FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ 2010

BLM442E Homework

Yasemin Maya Kara

1521221114

Summary

in this application a simple peer to peer (P2P) messaging system with several security features using python programming language has been implemented.

Handshaking and key generation:

when users connect to server they are asked to enter other user id to communicate with. Server can handle multiple users pairs at the same time. after both users enter id number of each others each client generate public and private key pairs using RSA algorithm and send public key to other client. Here we assume that public key is checked by server before sending to other side (public key certification is not implemented) then they generate a random nonce and send it with encrypted version using their own private key so when client receive encrypted nonce he decrypts it with other client public key and compare it with received nonce to verify it.

Message Encryption : if nonce is verified client can ensure identity of other client because it is encrypted with its private key (authentication) and reply attack also is prevented because clients can not generate symmetric key and start chatting before nonce is verified so if nonce is changed a new symmetric key will be generated. After that clients can safely exchange their symmetric key that encrypted with other client public key so no one else except cleint2 will be able to decrypts it even the server that exchange messages between them is not able to make encryption, after this steps the session starts and each sent message is encrypted with other client symmetric key using AES algorithm in CBC mod and received messages is decrypted with their own symmetric key so clients can chatting in secure way . sending and receiving messages is implemented in different threads so codes are not blocking each others.

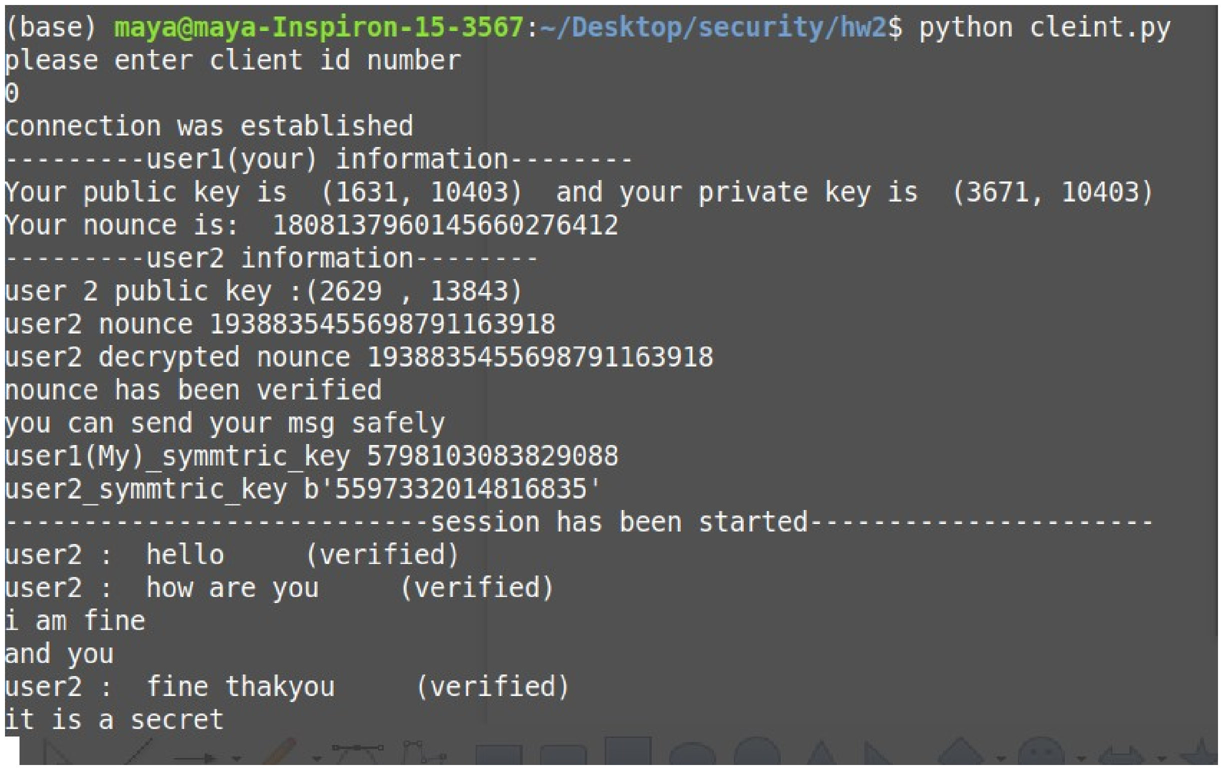
Integrity Check message integrity is checked with HMAC that sent with each encrypted message and verified with each decrypted message

more details are written as a comments in codes

Screenshots

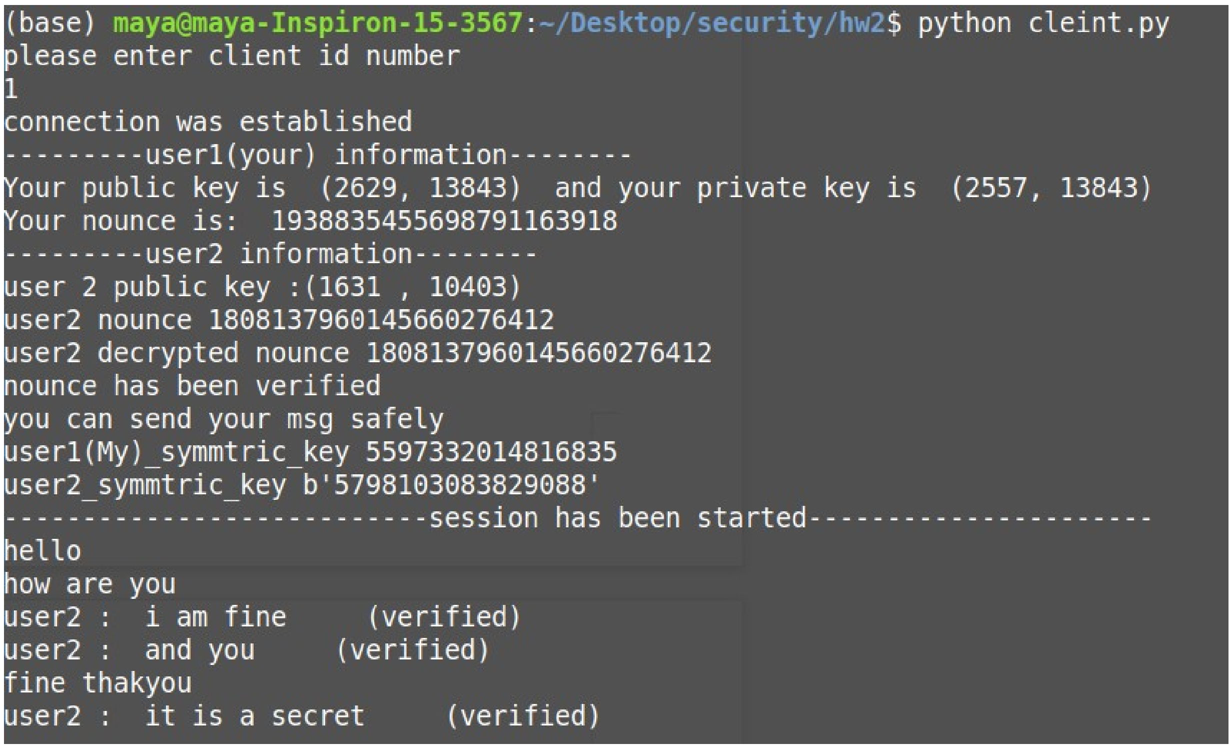
user enter only other cleint id to start communication then if nonce is verified session starts and client can start send messages. steps between connection was established and session has been started is checked automatically without user interaction

user1:



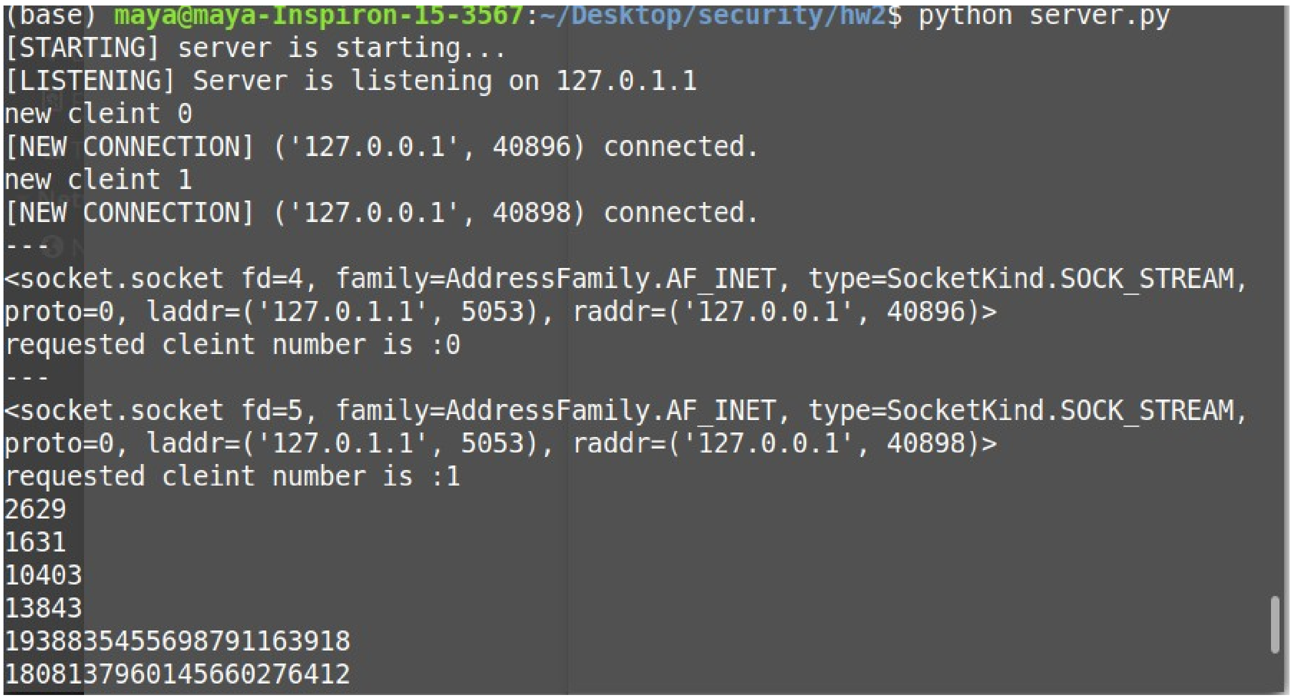
```
(base) maya@maya-Inspiron-15-3567:~/Desktop/security/hw2$ python cleint.py
please enter client id number
0
connection was established
---------user1(your) information--------
Your public key is  (1631, 10403)  and your private key is  (3671, 10403)
Your nounce is:  1808137960145660276412
---------user2 information--------
user 2 public key :(2629 , 13843)
user2 nounce 1938835455698791163918
user2 decrypted nounce 1938835455698791163918
nounce has been verified
you can send your msg safely
user1(My)_symmtric_key 5798103083829088
user2_symmtric_key b'5597332014816835'
---------------------------session has been started----------------------
user2 :  hello     (verified)
user2 :  how are you     (verified)
i am fine
and you
user2 :  fine thakyou     (verified)
it is a secret
```

user 2



```
(base) maya@maya-Inspiron-15-3567:~/Desktop/security/hw2$ python cleint.py
please enter client id number
1
connection was established
---------user1(your) information--------
Your public key is  (2629, 13843)  and your private key is  (2557, 13843)
Your nounce is:  1938835455698791163918
---------user2 information--------
user 2 public key :(1631 , 10403)
user2 nounce 1808137960145660276412
user2 decrypted nounce 1808137960145660276412
nounce has been verified
you can send your msg safely
user1(My)_symmtric_key 5597332014816835
user2_symmtric_key b'5798103083829088'
---------------------------session has been started----------------------
hello
how are you
user2 :  i am fine     (verified)
user2 :  and you     (verified)
fine thakyou
user2 :  it is a secret     (verified)
```

server except cleints and send encrypted messages between cleints



```
(base) maya@maya-Inspiron-15-3567:~/Desktop/security/hw2$ python server.py
[STARTING] server is starting...
[LISTENING] Server is listening on 127.0.1.1
new cleint 0
[NEW CONNECTION] ('127.0.0.1', 40896) connected.
new cleint 1
[NEW CONNECTION] ('127.0.0.1', 40898) connected.
---
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.1.1', 5053), raddr=('127.0.0.1', 40896)>
requested cleint number is :0
---
<socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.1.1', 5053), raddr=('127.0.0.1', 40898)>
requested cleint number is :1
2629
1631
10403
13843
19388354556987911633918
1808137960145660276412
```