Student Name: Jonathan Riordan

Student ID: C13432152

**Part 1**

Output

Plain text: AAAABBBBCCCCDDDDAA

32

Cipher text: 43d3215c92a75a1478fcf9cb950d20dbcb92e17929013822cf6dde42d50346b8

64

Decrypted: AAAABBBBCCCCDDDDAA00000000000001

Decrypted with no padding: AAAABBBBCCCCDDDDAA

Key: 1234567812345678

Code:

import base64

from Crypto.Cipher import AES

print "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Part 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

def addPadding(plaintext, length):

i = 0

j = 0

one = "1"

zero = "0"

block\_size = 32

temp = block\_size - 1

numOfPadding = block\_size - length

while i < block\_size:

if i > length:

plaintext += "0"

if i == block\_size - 1:

plaintext += "1"

i += 1

return plaintext

def removePadding(cipher, len):

i = 0

message = ""

block\_size = 16

numOfPadding = block\_size - len

length = block\_size - numOfPadding

while i < length:

message += cipher[i]

i += 1

return message

key = '1234567812345678'

plainText = "AAAABBBBCCCCDDDDAA"

ciphertext = ""

encoded = ""

length = len(plainText)

print "Plain text: " + plainText

ciphertext = addPadding(plainText, length)

obj = AES.new(key, AES.MODE\_ECB)

print len(ciphertext)

cp = obj.encrypt(ciphertext)

cp = cp.encode("hex")

print "Cipher text: " + cp

print len(cp)

dec = cp.decode("hex")

obj = AES.new(key, AES.MODE\_ECB)

dec = obj.decrypt(dec)

print "Decrypted: " + dec

decrypt = removePadding(dec,length)

print "Decrypted with no padding: " + decrypt

**Part 2.**

Plaintext: AAAABBBBCCCCDDDDAA

Cipher text: 43D3215C92A75A1478FCF9CB950D20DB502A485FD5735486D57AEA9AA809E3DD

Key: 1234567812345678

Code:

print "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Part 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

file = open("dictionary.txt", "r")

dictionary = []

for line in file:

if (len(line) >= 17):

dictionary.append(str(line[:-1]))

else:

dictionary.append(str(line))

file.close()

cipherPart2 = "43D3215C92A75A1478FCF9CB950D20DB502A485FD5735486D57AEA9AA809E3DD"

for i in dictionary:

key = i;

dec = cipherPart2.decode("hex")

obj2 = AES.new(i, AES.MODE\_ECB)

dec = obj2.decrypt(dec)

print removePadding(dec,len(plainText) )

In the dictionary file is

1234567812345611

1234567812345672

1234567812345623

1234567812345346

1234567812345678

To do brute force, each key is tested in the decryption cipher. As seen from the results below, the correct plaintext is the last one, therefore the key related to this is the last key in the dictionary file.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Part 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

í\_E4èQ–Xzª2û™˚ù¡

4æ≠Î\*yÊWö¬¥›È–(U

vdô9˛0eäÚ⁄“3—∞¥úYŒ

ÜUSd”é¢w

'UÎ<Ó ö6π

AAAABBBBCCCCDDDDAA