Student Name: Jonathan Riordan

Student ID: C13432152

Part 1

Output of code is:

Length of plaintext: 16

Key: 12345678

Cipher text: 19ff4637bb2fe77c19ff4637bb2fe77c

Plaintext: AAAABBBBAAAABBBB

Code

'''

Advanced Security

Student Name: Jonathan Riordan

Student ID: C13432152

'''

from Crypto.Cipher import DES

key = '12345678'

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

message = "AAAABBBBAAAABBBB"

print "Length of plaintext: " + str(len(message))

ciphertext = obj.encrypt(message)

cp = ciphertext.encode("hex")

print "Key: " + key

print "Cipher text: " + cp

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

print "Plaintext: " +obj.decrypt(ciphertext)

Part 2

Output of code:

Length of plaintext: 16

Plaintext: AAAABBBBAAAABBBB

Cipher Text: aac823f6bbe58f9eaf1fe0eb9ca7eb08

Decrypted: AAAABBBBAAAABBBB

Code:

'''

Advanced Security

Student Name: Jonathan Riordan

Student ID: C13432152

Lab 3 - Part 2

Output

Length of plaintext: 16

Plaintext: AAAABBBBAAAABBBB

Cipher Text: aac823f6bbe58f9eaf1fe0eb9ca7eb08

Decrypted: AAAABBBBAAAABBBB

'''

from Crypto.Cipher import DES

key = '12345678'

message = "AAAABBBBAAAABBBB"

iv = '00000000'

obj = DES.new(key, DES.MODE\_CBC,iv)

print "Length of plaintext: " + str(len(message))

ciphertext = obj.encrypt(message)

cp = ciphertext.encode("hex")

print "Plaintext: " + message

print "Cipher Text: " + cp

obj = DES.new(key, DES.MODE\_CBC,iv)

print "Decrypted: " + obj.decrypt(ciphertext)

Part 3

Output of code:

Plain text: ABCDEFGHIJK

16

Cipher text: 96de603eaed6256f20f2c43eb7cd8b67

32

Decrypted with padding: ABCDEFGHIJK00001

Decrypted: ABCDEFGHIJK

Code:

'''

Advanced Security

Student Name: Jonathan Riordan

Student ID: C13432152

Lab 3 - Part 3

Output

Plain text: ABCDEFGHIJK

16

Cipher text: 96de603eaed6256f20f2c43eb7cd8b67

32

Decrypted with padding: ABCDEFGHIJK00001

Decrypted: ABCDEFGHIJK

'''

import base64

from Crypto.Cipher import DES

def addPadding(plaintext, length):

i = 0

j = 0

one = "1"

zero = "0"

block\_size = 16

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 = '12345678'

plainText = "ABCDEFGHIJK"

ciphertext = ""

encoded = ""

length = len(plainText)

print "Plain text: " + plainText

ciphertext = addPadding(plainText, length)

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

dec = obj.decrypt(dec)

print "Decrypted with padding: " + dec

decrypt = removePadding(dec,length)

print "Decrypted: " + decrypt