# APPLIED CRYPTOGRAPHY

NAME: VISHWAS M

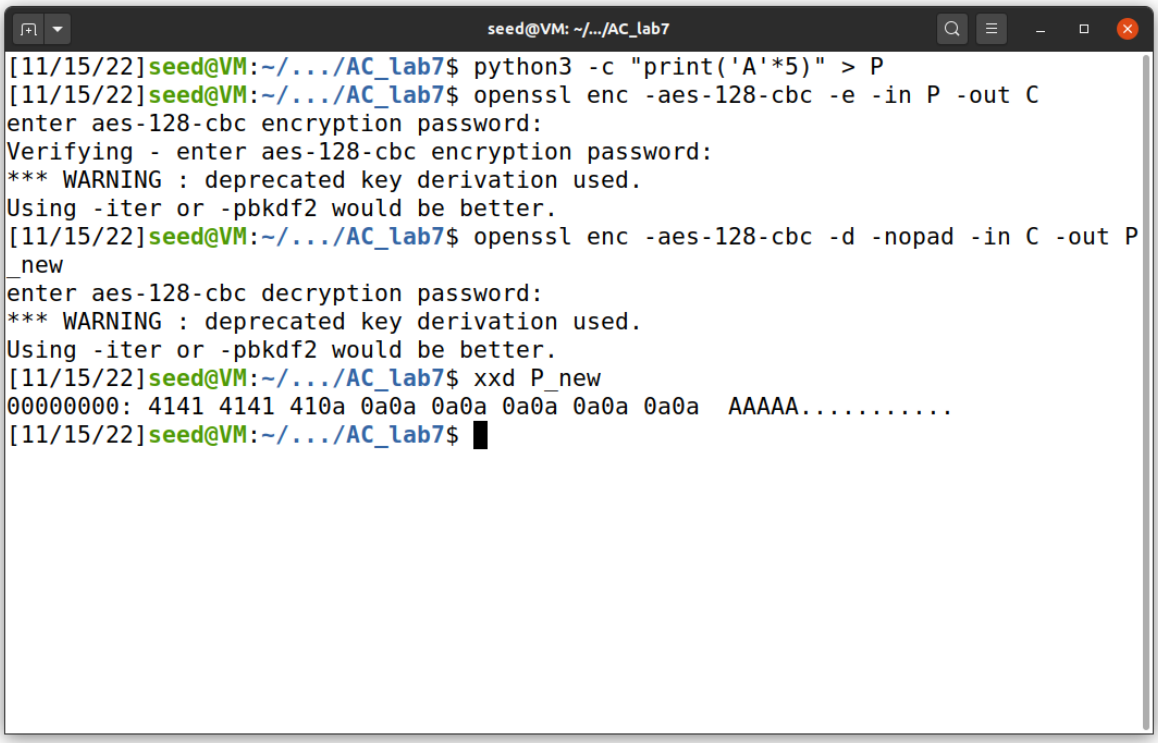
SRN: PES2UG20CS390

SEC: F

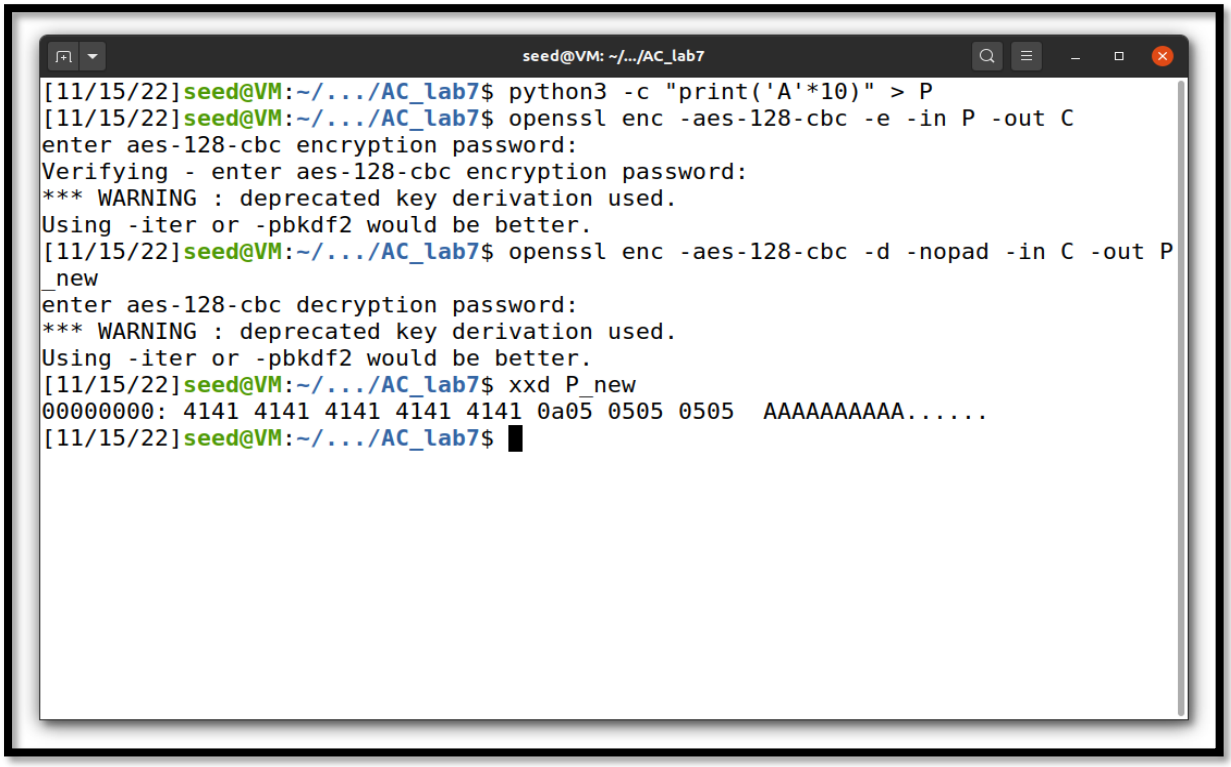
LAB: 7

TASK 1: Getting Familiar with Padding

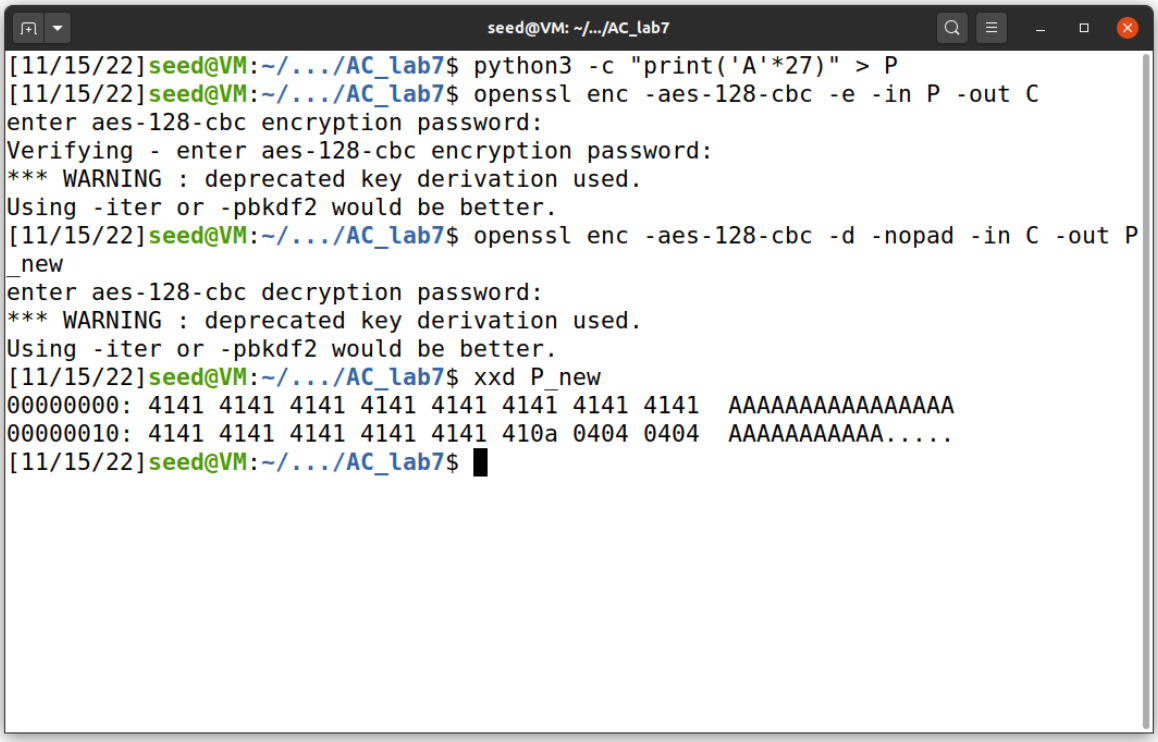
a) Length 5:



b) Length 10:


seed@VM: ~/.../AC_lab7
[11/15/22]seed@VM:~/.../AC_lab7$ python3 -c "print('A'*10)" > P
[11/15/22]seed@VM:~/.../AC_lab7$ openssl enc -aes-128-cbc -e -in P -out C
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/15/22]seed@VM:~/.../AC_lab7$ openssl enc -aes-128-cbc -d -nopad -in C -out P
_new
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/15/22]seed@VM:~/.../AC_lab7$ xxd P_new
00000000: 4141 4141 4141 4141 4141 0a05 0505 0505  AAAAAAAAAA......
[11/15/22]seed@VM:~/.../AC_lab7$

c) Length 27:

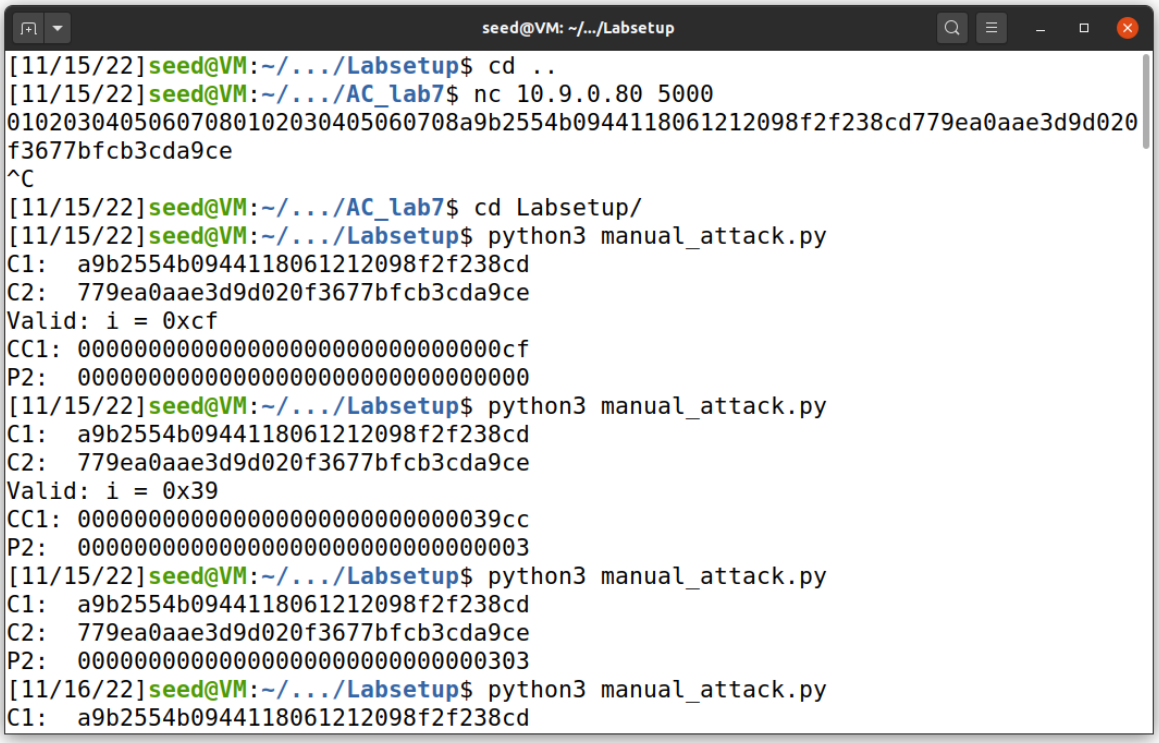


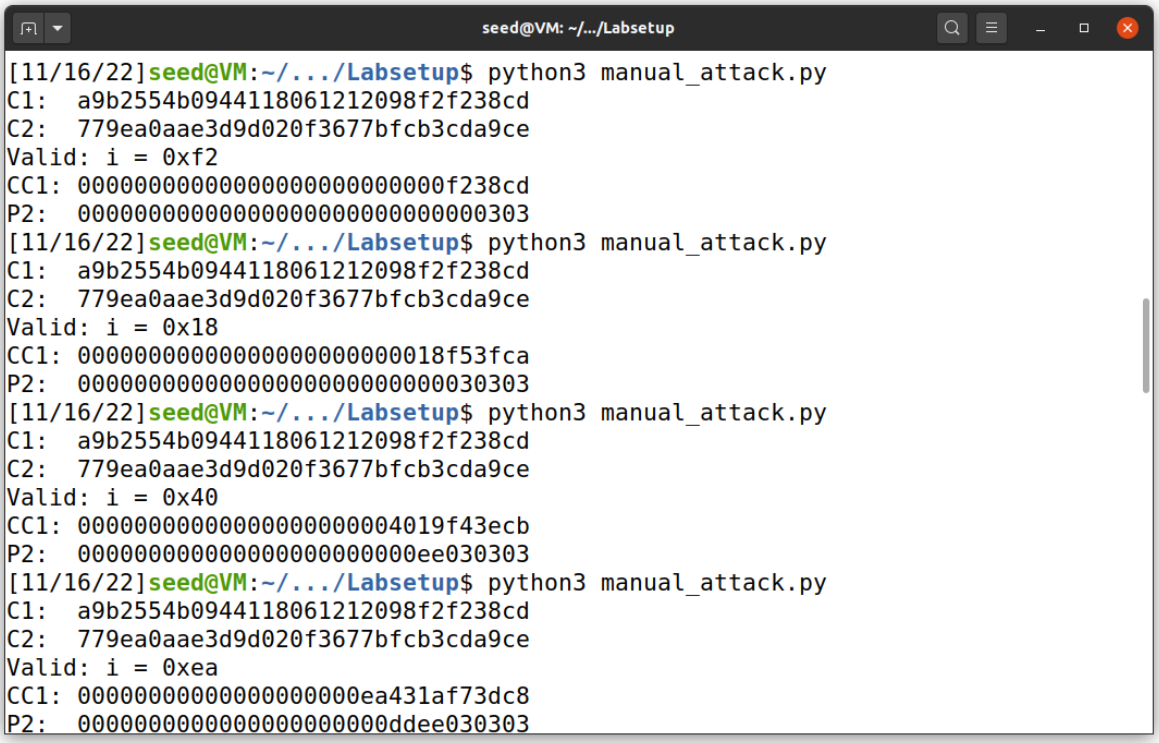

What do you deduce about the encryption scheme?

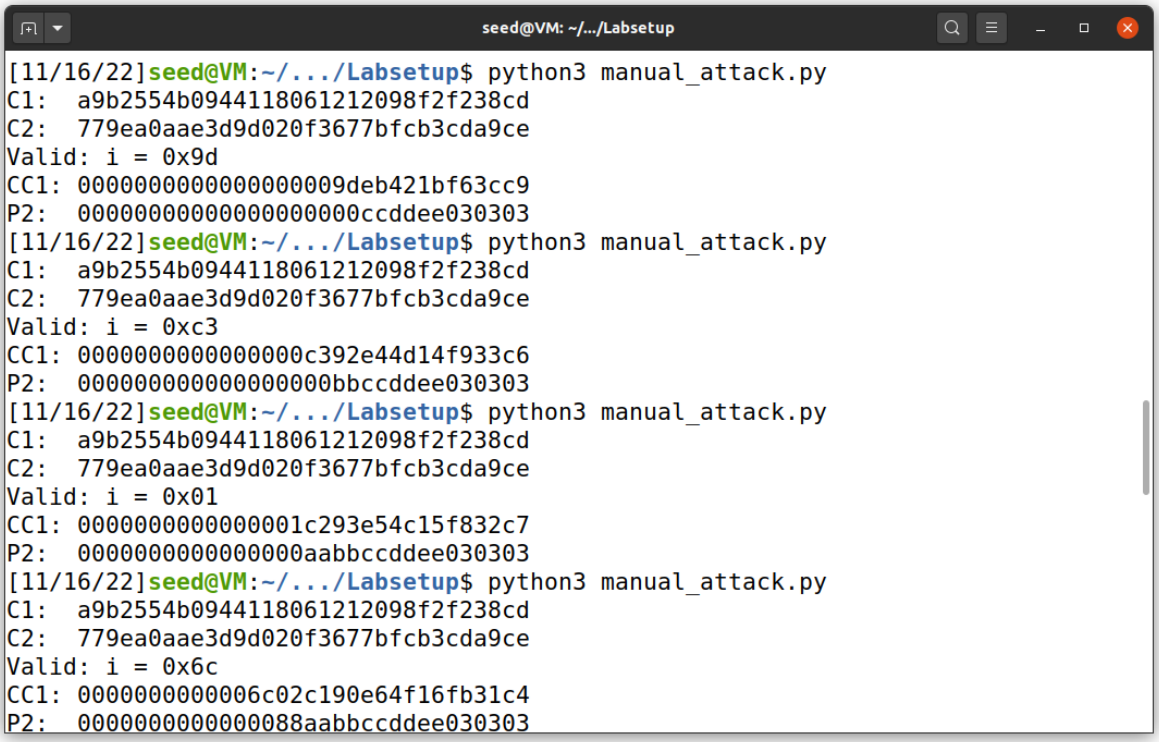
ANS: Padding is used to decrypt the ciphertext

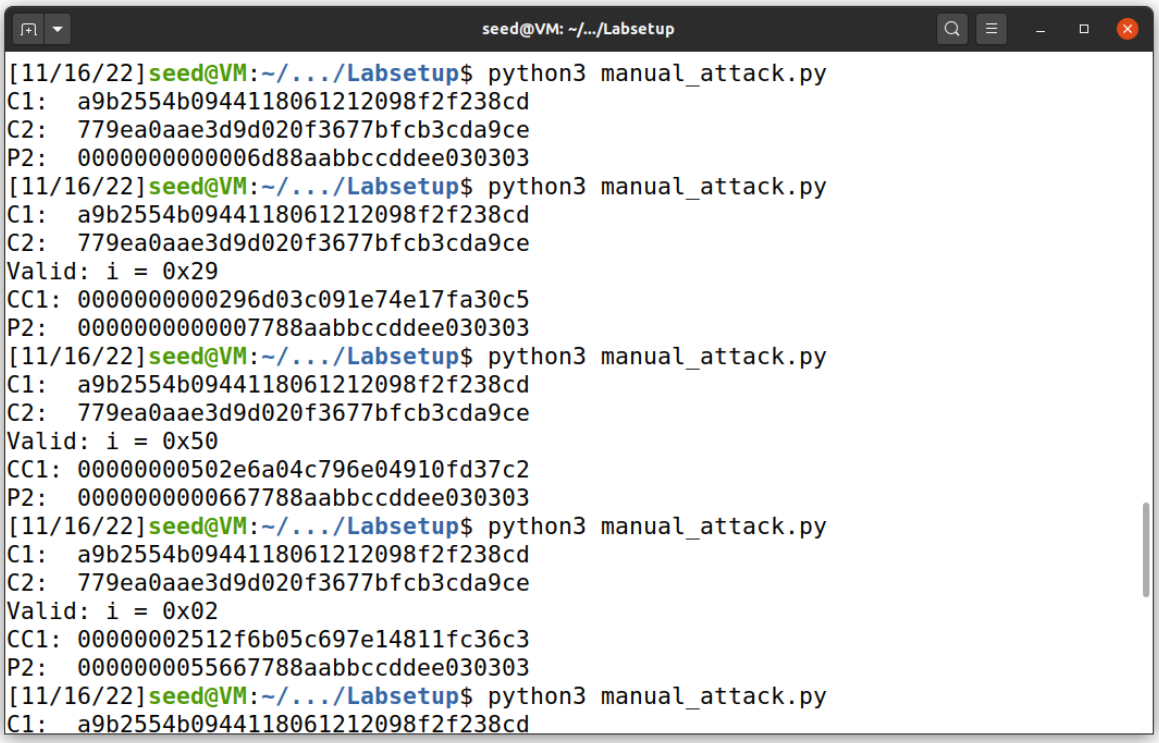
# Task 2: Padding Oracle Attack

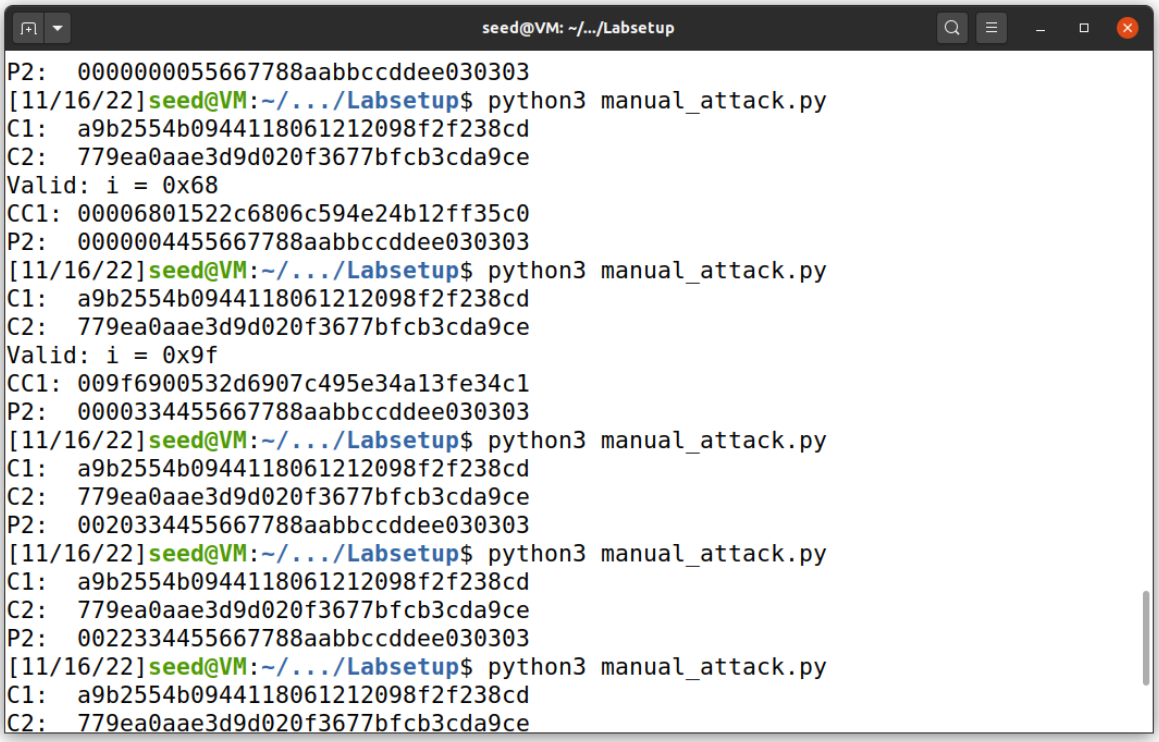
## Step 1 – Step 3 for 16 rounds

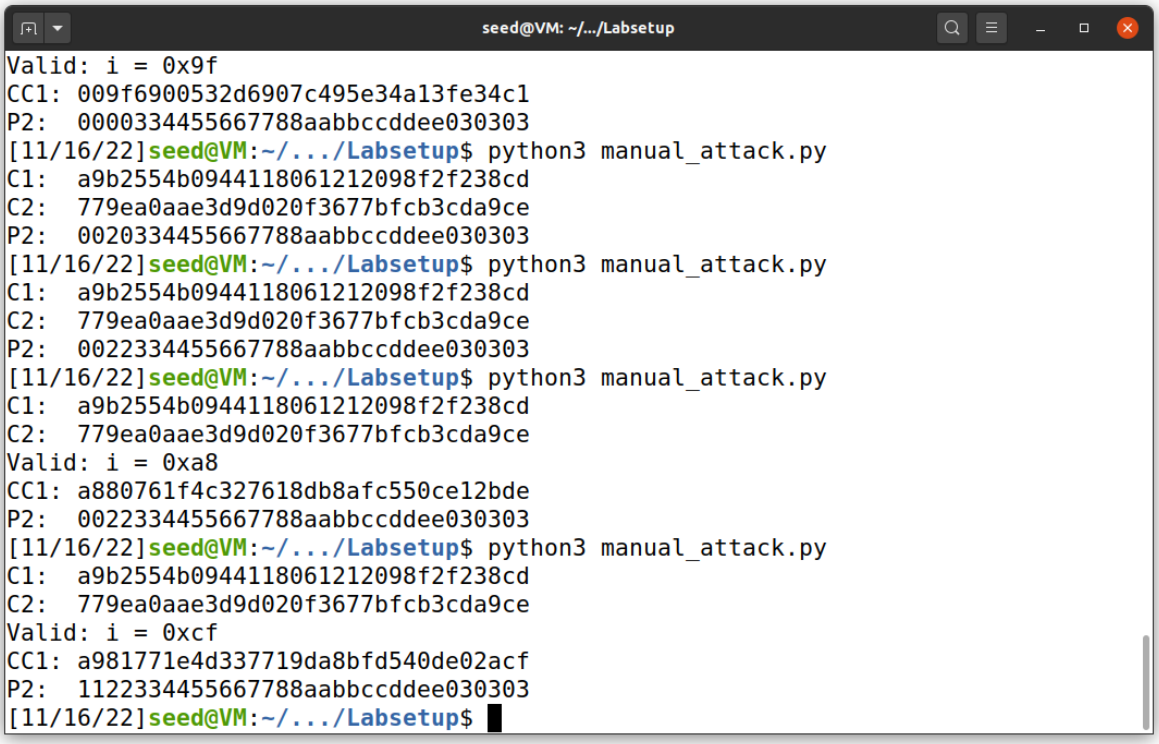


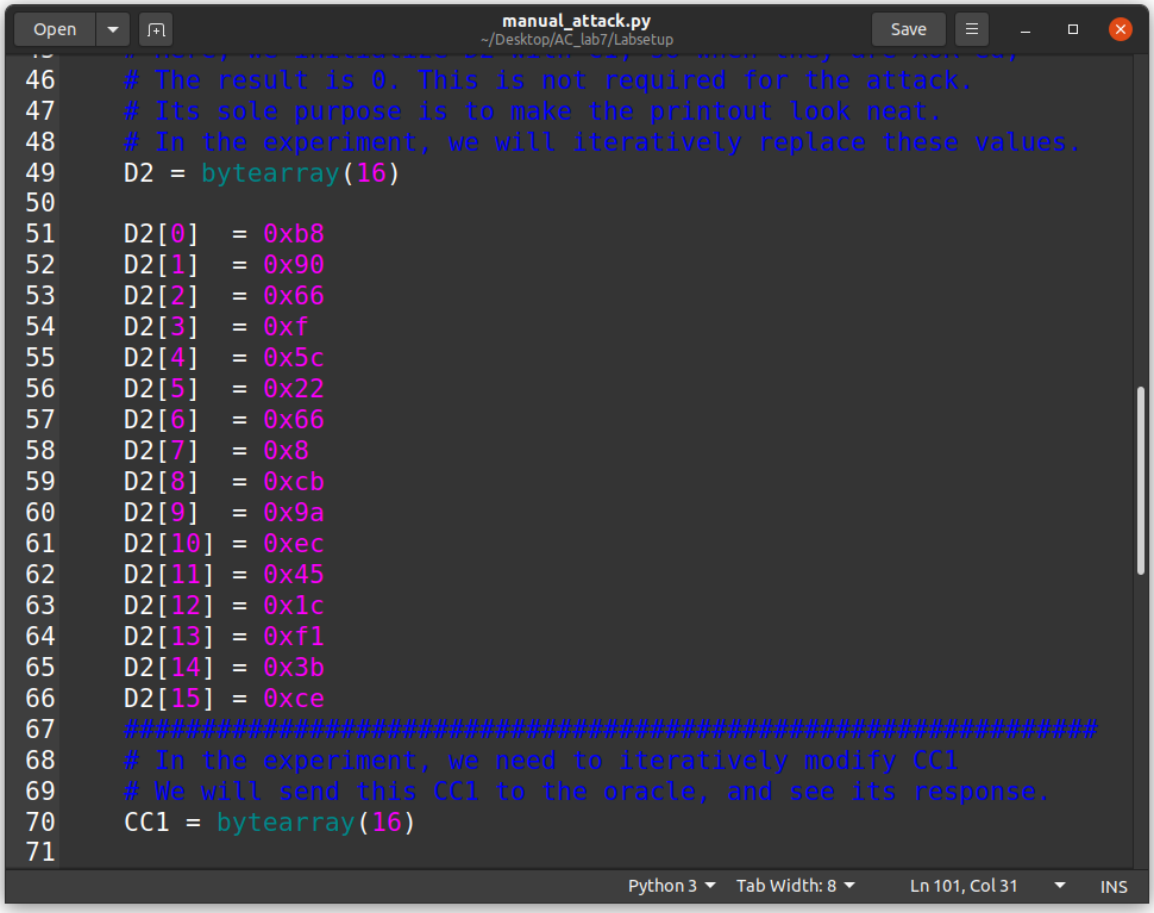

seed@VM: ~/.../Labsetup
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0x9d
CC1: 000000000000000009deb421bf63cc9
P2:  0000000000000000000ccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0xc3
CC1: 0000000000000000c392e44d14f933c6
P2:  0000000000000000000bbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0x01
CC1: 0000000000000001c293e54c15f832c7
P2:  0000000000000000aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0x6c
CC1: 0000000000006c02c190e64f16fb31c4
P2:  0000000000000088aabbccddee030303

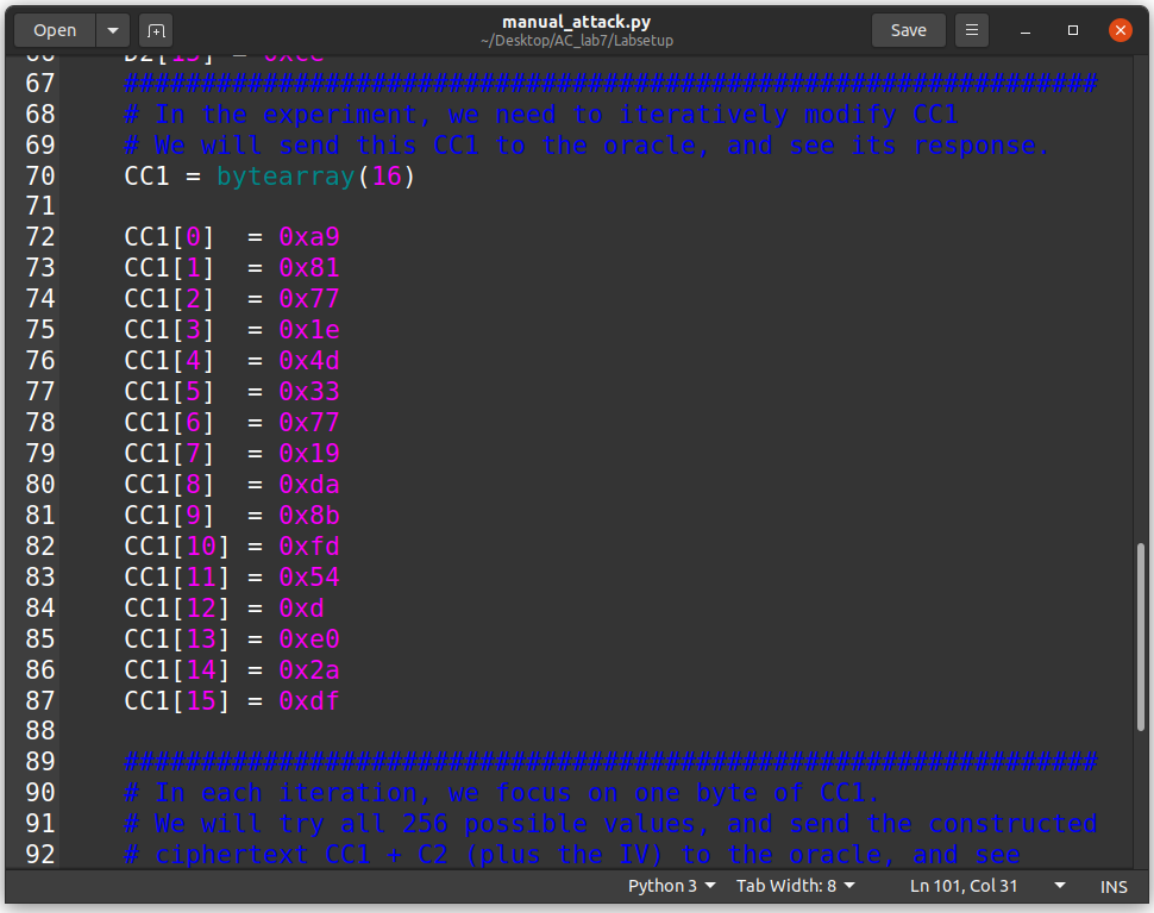

seed@VM: ~/.../Labsetup
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
P2:  0000000000006d88aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0x29
CC1: 0000000000296d03c091e74e17fa30c5
P2:  0000000000007788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0x50
CC1: 00000000502e6a04c796e04910fd37c2
P2:  0000000000667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0x02
CC1: 00000002512f6b05c697e14811fc36c3
P2:  0000000055667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd


seed@VM: ~/.../Labsetup
P2:  0000000055667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0x68
CC1: 00006801522c6806c594e24b12ff35c0
P2:  0000004455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0x9f
CC1: 009f6900532d6907c495e34a13fe34c1
P2:  0000334455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
P2:  0020334455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
P2:  0022334455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce

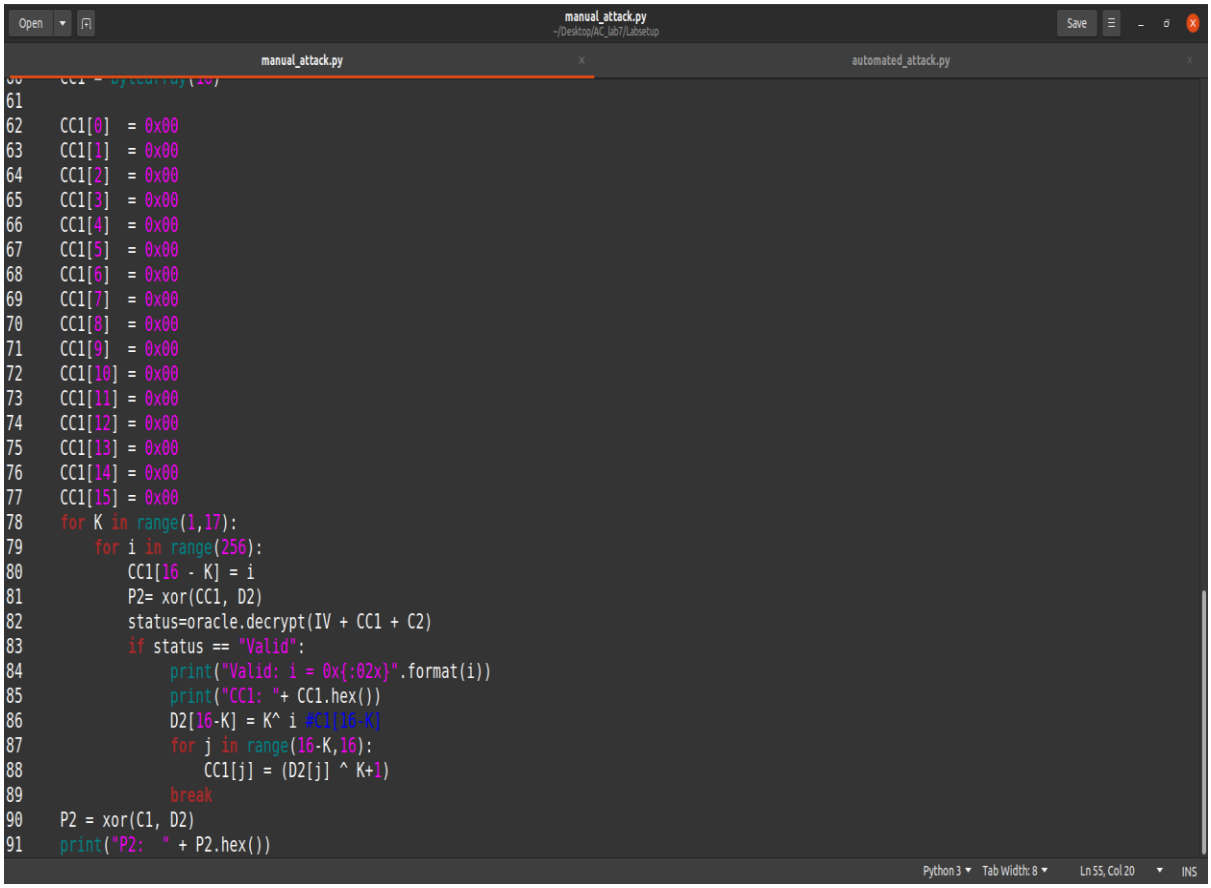

seed@VM: ~/.../Labsetup
Valid: i = 0x9f
CC1: 009f6900532d6907c495e34a13fe34c1
P2:  0000334455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
P2:  0020334455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
P2:  0022334455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0xa8
CC1: a880761f4c327618db8afc550ce12bde
P2:  0022334455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  a9b2554b09441180612120 98f2f238cd
C2:  779ea0aae3d9d020f3677bfcb3cda9ce
Valid: i = 0xcf
CC1: a981771e4d337719da8bfd540de02acf
P2:  1122334455667788aabbccddee030303
[11/16/22]seed@VM:~/.../Labsetup$

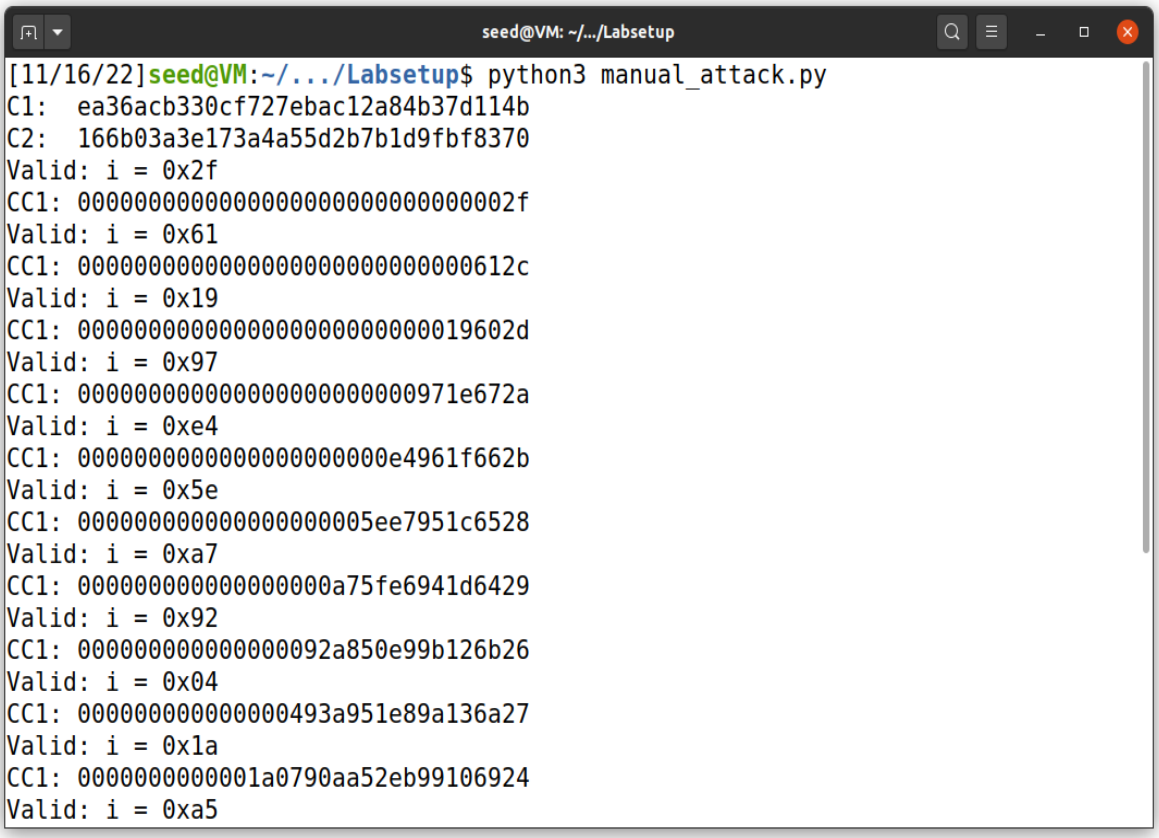
CODE:

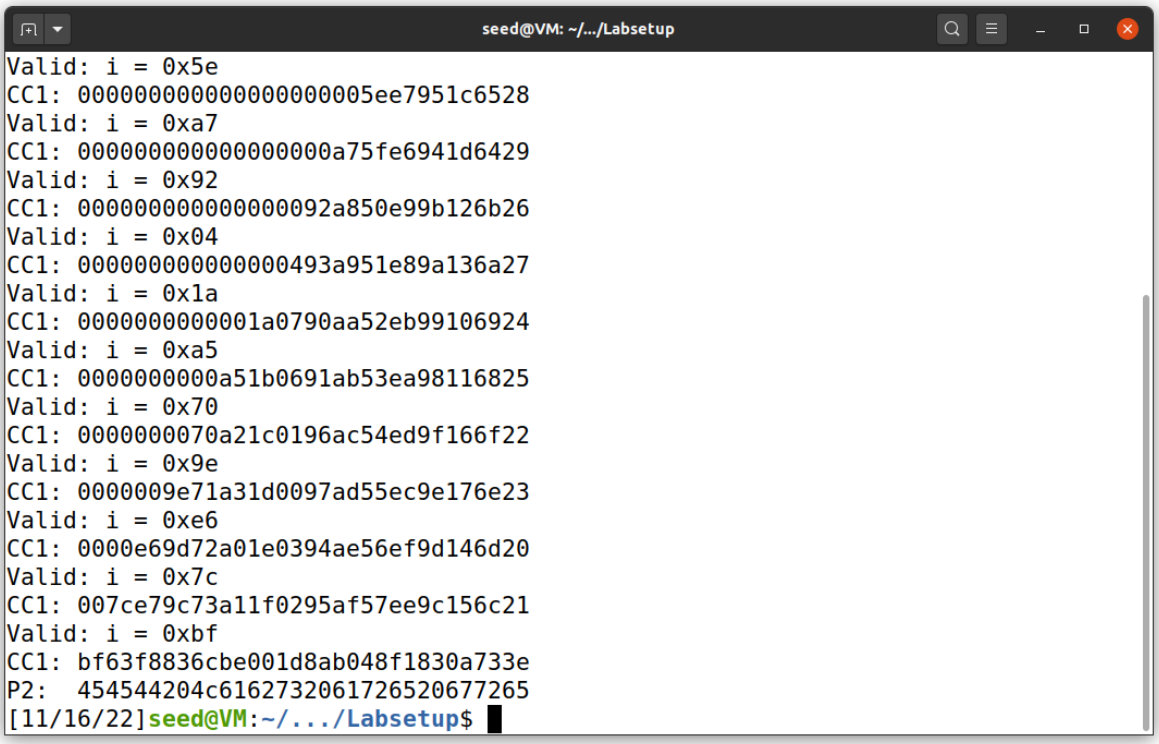

Open
manual_attack.py
~/Desktop/AC_lab7/Labsetup
Save
# The result is 0. This is not required for the attack.
# Its sole purpose is to make the printout look neat.
# In the experiment, we will iteratively replace these values.
D2 = bytearray(16)
D2[0] = 0xb8
D2[1] = 0x90
D2[2] = 0x66
D2[3] = 0xf
D2[4] = 0x5c
D2[5] = 0x22
D2[6] = 0x66
D2[7] = 0x8
D2[8] = 0xcb
D2[9] = 0x9a
D2[10] = 0xec
D2[11] = 0x45
D2[12] = 0x1c
D2[13] = 0xf1
D2[14] = 0x3b
D2[15] = 0xce
# In the experiment, we need to iteratively modify CC1
# We will send this CC1 to the oracle, and see its response.
CC1 = bytearray(16)
Python 3
Tab Width: 8
Ln 101, Col 31
INS


Open
manual_attack.py
~/Desktop/AC_lab7/Labsetup
Save
67 ##################################################################
68 # In the experiment, we need to iteratively modify CC1
69 # We will send this CC1 to the oracle, and see its response.
70 CC1 = bytearray(16)
71
72 CC1[0] = 0xa9
73 CC1[1] = 0x81
74 CC1[2] = 0x77
75 CC1[3] = 0x1e
76 CC1[4] = 0x4d
77 CC1[5] = 0x33
78 CC1[6] = 0x77
79 CC1[7] = 0x19
80 CC1[8] = 0xda
81 CC1[9] = 0x8b
82 CC1[10] = 0xfd
83 CC1[11] = 0x54
84 CC1[12] = 0xd
85 CC1[13] = 0xe0
86 CC1[14] = 0x2a
87 CC1[15] = 0xdf
88
89 ##################################################################
90 # In each iteration, we focus on one byte of CC1.
91 # We will try all 256 possible values, and send the constructed
92 # ciphertext CC1 + C2 (plus the IV) to the oracle, and see
Python 3
Tab Width: 8
Ln 101, Col 31
INS

# Task 3: Padding Oracle Attack(level 2)




seed@VM: ~/.../Labsetup
[11/16/22]seed@VM:~/.../Labsetup$ python3 manual_attack.py
C1:  ea36acb330cf727ebac12a84b37d114b
C2:  166b03a3e173a4a55d2b7b1d9fbf8370
Valid: i = 0x2f
CC1: 0000000000000000000000000000002f
Valid: i = 0x61
CC1: 0000000000000000000000000000612c
Valid: i = 0x19
CC1: 0000000000000000000000000019602d
Valid: i = 0x97
CC1: 0000000000000000000000000971e672a
Valid: i = 0xe4
CC1: 0000000000000000000000e4961f662b
Valid: i = 0x5e
CC1: 0000000000000000000005ee7951c6528
Valid: i = 0xa7
CC1: 0000000000000000a75fe6941d6429
Valid: i = 0x92
CC1: 0000000000000000092a850e99b126b26
Valid: i = 0x04
CC1: 0000000000000000493a951e89a136a27
Valid: i = 0x1a
CC1: 0000000000001a0790aa52eb99106924
Valid: i = 0xa5


seed@VM: ~/.../Labsetup
Valid: i = 0x5e
CC1: 0000000000000000000005ee7951c6528
Valid: i = 0xa7
CC1: 0000000000000000a75fe6941d6429
Valid: i = 0x92
CC1: 0000000000000000092a850e99b126b26
Valid: i = 0x04
CC1: 0000000000000000493a951e89a136a27
Valid: i = 0x1a
CC1: 0000000000001a0790aa52eb99106924
Valid: i = 0xa5
CC1: 0000000000a51b0691ab53ea98116825
Valid: i = 0x70
CC1: 0000000070a21c0196ac54ed9f166f22
Valid: i = 0x9e
CC1: 0000009e71a31d0097ad55ec9e176e23
Valid: i = 0xe6
CC1: 0000e69d72a01e0394ae56ef9d146d20
Valid: i = 0x7c
CC1: 007ce79c73a11f0295af57ee9c156c21
Valid: i = 0xbf
CC1: bf63f8836cbe001d8ab048f1830a733e
P2:  454544204c6162732061726520677265
[11/16/22]seed@VM:~/.../Labsetup$