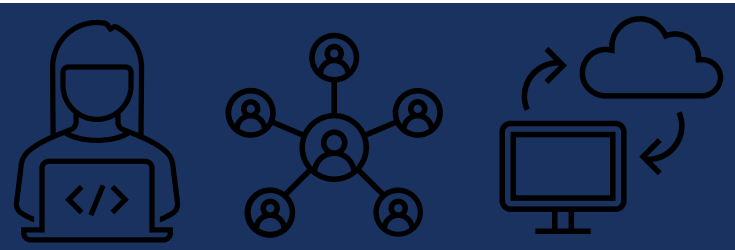
ChatGPT

GROUP 16: ARUSHI GHILDIYAL, ARABELLE BETZWIESER, CONNOR FRENCH, KENYON TINER, TYLER SAIZAN, ABBY DEBENPORT

TEAM BREAKDOWN



- Encryption:
  - Arushi Ghildiyal : Information Storage / Integration
  - Connor French : Key generation / Integration
  - Kenyon Tiner : Encryption / Decryption
- Networking:
  - Arabelle Betzwieser : Server / Client / Message Handling
  - Tyler Saizan : Data Flow of Messages / Design Pattern
  - Abby Debenport : System Architecture / UI

THE PROBLEM

Big Brother is always reading your messages

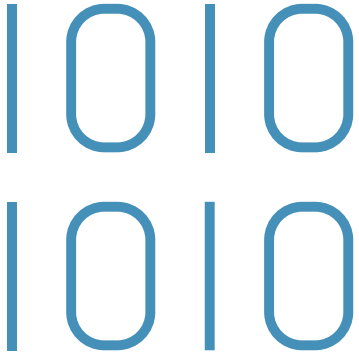
Can you REALLY trust WhatsApp?

Buying all these burner phones is getting expensive

Our Solution:

Networking

+



Encryption

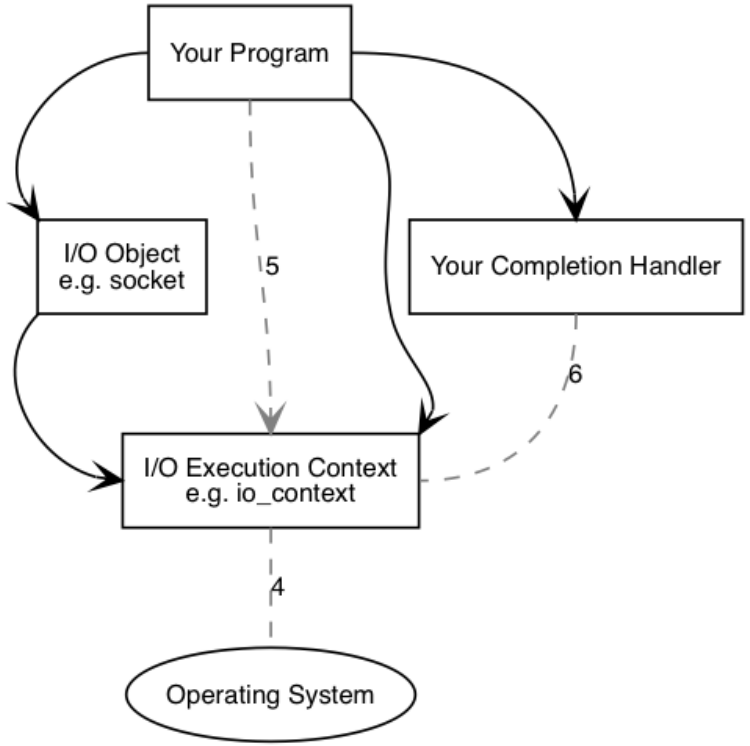
=

Safe and Secure Communication Line

CREATING AN ENCRYPTED COMMUNICATION CHANNEL OR CHAT ROOM

TECHNOLOGIES

- Asio
- Libsodium
- Nhlohmann
- ftxui
- C++
- C



JSON for Modern C++

What if JSON was part of modern C++?

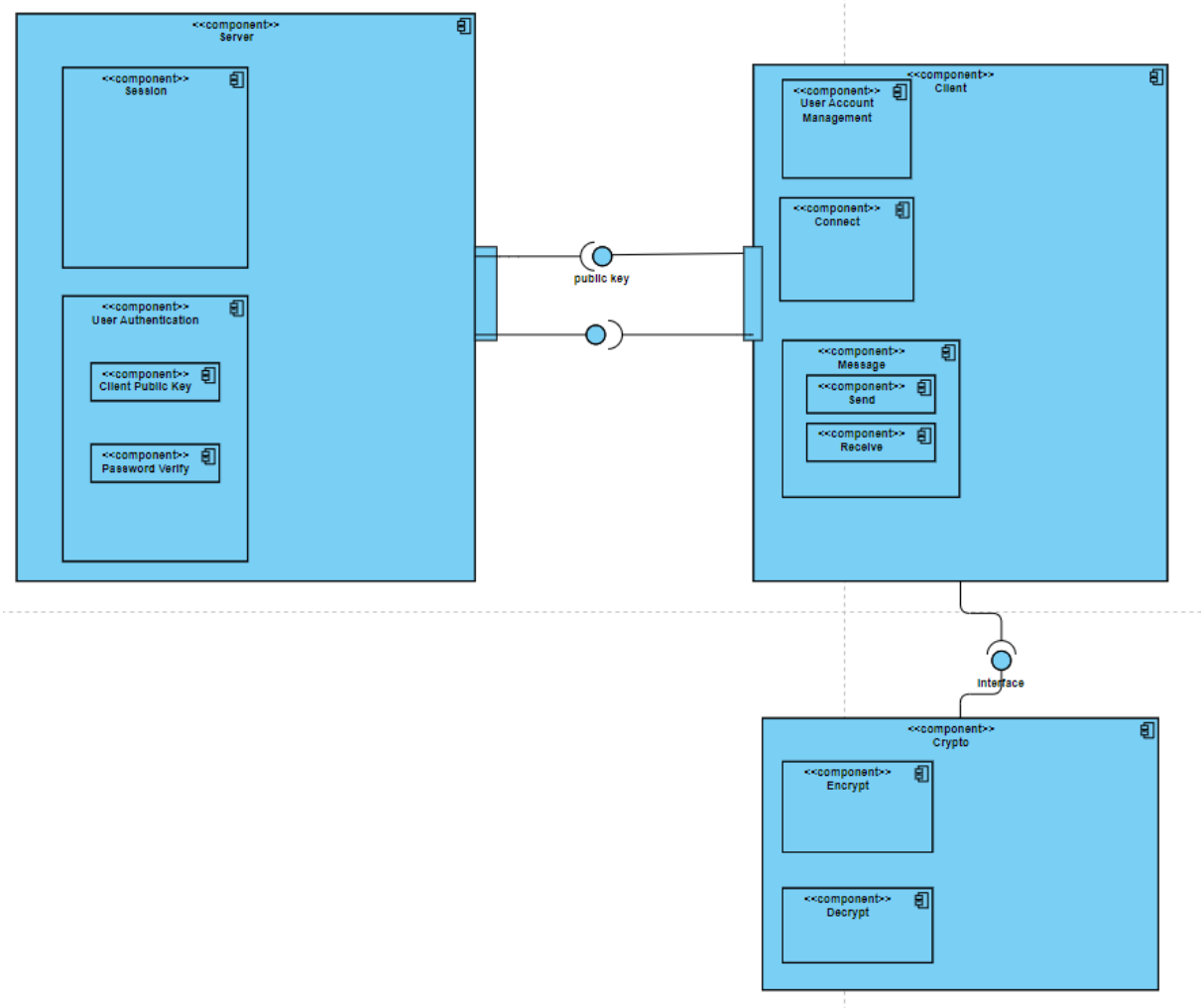
3.11.2

This is an animated GIF. Please wait for a feature slideshow.

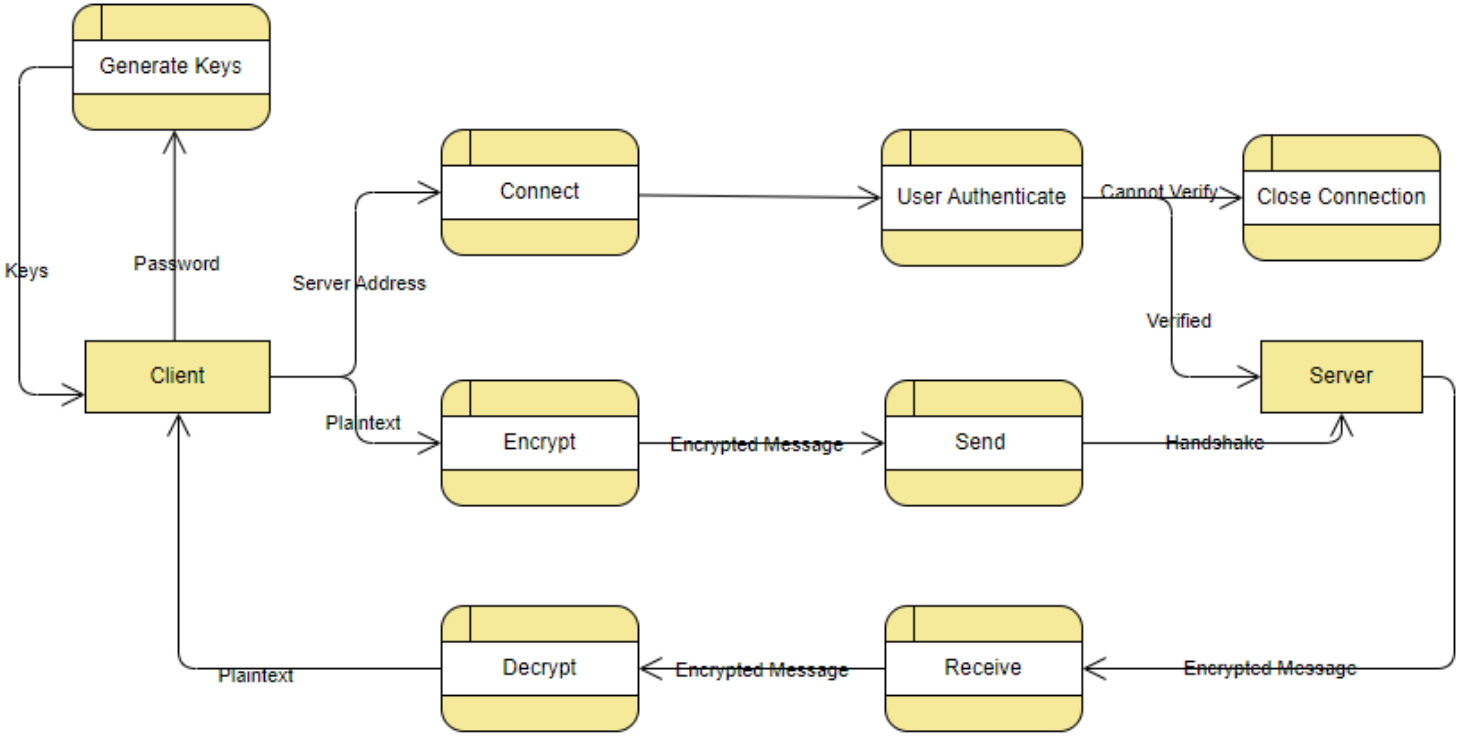
KEY FEATURES

- End-to-end encryption
- Asynchronous messaging
- Storage of user information server-side
- Group chatting capability

SYSTEM ARCHITECTURE – COMPONENT DIAGRAM



SYSTEM ARCHITECTURE – DATA FLOW

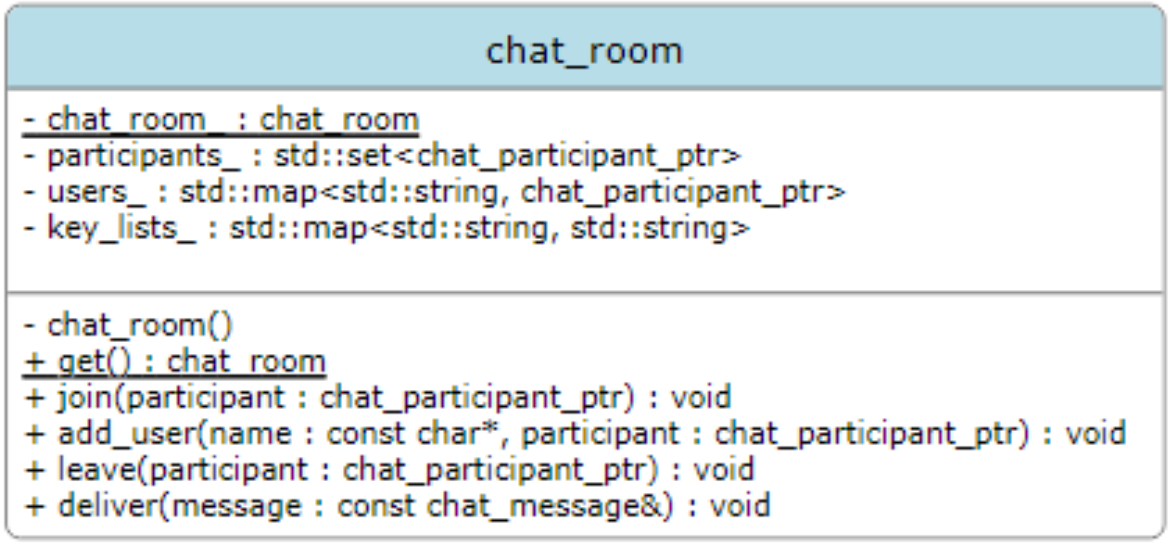


SYSTEM ARCHITECTURE – EXTERNAL DATA SOURCES AND SYSTEMS

NONE!

PREVIOUSLY MENTIONED LIBRARIES :^)

DESIGN PATTERN CLASS DIAGRAM :TYLER



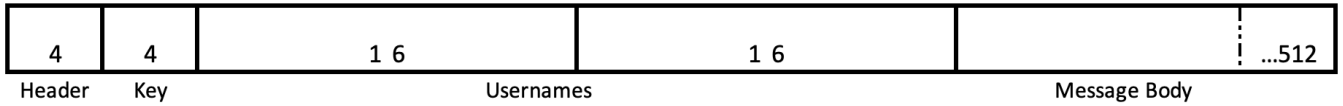
CONTRIBUTION: TYLER

chat_message

- data_
- body_length_
- target_user_
- source_user_
- key_signal_

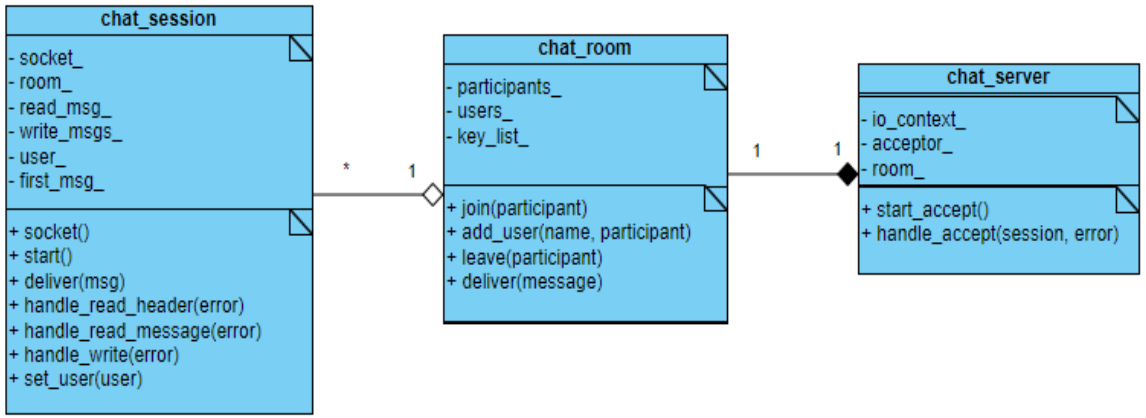
+ chat_message()
+ data() : char*
+ length() : size_t
+ body() : char*
+ message_length() : size_t
+ body_length() : size_t
+ body_length(new_length : size_t) : void
+ decode_header() : bool
+ encode_header() : void
+ source_username() : char*
+ target_username() : char*
+ decode_usernames(): void
+ encode_usernames(source_user : char* const, target_user : char* const) : void
+ has_key() : bool
+ decode_key() : void
+ encode_key(has_key : bool) : void

- How do we know where the username is in a message?
- How long is the message?



CONTRIBUTION: ARABELLE

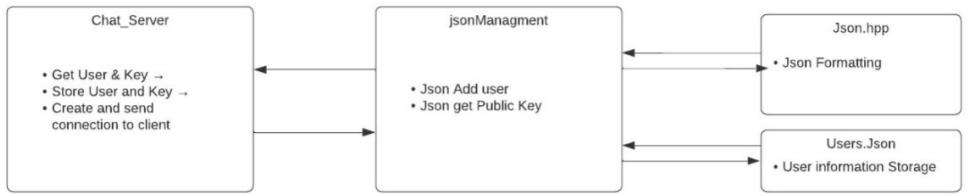
Class Diagram



- How do the clients/server read/write messages?
- What do the clients/server do when receiving messages?
- How does the server know where to send messages?

CONTRIBUTION :ARUSHI GHILDIYAL

Class Diagram

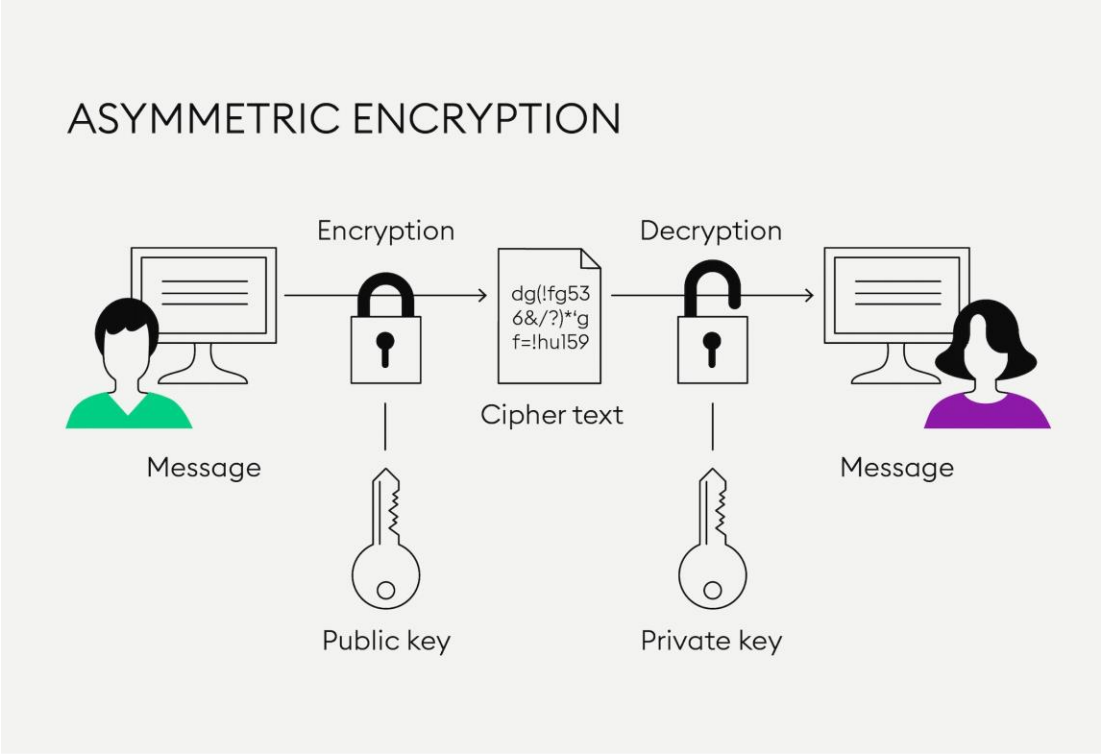


Contribution

- Information capture and storage
- UTF-8 Encoding
- Database integration

ENCRYPTION / DECRYPTION

- Using Public Key Encryption Aka Asymmetric Encryption
- Using libsodium's crypto_box functions, we can utilize block cypher encryption for each message that is passed into the encryption / decryption functions
- Encryption returns a string which is made up of the encrypted message and a random one-time use authentication code called a nonce
- The decryption function separates the nonce and message and returns the original plaintext for the receiver only if key pairs and nonce match up



INTEGRATION HELL

```
User bbb is connected.

public key received from the message:
bacb247e1a6fd17e6acbd1ddba6e2dbf1369852123d6
a73ef82ed707447
keylist:
bbb bacb247e1a6fd17e6acbd1ddba6e2dbf13698521
23d6a73ef82ed7000
```

```
sender_pub_key.data:
bacb247e1a6fd17e6acbd1ddba6e2dbf
1369852123d6a73ef82ed7dc8a32

noncelength:
24

msg size:3
Decryption failed. Message tampe
red or invalid key pair.
terminate called after throwing
an instance of 'std::logic_error
'
  what():  basic_string::_M_cons
truct null not valid
Aborted (core dumped)
```

```
User ccc is connected.

public key received from the mess
age:
a94586f0627ebd368b4cd05c86b9ab6d4
6508c63d889b6d42389827e126d2c
ccc:
privatekey:
9ec8b7c7cd68304de6567380b1af494f6
d149de0d79c64a78f1021585357c94e
senderpub:
7084ad0000000000008083ad00000a08
3ad00000
msgbody:
7084adDecryption failed. Message
tampered or invalid key pair.
Segmentation fault (core dumped)
@w-c-f →/workspaces/csc3380-fall
-2023-project-group-16 (main) $
```

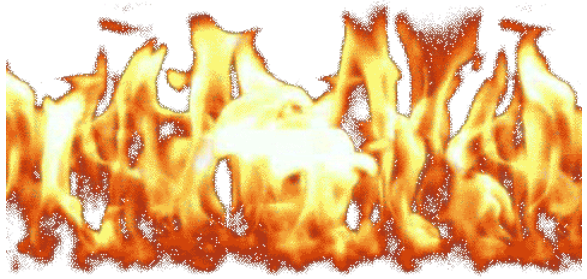
```
decrypt: cipher_with_nonce:
47ffffffb4ffffffbd5d76ffffff8339
1236ffffff9b316519ffffffad1e45ff
ffffff9ffffff84ffffffc463ffffffd8
ffffff89b6d423ffffff89ffffff827e
126d2c
Decryption failed. Message tampe
```

```
username sending to:bbb
keylist:
aaa ⸮E⸮⸮b~⸮6⸮L⸮\⸮⸮⸮mFP⸮c⸮
                          m#⸮⸮~m,
⸮⸮D⸮
bbb ⸮E⸮⸮b~⸮6⸮L⸮\⸮⸮⸮mFP⸮c⸮
                          m#⸮⸮~m,
⸮⸮D⸮
size of key_list: 2
```

```
/build/client localhost 1225
What is your username? (max of 16
 characters)
aaa
Please enter your password
123
public key:
54f5be656f35ffd4bac84afb681a983b9
2ae0d477161cc2e3e0f97efdc8a32
private key:
d947c1b0776433f7e551826a35c60a822
c0161939c0f5b9b7d28e7a0658a0
User aaa is connected.
User bbb is connected.
keylist:
bbb bacb247e1a6fd17e6acbd1ddba6e2
dbf1369852123d6a73ef82ed7000
User ccc is connected.
keylist:
bbb bacb247e1a6fd17e6acbd1ddba6e2
dbf1369852123d6a73ef82ed7000
ccc a94586f0627ebd368b4cd05c86b9a
b6d46508c63d889b6d42389827e126d2c
```

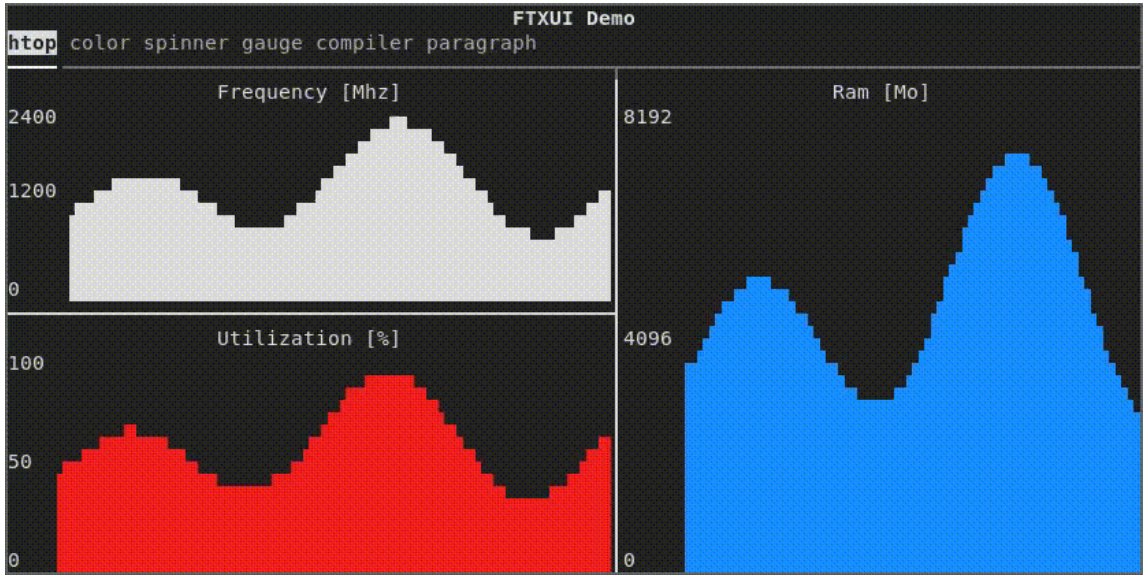
```
/build/client localhost 1225
What is your username? (max of 16
 characters)
bbb
Please enter your password
123
public key:
bacb247e1a6fd17e6acbd1ddba6e2dbf1
369852123d6a73ef82ed707447
private key:
9ec8b7c7cd68304de6567380b1af494f6
d149de0d79c64a78f1021585357c94e
User aaa is connected.
keylist:
aaa 54f5be656f35ffd4bac84afb681a9
83b92ae0d477161cc2e3e0f97efdc8a32
User bbb is connected.
User ccc is connected.
keylist:
aaa 54f5be656f35ffd4bac84afb681a9
83b92ae0d477161cc2e3e0f97efdc8a32
ccc a94586f0627ebd368b4cd05c86b9a
b6d46508c63d889b6d42389827e126d2c
```

```
6 characters)
ccc
Please enter your password
123
public key:
a94586f0627ebd368b4cd05c86b9ab6d
46508c63d889b6d42389827e126d2c
private key:
ee94d9d09f1bdfd2b517f40f6580d72e
cc5a89a74ef858f11da82ac6995
User aaa is connected.
keylist:
aaa 54f5be656f35ffd4bac84afb681a
983b92ae0d477161cc2e3e0f97efdc8a
32
User bbb is connected.
keylist:
aaa 54f5be656f35ffd4bac84afb681a
983b92ae0d477161cc2e3e0f97efdc8a
32
bbb bacb247e1a6fd17e6acbd1ddba6e
2dbf1369852123d6a73ef82ed7dc8a32
User ccc is connected.
```



USER INTERFACE USING FTXUI : ABBY DEBENPORT

- Upgrade from basic command line interface
- FTXUI library



THANK YOU