


Symmetric Key

Objective: The key objective of this lab is to investigate the basics of symmetric key encryption.

Python Coding (Encrypting)

In this part of the lab, we will investigate the usage of Python code to perform different padding methods and using AES. First download the code from:

 **Web link (Cipher code):** <http://asecuritysite.com/cipher01.zip>

The code should be:

```
from Crypto.Cipher import AES
import hashlib
import sys
import binascii
import Padding

val='hello'
password='hello'

plaintext=val

def encrypt(plaintext,key, mode):
    encobj = AES.new(key,mode)
    return(encobj.encrypt(plaintext))

def decrypt(ciphertext,key, mode):
    encobj = AES.new(key,mode)
    return(encobj.decrypt(ciphertext))

key = hashlib.sha256(password).digest()

plaintext = Padding.appendPadding(plaintext,blocksize=Padding.AES_blocksize,mode='CMS')
print "After padding (CMS): "+binascii.hexlify(bytearray(plaintext))

ciphertext = encrypt(plaintext,key,AES.MODE_ECB)
print "Cipher (ECB): "+binascii.hexlify(bytearray(ciphertext))

plaintext = decrypt(ciphertext,key,AES.MODE_ECB)
plaintext = Padding.removePadding(plaintext,mode='CMS')
print "  decrypt: "+plaintext

plaintext=val
```

Now update the code so that you can enter a string and the program will show the cipher text. The format will be something like:

`python cipher01.py hello mykey`

where “hello” is the plain text, and “mykey” is the key. A possible integration is:

```
import sys

if (len(sys.argv)>1):
    val=sys.argv[1]

if (len(sys.argv)>2):
    password=sys.argv[2]
```

Now determine the cipher text for the following (the first example has already been completed):

Message	Key	CMS Cipher
"hello"	"hello123"	0a7ec77951291795bac6690c9e7f4c0d
"inkwell"	"orange"	
"security"	"qwerty"	
"Africa"	"changeme"	

Now copy your code and modify it so that it implements **64-bit DES** and complete the table (Ref to: http://asecuritysite.com/encryption/padding_des):

Message	Key	CMS Cipher
"hello"	"hello123"	8f770898ddb9fb38
"inkwell"	"orange"	
"security"	"qwerty"	
"Africa"	"changeme"	

Now modify the code so that the user can enter the values from the keyboard, such as with:

```
cipher=raw_input('Enter cipher:')
password=raw_input('Enter password:')
```

Python Coding (Decrypting)

Now modify your coding for 256-bit AES ECB encryption, so that you can enter the cipher text, and an encryption key, and the code will decrypt to provide the result. You should use CMS for padding. With this, determine the plaintext for the following (note, all the plain text values are countries around the World):

CMS Cipher (256-bit AES ECB)	Key	Plain text
b436bd84d16db330359edebf49725c62	"hello"	
4bb2eb68fccd6187ef8738c40de12a6b	"ankle"	
029c4dd71cdae632ec33e2be7674cc14	"changeme"	
d8f11e13d25771e83898efdbad0e522c	"123456"	

Now modify your coding for 64-bit DES ECB encryption, so that you can enter the cipher text, and an encryption key, and the code will decrypt to provide the result. You should use CMS

for padding. With this, determine the plaintext for the following (note, all the plain text values are countries around the World):

CMS Cipher (64-bit DES ECB)	Key	Plain text
f37ee42f2267458d	“hello”	
67b7d1162394b868	“ankle”	
ac9feb702ba2ecc0	“changeme”	
de89513fbd17d0dc	“123456”	

Now update your program, so that it takes a cipher string in Base-64 and converts it to a hex string and then decrypts it. From this now decrypt the following Base-64 encoded cipher streams (which should give countries of the World):

CMS Cipher (256-bit AES ECB)	Key	Plain text
/vA6BD+ZXu8j6KrTHi1Y+w==	“hello”	
ni tTRpxMhGlaRkuyXWYxtA==	“ankle”	
i rwjGCAu+mmdNeu6Hq6ciw==	“changeme”	
5I71KpfT6RdM/xhUJ5IKCQ==	“123456”	

PS ... remember to add “import base64”.