**Laporan Penugasan ETS Kriptografi**

**Enkripsi dan Dekripsi**

**Dibuat Oleh : Kelompok 6**

Ian Felix Jonathan Simanjuntak - 05311940000008

Dian Arofati Nur Zamzamy - 05311940000011

Shavica Qurota Ayun Laksana - 05311940000013

Daniel Evan Yudhiputra, Tjioe - 05311940000016

Bryan Yehuda Mannuel - 05311940000021

**Tugas Yang Diberikan**

1. Melakukan enkripsi DES terhadap sebuah string/text
2. Melakukan enkripsi AES terhadap sebuah image
3. Terapkan dalam sistem Client Server
4. Buat Program dan juga dokumentasi terhadap Hasil pengerjaan tugas

**Gambaran Besar**

Untuk melakukan pengerjaan terhadap berbagai macam tugas yang diberikan untuk penilaian Tengah Semester dari Mata Kuliah Kriptografi, Kelompok kami berusaha untuk membuat sebuah program untuk melakukan enkripsi dan dekripsi terhadap text menggunakan algoritma DES dan juga melakukan enkripsi dan dekripsi terhadap gambar menggunakan algoritma AES. Kelompok kami juga menerapkan kedua program ini melalui sebuah hubungan Client Server dimana Client akan melakukan Request terhadap Server berupa text (untuk DES) dan image (untuk AES) dan nantinya Server akan memberikan Response berupa hasil enkripsi dari text ataupun image yang diminta.

Pertama-tama, kelompok kami melakukan pencarian referensi untuk berbagai macam Algoritma Enkripsi dan juga Dekripsi AES dan juga DES. Kelompok kami juga melakukan pencarian referensi bagaimana kita bisa melakukan hubungan Client Server dalam sebuah sistem Localhost. Kami juga mempelajari berbagai macam referensi yang diberikan oleh Ibu Annissa selaku Dosen pada Mata Kuliah Kriptografi ini untuk memperdalam pemahaman kami terhadap penugasan ini.

Setelah melakukan pencarian referensi, Kelompok kami memutuskan untuk membuat kode program dalam Bahasa Python karena dalam Bahasa ini kita bisa lebih leluasa menggunakan berbagai macam module dan library termasuk untuk melakukan hubungan Client Server. Kemudian kelompok kami membuat sebuah sistem TCP Client Server sederhana yang awalnya digunakan untuk melakukan pengiriman teks seperti chatting. Program Client Server ini kemudian kami modifikasi dengan melakukan penambahan fitur enkripsi dekripsi DES untuk Text dan juga enkripsi dekripsi AES untuk image yang nantinya akan dijelaskan lebih lanjut di bagian bawah.

# Client-Server AES untuk image

1. Cara kerja  
   Secara keseluruhan client-server yang kami buat menggunakan alur seperti pada diagram dibawah ini

|  |
| --- |
| **CLIENT**  **SERVER**  Mengirim ukuran image  ---------------------------------->  GOT SIZE  <----------------------------------  Mengirim image itu sendiri  ---------------------------------->  GOT IMAGE  <----------------------------------  BYE BYE  ---------------------------------->  server closes socket |

Pertama Client akan mengirimkan ukuran foto, apabila server sudah menerima ukuran foto, server akan mengirim pesan “GOT SIZE”, setelah itu client akan mengirimkan file foto, apabila berhasil server akan mengirim pesan “GOT IMAGE”, terakhir server akan mengirim pesan “BYE BYE” dan client akan mengirim close socket

1. Source code Client

|  |
| --- |
| #!/usr/bin/env python  import random import socket, select from time import gmtime, strftime from random import randint import time import os  #image = raw\_input("enter the name of the image file: ")  HOST = '127.0.0.1' PORT = 6662  sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) server\_address = (HOST, PORT) sock.connect(server\_address) command = int(input("Masukkan command :\n1.Encrpyt\n2.Decrypt\n"))  try:  if command==1 :  # open image  os.system('python3 enc.py')  time.sleep(10)  myfile = open('encrypted.enc', 'rb')  bytes = myfile.read()  size = len(bytes)  sock.send('1')  elif command==2 :  # open image  os.system('python3 dec.py')  time.sleep(10)  myfile = open('output.png', 'rb')  bytes = myfile.read()  size = len(bytes)  sock.send('2')   # send image size to server  sock.sendall("SIZE %s" % size)  answer = sock.recv(4096)   print 'answer = %s' % answer   # send image to server  if answer == 'GOT SIZE':  sock.sendall(bytes)   # check what server send  answer = sock.recv(4096)  print 'answer = %s' % answer   if answer == 'GOT IMAGE' :  sock.sendall("BYE BYE ")  print 'Image successfully send to server'   myfile.close()  finally:  sock.close() |

Berikut adalah penjelasan dari code diatas :

1. Library yang kami gunakan

|  |
| --- |
| import random import socket, select from time import gmtime, strftime from random import randint import time import os |

Random digunakan untuk generate random number, socket untuk membuat socket yang diperlukan dalam client-server, time digunakan untuk mendapatkan waktu, os digunakan untuk menjalankan perintah system, yang diperlukan untuk memanggil file encrypt dan decrypt

1. Create Socket

|  |
| --- |
| HOST = '127.0.0.1' PORT = 6662  sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) server\_address = (HOST, PORT) sock.connect(server\_address) |

Diatas merupakan fungsi-fungsi yang digunakan untuk membuat socket, sehingga antara client dan server dapat berkomunikasi

1. Mendapatkan mode

|  |
| --- |
| command = int(input("Masukkan command :\n1.Encrpyt\n2.Decrypt\n"))  try:  if command==1 :  # open image  os.system('python3 enc.py')  time.sleep(10)  myfile = open('encrypted.enc', 'rb')  bytes = myfile.read()  size = len(bytes)  sock.send('1')  elif command==2 :  # open image  os.system('python3 dec.py')  time.sleep(10)  myfile = open('output.png', 'rb')  bytes = myfile.read()  size = len(bytes)  sock.send('2') |

Pada code diatas, program akan mengambil input dari user yang kemudian akan ditentukan apakah user memilih encrypt atau decrypt dengan menggunakan if, jika encrypt maka program akan memanggil program encrpyt (“enc.py”), dan mengirimkan file yang sudah terenkripsi (“encrypted.enc”) ke server.

Namun jika user memilih decrypt maka program akan memanggil program decrypt (“dec.py”), dan mengirimkan file “output.png” ke server.

1. Mengirim image size dan file image

|  |
| --- |
| sock.sendall("SIZE %s" % size)  answer = sock.recv(4096)  print 'answer = %s' % answer  # send image to server if answer == 'GOT SIZE': sock.sendall(bytes)  # check what server send answer = sock.recv(4096) print 'answer = %s' % answer |

Code diatas berfungsi untuk mengirimkan size dari file ke server dan melakukan print feedback dari server yang berupa string “GOT SIZE”. Jika feedback nya adalah “GOT SIZE” maka program akan mengirim file image. Selanjutnya program akan mencetak feedback dari server.

1. Menutup Socket

|  |
| --- |
| if answer == 'GOT IMAGE' : sock.sendall("BYE BYE ") print 'Image successfully send to server'  myfile.close()  finally:  sock.close() |

Jika feedback dari server adalah “GOT IMAGE” maka socket akan ditutup.

1. Source code Server

|  |
| --- |
| #!/usr/bin/env python  import random import socket, select from time import gmtime, strftime from random import randint import os  imgcounter = 1   HOST = '127.0.0.1' PORT = 6662  connected\_clients\_sockets = []  server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  server\_socket.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1) server\_socket.bind((HOST, PORT)) server\_socket.listen(10)  connected\_clients\_sockets.append(server\_socket)  while True:   read\_sockets, write\_sockets, error\_sockets = select.select(connected\_clients\_sockets, [], [])   for sock in read\_sockets:  if sock == server\_socket:   sockfd, client\_address = server\_socket.accept()  connected\_clients\_sockets.append(sockfd)   else:  command = sock.recv(1)  if command=="1" :  basename = "image.enc"  elif command=="2" :  basename = "image.png"   try:  data = sock.recv(4096)  txt = str(data)   if data:   if data.startswith('SIZE'):  tmp = txt.split()  size = int(tmp[1])   print 'got size'   sock.sendall("GOT SIZE")   elif data.startswith('BYE'):  sock.shutdown()   else :   myfile = open(basename, 'wb')  myfile.write(data)   data = sock.recv(40960000)  if not data:  myfile.close()  break  myfile.write(data)  myfile.close()   sock.sendall("GOT IMAGE")  sock.shutdown()  except:  sock.close()  connected\_clients\_sockets.remove(sock)  continue  imgcounter += 1 server\_socket.close() |

Berikut adalah penjelasan dari code diatas :

1. Library yang kami gunakan

|  |
| --- |
| import random import socket, select from time import gmtime, strftime from random import randint import os |

Random digunakan untuk melakukan generate random number, socket untuk melakukan socket programming, time digunakan untuk mendapatkan waktu, os untuk memanggil os.system

1. Create Socket

|  |
| --- |
| HOST = '127.0.0.1' PORT = 6662  connected\_clients\_sockets = []  server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  server\_socket.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1) server\_socket.bind((HOST, PORT)) server\_socket.listen(10)  connected\_clients\_sockets.append(server\_socket)  while True:   read\_sockets, write\_sockets, error\_sockets = select.select(connected\_clients\_sockets, [], [])   for sock in read\_sockets:  if sock == server\_socket:   sockfd, client\_address = server\_socket.accept()  connected\_clients\_sockets.append(sockfd) |

Code diatas digunakan untuk membuat socket dan memeriksa apakah socket dapat di create atau tidak

1. Menerima command

|  |
| --- |
| command = sock.recv(1)  if command=="1" :  basename = "image.enc"  elif command=="2" :  basename = "image.png" |

Jika mode yang dipilih adalah enkripsi maka file akan disimpan dengan nama file “image.enc”. Namun, jika yang dipilih adalah dekripsi maka file akan disimpan dengan nama “image.png”.

1. Mendapatkan size image

|  |
| --- |
| try:  data = sock.recv(4096)  txt = str(data)   if data:   if data.startswith('SIZE'):  tmp = txt.split()  size = int(tmp[1])   print 'got size'   sock.sendall("GOT SIZE") |

Jika server mendapatkan size image maka akan mengirimkan “GOT SIZE” ke client.

1. Menutup Socket

|  |
| --- |
| except:  sock.close()  connected\_clients\_sockets.remove(sock)  continue  imgcounter += 1 server\_socket.close() |

Untuk menutup socket, digunakan socket.close().

**Enkripsi AES untuk image**

|  |
| --- |
| from Crypto.Cipher import AES from Crypto import Random  key = Random.new().read(AES.block\_size) iv = Random.new().read(AES.block\_size)  with open("tux.png",'rb') as f:  input\_data = f.read() #input\_file.close()  cfb\_cipher = AES.new(key, AES.MODE\_CFB, iv) enc\_data = cfb\_cipher.encrypt(input\_data)  enc\_file = open("encrypted.enc", "wb") enc\_file.write(enc\_data) #enc\_file.close() |

Berikut adalah penjelasan dari code diatas :

1. Library yang digunakan

|  |
| --- |
| from Crypto.Cipher import AES from Crypto import Random |

Library AES digunakan untuk mengenkripsi dengan algoritma AES, sedang Random untuk melakukan generate key.

1. Generate key

|  |
| --- |
| key = Random.new().read(AES.block\_size) iv = Random.new().read(AES.block\_size) |

Code diatas digunakan untuk melakukan generate key dengan random

1. Membuka file

|  |
| --- |
| with open("tux.png",'rb') as f:  input\_data = f.read() |

Code diatas digunakan untuk membuka gambar yang akan di enkripsi

1. Meng-enkripsi file

|  |
| --- |
| cfb\_cipher = AES.new(key, AES.MODE\_CFB, iv) enc\_data = cfb\_cipher.encrypt(input\_data) |

Code diatas digunakan untuk melakukan enkripsi dengan algoritma AES

1. Menyimpan hasil enkripsi di file lain

|  |
| --- |
| enc\_file = open("encrypted.enc", "wb") enc\_file.write(enc\_data) |

Code diatas digunakan untuk menyimpan hasil enkripsi di “encrypted.enc”.

**Dekripsi AES untuk image**

|  |
| --- |
| from Crypto.Cipher import AES from Crypto import Random  key = Random.new().read(AES.block\_size) iv = Random.new().read(AES.block\_size)  #enc\_file2 = open("encrypted.enc") #enc\_data2 = enc\_file2.read() #enc\_file2.close()  with open("image.enc",'rb') as f:  enc\_data2 = f.read()  cfb\_decipher = AES.new(key, AES.MODE\_CFB, iv) plain\_data = cfb\_decipher.decrwith open("image.enc",'rb') as f:  enc\_data2 = f.read()  output\_file = open("output.png", "wb") output\_file.write(plain\_data) output\_file.close() |

Berikut adalah penjelasan code diatas :

1. Library yang digunakan

|  |
| --- |
| from Crypto.Cipher import AES from Crypto import Random |

AES digunakan untuk mendekripsi dengan algoritma AES.

1. Membuka file yang telah terenkripsi

|  |
| --- |
| with open("image.enc",'rb') as f: enc\_data2 = f.read() |

Code diatas digunakan untuk membuka file yang telah terenkripsi

1. Melakukan dekripsi AES

|  |
| --- |
| cfb\_decipher = AES.new(key, AES.MODE\_CFB, iv) plain\_data = cfb\_decipher.decrwith open("image.enc",'rb') as f:  enc\_data2 = f.read() |

Code di atas digunakan untuk melakukan dekripsi AES.

1. Menyimpan hasil dekripsi di file gambar

|  |
| --- |
| output\_file = open("output.png", "wb") output\_file.write(plain\_data) output\_file.close() |

Code di atas digunakan untuk menyimpan hasil dekripsi ke “output.png”.

**Hasil Run AES untuk image**







