

GTÜ CSE 495&496

BLOCKCHAIN BASED
SECURE MESSAGING APPLICATION

π 2Pcath

Muhammed Bedir ULUCAY

Project Consultant:

Prof. Dr. Ibrahim SOGUKPINAR

December 2022

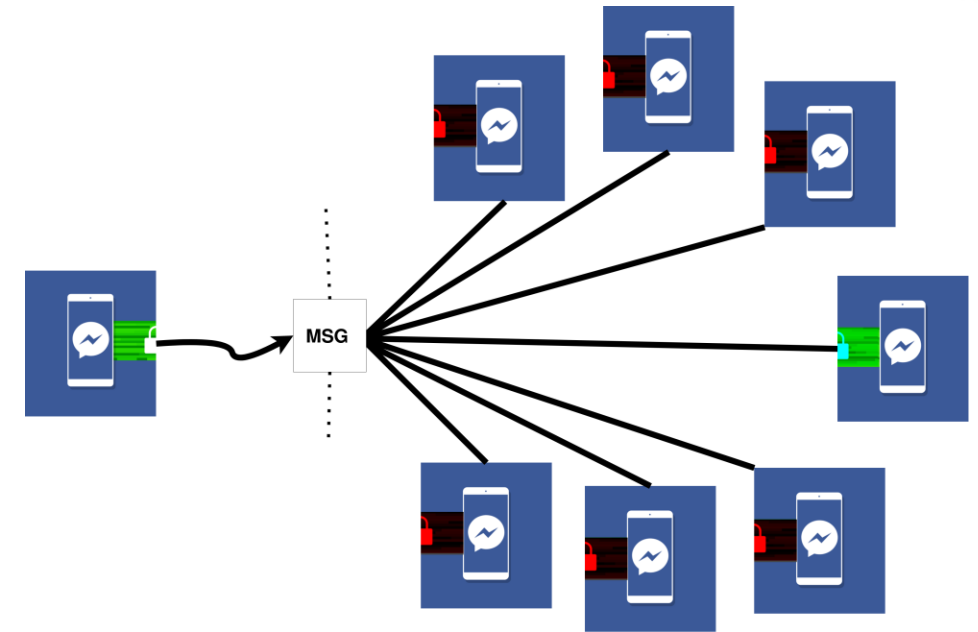


CONTENTS

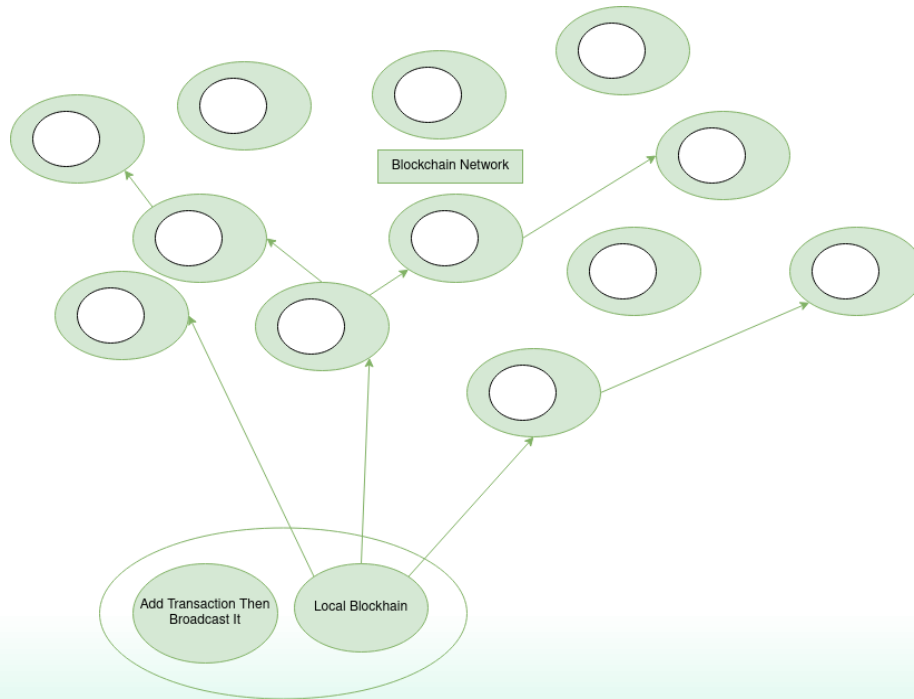
- Scheme and Description of the Project
- Project Design Plan
- Project Requirements
- The Work Done
- Implementation Design And Test
- Things To Do
- Success Criteria
- References

SCHEME AND DESCRIPTION OF THE PROJECT

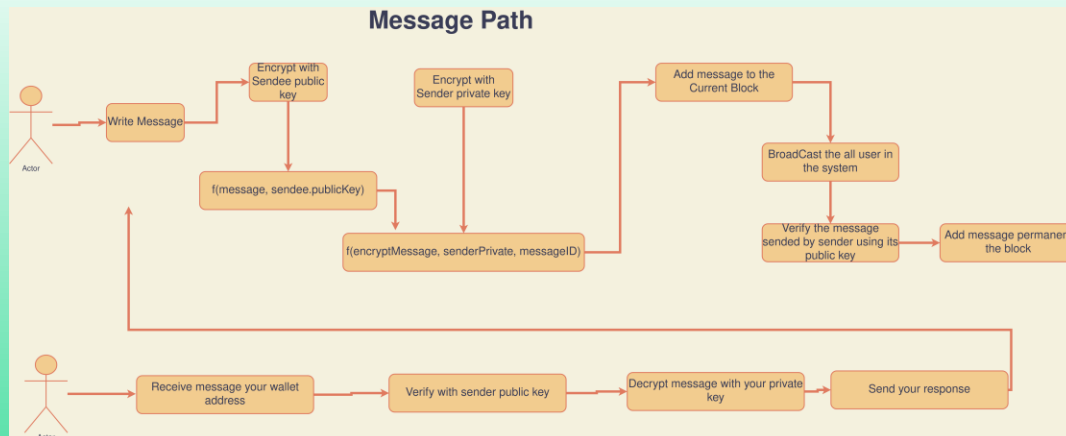
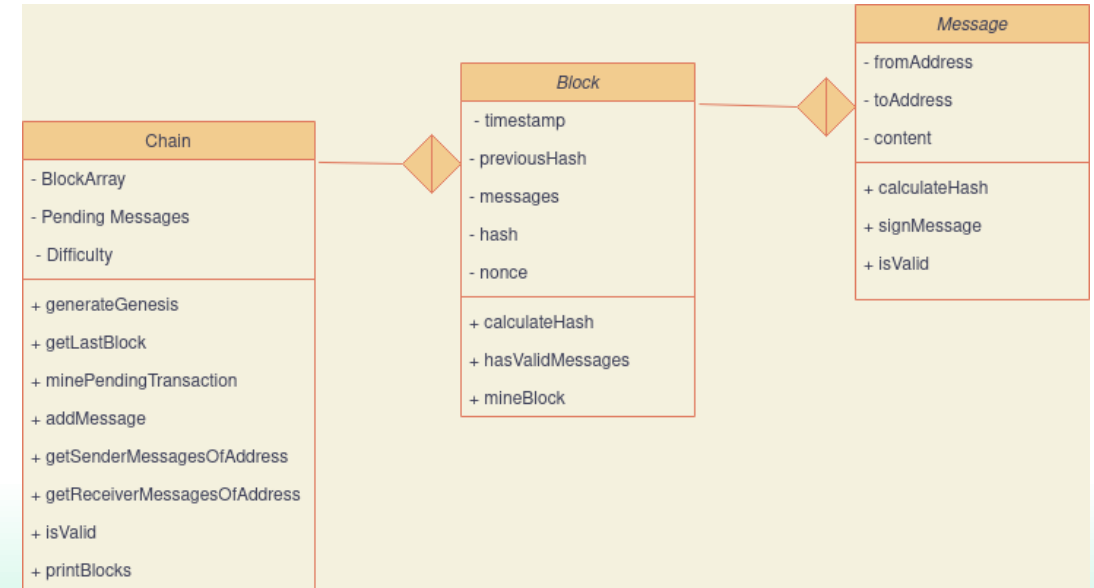
An application where people can securely communicate with each other as a peer to peer. To create a reliable and unchangeable communication channel between users using blockchain and cryptology. In this way, users can be absolutely sure that their data cannot be accessed by third parties or malicious people and from the recipient side.



PROJECT DESIGN PLAN



People will communicate
via smart contracts and a blockchain network



Similar Project:

- Dust
- E-Chat
- OpenChat
- Status
- BeeChat
- BChat

PROJECT REQUIREMENTS - 1

- It must encode Messages asymmetrically according to the algorithm to be selected.
- It must create the necessary blocks for new messages
- I needs to research algorithms for unique number values to be used in the system.
- I needs to create a list of smart contracts required for the blockchain chain
- I should research the languages that I can use for the mobile application.
- I should decrease the size of the data so that it is appropriate for the users
- It needs to increase the message sending speed above a certain threshold.
- It must create the necessary unique data for each user locally so that others cannot access it.

PROJECT REQUIREMENTS - 2

- I need mobile Cross platform library
- Go / Rust, JS, NodeJS
- Solidity for smart contracts of the blockchain structure
- 2 phone for testing
- Cryptographic hash algorithms
- Algorithms that generate 256-bit unique numbers
- Meta mask extension for connect the Ethereum blockchain network
- Ethereum mobile boilerplate
- Truffle Framework for local chain
- Libp2p, web3.js, Parity

THE WORK DONE

- Build a local blockchain structure with message and block classes
- Build an interface for interaction with the blockchain
- Asymmetric encryption and decryption for a user
- Creating and sending message from a user to an address (public key)
- Mine block for pending messages and hashing with the new blocks
- Make sign a transaction to not change
- Read the article about designing a blockchain chat application.
- I learn the solidity, JS and angular for building application

IMPLEMENTATION DESIGN AND TEST



MessageApp

SettingsCreate & Send MessagePending Messages

User Public Key:
MIGfMA0GCsQGSib3DQEBAQUAA4GNADCBiQKBgQCqZZJKF9hZ1BEP3BVTzQmzidnDLF4obCv6WXibixJ8myqHfkSj13w2ilY8QEY3ON9zE7fq6iDreKdJX2l3zjUFRkbkfAPrLVvAeZldOzo1XWRbclhOiBeBeA6e

User Private Key:
MIICdglBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAKpIkuX2FnUEQ/cFVPNCbOJ2cMsXihsK/pZeJuLEnybKod+RKPXiDalhixARjc433MTt+rqIOt4p0ffaXfONQWuRuR/8A+stW8B5mV05mjVdZFyW

Blocks On Chain

Each card represents a block in chain. For detail click the block on the card

Hash

a6677ad5d257ab45b544cae8e8a005...

Hash of previous block

0

Nonce

0

Timestamp

1670274103728

Main page of the application:
We see genesis block and its message contents listed below the page

Messages in block

#	From	To	Content	Valid?
0	04a51d4836812fe04...	mbulucay	mbulucay thank you for your services	✓

MessageApp

SettingsCreate & Send MessagePending Messages

Create Message

Send message

From address

04e9723b457bad08ab0c4a83b800107df181e5be33245c43c26e7637f9526b3fbe7b8014053ecca5f6b09c043acda6511ff07d3d0d9181bf2e8cf58de700ab

This is your wallet address. You cannot change it because you can only spend your own coins.

To Address

MIGfMA0GCsQGSib3DQEBAQUAA4GNADCBiQKBgQCqZZJKF9hZ1BEP3BVTzQmzidnDLF4obCv6WXibixJ8myqHfkSj13w2ilY8QEY3ON9zE7fq6iDre

Content

Blockchain based chat application

Sign & send message

Message creation page:
Sending message and sender address and receiver address and filing the content of the messages. The button sending transaction to pending transaction

MessageApp

SettingsCreate & Send MessagePending Messages

#	From	To	Content
0	04e972...	MIGfMA...	SYAxo1MLrMxZDi29aHAFe8h7Tx9RKsY78YGgF2nOKzcRWsCVdXNHjmTcQvQepE8kpk+uhz15N9IE6BJDk93J3RMLccVARSt55xlz6i+2t1vzGWfge2Vy8YXA/Zyt7pBEy5ywhx5e

Start Transactions



Mine the new block with the pending transaction never to be changed again.

You can see the content of the message is encrypted. Just the owner can see the main text.

MessageApp

SettingsCreate & Send MessagePending Messages

User Public Key:
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCqZZJVkF9hZ1BEP3BVTzQmzidnDLF4obCv6WXibixJ8myqHfkSj13w2ilY8QEY3ON9zE7fq6iDreKdJX2i3zjUFRkbkI/APrLVvAeZldOZo1XWRbclhIOiBeBeA6e

User Private Key:
MIICdglBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAKpikiUoX2FnUEQ/cFVPNCbOJ2cMsXihsK/pZeJuLEnybKod+RKPXfDalhjxARjc433MT+rqIOt4p0lfaXfONQWuRuR/BA+stW8B5mV05mjVdZFtyW

Blocks On Chain

Each card represents a block in chain. For detail click the block on the card

Hash

a6677ad5d257ab45b544c4ae8e88a005...

Hash of previous block

0

Nonce

0

Timestamp

1670274103728

Hash

0038cfc5c8020bb9aeee912218b12f654...

Hash of previous block

a6677ad5d257ab45b544c4ae8e88a005...

Nonce

359

Timestamp

1670274275154

Messages in block

#	From	To	Content
0	04e972...	MIGfMA...	SYAxo1MLrMxZDi29aHAFe8h7Tx9RKsY78YGgF2nOKzcRWsCVdXNHjmTcQvQepE8kpk+uhz15N9IE6BJDk93J3RMLccVARSt55xlz6i+2t1vzGWfge2Vy8YXA/Zyt7pBEy5ywhx5e

Mined block and added to the chain. The hashing is also working with previous hash.

Validate the encrypted message

RSA Decryption

Enter Encrypted Text to Decrypt (Base64)

```
SYAxoIMLrMxZDI29aHAFe8h7Txf9RKsY78YGgF
2nOKzcRWsCVdXNHjmTcQvQepE8kpk+uhz15
N9IE6BJDk93J3RMLccVARst55xlz6i+2tlvzGWf
```

Enter Public/Private key

```
MIICdgIBADANBgkqhkiG9w0BAQEFAASCAMA
wggJcAgEAAoGBAKplUoX2FnUEQ/cFVPNCb
QJ2cMsXihsK/pZeJuLEnybKod+RKPXFdlhxA
Rlc433MTt+rqOt4p0IfaXfONQWuRuR/8A+st
W8B5mV05mjVdZFtyWEg6IF4F4Dp7UfSSG6
```

RSA Key Type: ☐ Public key ☒ Private Key

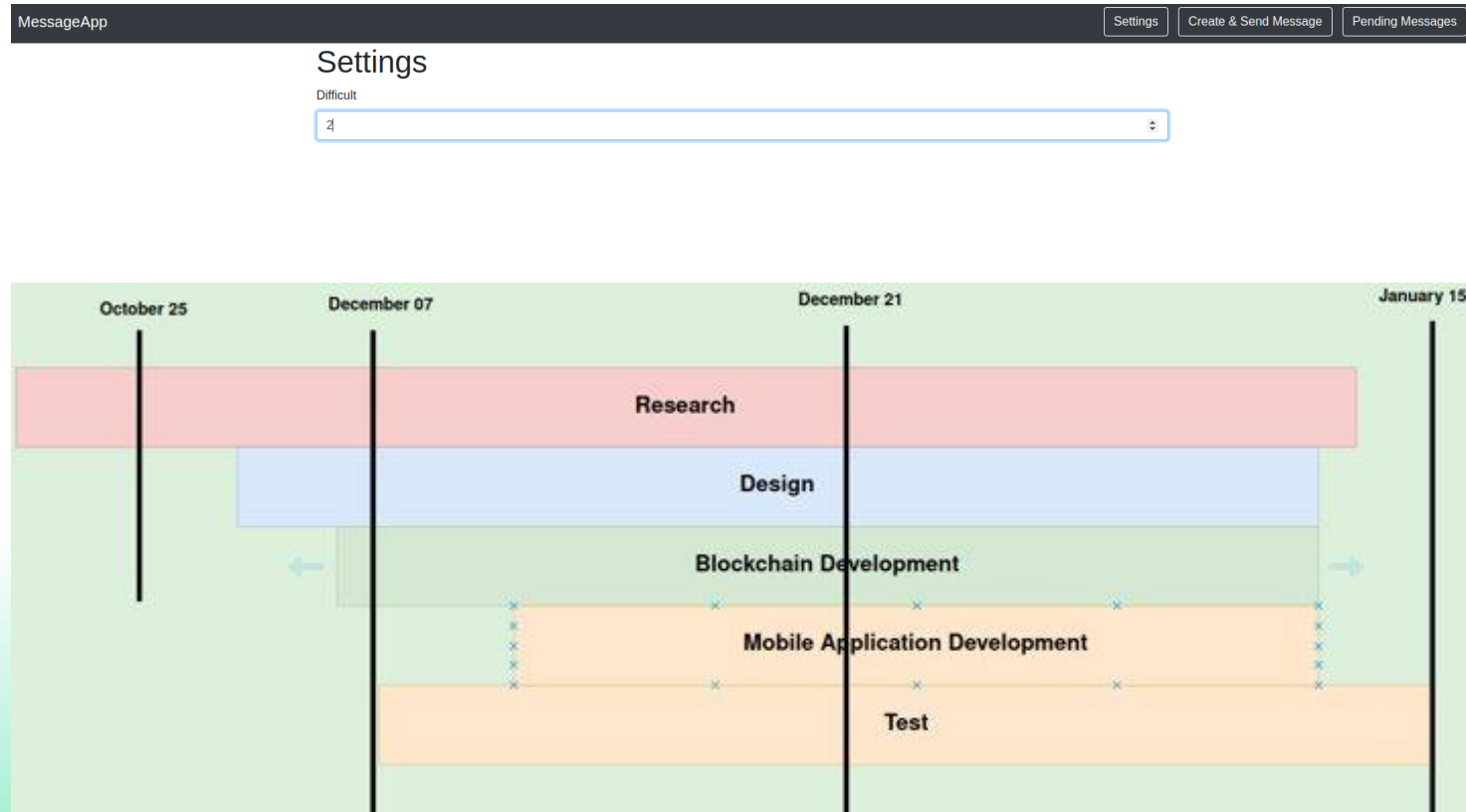
Select Cipher Type

RSA/ECB/PKCS1Padding

Decrypt

Decrypted Output:

Blockchain based chat application



For Code:

<https://github.com/mbulucay/Blockchain/tree/main/OwnBlockchain/angular/MessageApp>

THINGS TO DO

- Creating a similar application using the smart contract.
- Build local blockchain network using truffle/ganache framework.
- Learn Ethereum blockchain network usage.
- Deploy own blockchain on Ethereum blockchain network.
- Build an interface for the application mobile/web.
- Create an Api services for broadcasting messages.
- Connect Meta mask extension with using tools(ganache/blockchain network)
- Learn Go or rust for the Api services
- Lots of research about blockchain

SUCCESS CRITERIA



Making message transaction below 4 second (human psychological)



Usage and signing of the program after installation in 10 minute.



Loading message transactions from blockchain under 2 minute.



When a text search is requested, all messages containing the text written in less than 2 minute can be displayed to the user



(depends on the size of blockchain)

REFERENCES

- https://www.researchgate.net/publication/318131748_An_Overview_of_Blockchain_Technology_Architecture_Consensus_and_Future_Trends
- <https://www.irjet.net/archives/V7/i5/IRJET-V7I5531.pdf>
- <https://github.com/machinomy/awesome-non-financial-blockchain#readme>
- <https://scholarworks.calstate.edu/concern/theses/qj72pb04f?locale=en>
- <https://bitcoin.org/bitcoin.pdf>
- <https://github.com/TristanBilot/blockchain-chat-app>
- https://www.youtube.com/watch?v=hYip_Vuv8J0&t=261s
- <https://www.youtube.com/watch?v=bBC-nXj3Ng4&t=1065s>
- https://www.youtube.com/watch?v=ZEAplE8KkE&list=PLxz5ldaTYSOUmhECFNN-WfGeJrzlnXn_a
- <https://www.blockchain.com/tr/explorer>
- <https://www.youtube.com/watch?v=yubzJw0uiE4&t=261s>
- <https://medium.com/adamant-im/how-decentralized-blockchain-messenger-works-b9932834a639>
- <https://medium.com/@BeFastTV/top-blockchain-messaging-apps-crypto-messengers-28893e5f908f>
- <https://hal.archives-ouvertes.fr/hal-02180329/document/>
- <https://docs.soliditylang.org/en/v0.8.17/introduction-to-smart-contracts.html>
- <https://remix.ethereum.org>
- https://www.researchgate.net/publication/328160285_Survey_of_Consensus_Protocols
- <https://blog.harmony.one/peer-discovery-in-harmony-network/>
- <https://rejolut.com/blog/creating-your-own-blockchain-network/>