Lab Assignment One

Code for single DES

The DES function is called once and the key size is 64 bits.
This is a screenshot from “socket_client.py”

```
key = "AABB09182736CCDD"
DES.prepareKey(key)
toSend = ""
n = len(message)
for i in range(0, n, 16):
    toEnc = message[i:i+16]
    toEnc = DES.encrypt(toEnc, DES.rkb, DES.rk)
    toEnc = DES.bin2hex(toEnc)
    toSend = toSend + toEnc
```

Code for double DES

The DES function is called twice and 2 keys are used. The first DES uses the first key and the second DES uses the second key.
This is a screenshot from “socket_client_2des.py”

```
key = "AABB09182736CCDD"
DES.prepareKey(key)
toSend = ""
n = len(message)
for i in range(0, n, 16):
    toEnc = message[i:i+16]
    toEnc = DES.encrypt(toEnc, DES.rkb, DES.rk)
    toEnc = DES.bin2hex(toEnc)
    toSend = toSend + toEnc

message = toSend
key = "BABB09182736CCDD"
DES.prepareKey(key)
toSend = ""
n = len(message)
for i in range(0, n, 16):
    toEnc = message[i:i + 16]
    toEnc = DES.encrypt(toEnc, DES.rkb, DES.rk)
    toEnc = DES.bin2hex(toEnc)
    toSend = toSend + toEnc
```

Code for triple DES with 2 keys

The DES function is called thrice and 3 keys are used. The first DES uses the first key, the second DES is actually a reverse DES and it uses the second key, and the last DES uses the first key.

This is a screenshot from “socket_client_3des.py”

```
key = "AABB09182736CCDD"
DES.prepareKey(key)
toSend = ""
n = len(message)
for i in range(0, n, 16):
    toEnc = message[i:i+16]
    toEnc = DES.encrypt(toEnc, DES.rkb, DES.rk)
    toEnc = DES.bin2hex(toEnc)
    toSend = toSend + toEnc

message = toSend
key = "BABB09182736CCDD"
DES.prepareKey(key)
toSend = ""
n = len(message)
for i in range(0, n, 16):
    toEnc = message[i:i + 16]
    toEnc = DES.encrypt(toEnc, DES.rkb_rev, DES.rk_rev)
    toEnc = DES.bin2hex(toEnc)
    toSend = toSend + toEnc

message = toSend
key = "AABB09182736CCDD"
DES.prepareKey(key)
toSend = ""
n = len(message)
for i in range(0, n, 16):
    toEnc = message[i:i + 16]
    toEnc = DES.encrypt(toEnc, DES.rkb, DES.rk)
    toEnc = DES.bin2hex(toEnc)
    toSend = toSend + toEnc
```

Snapshots

Client using single DES snapshot:

```
C:\Users\Oscar\Desktop\DES chat>python socket_client.py
 > hello
 > this is the lab assignment 1
 > bye
 >
```

Server using single DES snapshot:

```
C:\Users\Oscar\Desktop\DES chat>python socket_server.py
Listening for connections on 127.0.0.1:1234...
Accepted new connection from 127.0.0.1:14458, username: client
Received message from client: 30D29F56DB54885D
Decrypted message: 68656C6C6F000000
Decoded message: hello
Received message from client: 8295AB753B1783951FB798CB29501046D69E0483253DA3E62E91CD0BF0CCF6AD

Decrypted message: 7468697320697320746865206C6162206173736967E6D656E74203100000000
Decoded message: this is the lab assignment 1
Received message from client: AAA444F04A224EE2
Decrypted message: 6279650000000000
Decoded message: bye
```

Client using double DES snapshot:

```
C:\Users\Oscar\Desktop\DES chat>python socket_client_2des.py
 > bye
```

Server using double DES snapshot:

```
C:\Users\Oscar\Desktop\DES chat>python socket_server_2des.py
Listening for connections on 127.0.0.1:1234...
Accepted new connection from 127.0.0.1:14522, username: client
Received message from client: C6FF8EA175A5E023
Middle text: AAA444F04A224EE2
Text after second DES: 6279650000000000
Decoded message: bye
```

Client using triple DES snapshot:

```
C:\Users\Oscar\Desktop\DES chat>python socket_client_3des.py
 > hello
```

Server using triple DES snapshot:

```
C:\Users\Oscar\Desktop\DES chat>python socket_server_3des.py
Listening for connections on 127.0.0.1:1234...
Accepted new connection from 127.0.0.1:14483, username: client
Received message from client: 2F537F7FE64461C2
Message after first reverse DES: 7FA687403267A67D
Message after DES with k2: 30D29F56DB54885D
Message after second reverse DES: 68656C6C6F000000
Decoded message: hello
```