" > tmp
[05/24/24] seed@VM:~/Desktop$ head -c 40 tmp > m40
[05/24/24] seed@VM:~/Desktop$ head -c 24 tmp > m24
[05/24/24] seed@VM:~/Desktop$ cat p m40 m24 middle m40 suffix > s
[05/24/24] seed@VM:~/Desktop$ cat q m40 m24 middle m40 suffix > k
[05/24/24] seed@VM:~/Desktop$ chmod +x s
[05/24/24] seed@VM:~/Desktop$ chmod +x k
```

Get the last 160 bytes [32 (char A) + 128 (md5 padding)] from p. The length of the array is 200, hence, 40 bytes (200-160) need to be filled in to the a and b arrays. However, the difference between the two arrays is  $0x3100 - 0x3020 = 0xE0$  (224bytes), not 200 bytes, hence insert another 24 bytes of zero in between.



Task4

00002fd0	30 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0.....
00002fe4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00002f8f	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 40 00 00	.....@..
0000300c	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00003020	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003034	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003048	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
0000305c	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003070	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003084	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003098	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
000030ac	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
000030c0	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
000030d4	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
000030e8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000030cf	00 00 00 00 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	....AAAAAAAAAAAA
00003110	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003124	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003138	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
0000314c	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003160	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003174	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00003188	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
0000319c	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
000031b0	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
000031c4	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAGCC: (Ubuntu 9.3
000031d8	2E 30 2D 31 37 75 62 75 6E 74 75 31 7E 32 30 2E 30 34 29 20	.0-17ubuntu1~20.04)
000031ec	39 2E 33 2E 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	9.3.0.....

Signed 8 bit: 71      Signed 32 bit: 1195590458      Hexadecimal: 47 43 43 3A  
 Unsigned 8 bit: 71      Unsigned 32 bit: 1195590458      Decimal: 071 067 067 058  
 Signed 16 bit: 18243      Float 32 bit: 49987.23      Octal: 107 103 103 072  
 Unsigned 16 bit: 18243      Float 64 bit: 2.00034333882626E+35      Binary: 01000111 01000011 0100001  
☐ Show little endian decoding      ☐ Show unsigned as hexadecimal      ASCII Text: GCC:  
 Offset: 0x31c8 / 0x431f      Selection: 0x3100 to 0x31c7 (0xc8 bytes) INS

```

seed@VM: ~/Desktop
[05/24/24] seed@VM:~/Desktop$ ./s; ./k
benign code!
malicious code!
[05/24/24] seed@VM:~/Desktop$ md5sum s; md5sum k
e33cf0cf0d4805ad3a5e5a42484d49bc s
e33cf0cf0d4805ad3a5e5a42484d49bc k
[05/24/24] seed@VM:~/Desktop$
  
```

Executing both codes (s and k) returns different output, s returns benign code, k returns malicious code. However, the hash value for s and k are the same.