SPL-1 Project Report, 2023

BlockChainED

SE: 305

Submitted by

Nusrat Jahan Lia

BSSE Roll No. : 1306

BSSE Session: 2020-2021

Supervised by

Dr.Ahmedul Kabir

Designation: Associate Professor

Institute of Information Technology

Institute of Information Technology

University of Dhaka

[21-05-2023]

Table of Contents

1. Introduction

BlockChainEd is an educational project that empowers users to create and explore their own blockchain networks. With an intuitive app, users can easily generate wallets, conduct transactions, assign miners, and examine decentralized information. The application validates transactions, executes proof of work, and seamlessly integrates verified blocks into the blockchain. Additionally, BlockChainEd offers a unique feature that allows users to simulate attacks on their blockchain, enabling them to modify information within a valid block. However, the app promptly safeguards against such attacks by recalculating hashes and breaking the chain from the corrupted block. BlockChainEd serves as an interactive platform that not only educates users about blockchain technology but also allows them to experience its practical implementation.

1.1. Background Study

To implement the "BlockChainEd" project, I had to gather the following study and background knowledge:

1. ***Blockchain Technology:*** Understanding the fundamental concepts and principles of blockchain technology, including decentralized consensus mechanisms, distributed ledger systems, blocks, and transactions.

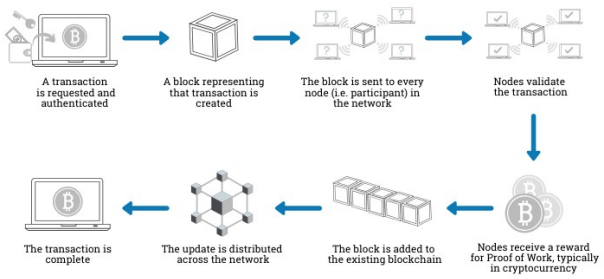


Figure 1 : Process of blockchain

2. ***Cryptography:*** Knowledge of cryptographic techniques used in blockchain systems, such as public key cryptography, elliptic curve cryptography (ECC), and digital signatures. Understanding the concepts of private keys, public keys, and how they are used to secure transactions and wallets.

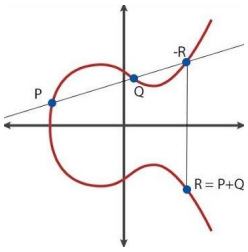
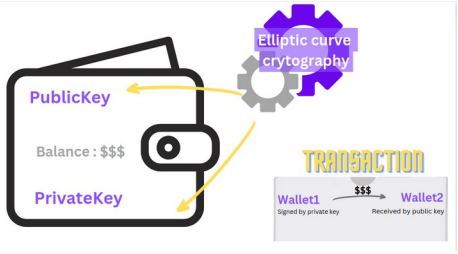


Figure 2 : Point addition in Elliptic Curve Cryptography

3. ***Wallet Creation:*** Learning about the process of generating public-private key pairs for wallet addresses, creating wallets, and managing key pairs securely. Understanding how wallet addresses are used to send and receive transactions.



Wallets with ECC generated Keys

4. ***Transactions and Mining:*** Understanding how transactions are created, signed, and propagated in a blockchain network. Studying the process of appointing miners to validate transactions, perform proof-of-work (PoW) calculations, and add verified blocks to the blockchain.

5. ***Consensus Mechanisms:*** Researching different consensus mechanisms used in blockchain networks, such as PoW or proof-of-stake (PoS). Understanding how consensus algorithms ensure the agreement and security of the blockchain.

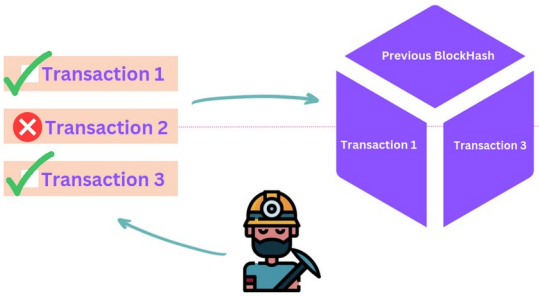


Figure 3 : Miner verifying transactions .

6. ***SHA-256 Hashing Algorithm:*** Learning about the SHA-256 algorithm used for hashing in blockchain systems. Understanding how it ensures the integrity and security of transactions and blocks.

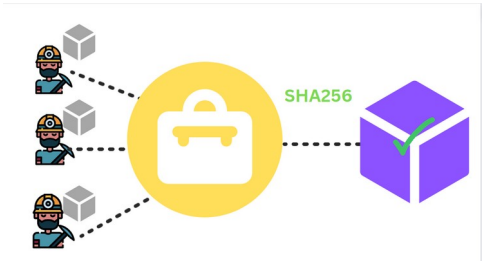


Figure 4 : Miners going through proof-of-work algorithm to find valid hash of block

7. ***Blockchain Attacks:*** Researching various types of attacks that can occur in a blockchain network, such as double-spending attacks. Understanding the vulnerabilities and countermeasures to prevent such attacks.

8. ***Information Inspection:*** Exploring methods to inspect and retrieve information stored in a decentralized environment. Understanding how to access and verify transaction details, block contents, and other relevant data.

By gathering the above study and background knowledge, one would have the necessary foundation to implement the "BlockChainEd" project successfully.

1.2. Challenges

Implementing the "BlockChainEd" project comes with several challenges:

1. ***Complexity:*** Blockchain technology involves intricate concepts like consensus algorithms, cryptographic techniques, and decentralized data structures. Understanding and implementing these concepts correctly can be challenging, especially for developers who are new to blockchain.

2. ***Security:*** Security is a crucial aspect of blockchain systems. Ensuring the secure generation and storage of private keys, implementing robust authentication and encryption mechanisms,and protecting against potential attacks require careful attention and expertise.

3. ***Performance and Scalability:*** As the blockchain grows in size and the number of transactions increases, maintaining optimal performance and scalability becomes challenging. Efficient transaction processing, block validation, and network scalability need to be considered during the development process.

4. ***Consensus Mechanisms:*** Implementing a consensus mechanism, such as proof-of-work (PoW) or proof-of-stake (PoS), involves addressing challenges related to block validation,miner selection, and network coordination. Each consensus algorithm has its own complexities that need to be understood and implemented correctly.

5. ***Blockchain Attacks:*** Anticipating and preventing various attacks, such as 51% attacks,double-spending attacks, or Sybil attacks, requires a deep understanding of the vulnerabilities and

countermeasures specific to blockchain systems. Implementing robust security measures to protect the integrity of the blockchain is crucial.

6. ***User Experience:*** Designing a user-friendly interface that allows users to create wallets, make transactions, and inspect blockchain information intuitively can be a challenge. Balancing simplicity with the complex underlying technology is essential to ensure a positive user experience.

7. ***Testing and Debugging:*** Blockchain applications require thorough testing to identify and fix potential bugs or vulnerabilities. Ensuring the reliability and stability of the application, particularly in a decentralized environment, can be challenging and time-consuming.

8. ***Integration and Interoperability***: Integrating the blockchain application with other systems or platforms, such as cryptocurrency wallets or third-party services, may present challenges. Ensuring compatibility and seamless interaction between different components can be complex.

Overcoming these challenges requires a combination of deep technical knowledge, careful planning, and continuous testing and improvement. It is important to stay updated with the latest advancements and best practices in blockchain development to address these challenges effectively.

2. Project Overview

BlockChainEd is an innovative project that I have been working on, aimed at empowering users to create their own blockchain networks. With BlockChainEd, users have the ability to create wallets, make transactions, and even appoint miners within their personalized blockchain environment. The application incorporates various essential features such as transaction verification, proof of work, and the seamless addition of verified blocks to the blockchain.

One of the key functionalities of BlockChainEd is the ability for users to inspect and explore the information within their decentralized environment. This provides users with a comprehensive understanding of the blockchain's inner workings and enhances their overall control over the system.

To ensure the utmost security and privacy, BlockChainEd leverages public-key cryptography,with a specific focus on Elliptic Curve Cryptography (ECC). This cryptographic technique enables secure key generation, encryption, and digital signature operations, ensuring the integrity and confidentiality of the user's transactions.

Furthermore, BlockChainEd implements the widely adopted SHA256 hashing algorithm. This algorithm plays a vital role in verifying the integrity of the data stored within the blockchain bygenerating a unique hash for each block. This ensures that any modifications to the information contained within a block will be immediately detected, safeguarding the integrity of the blockchain. An intriguing feature of BlockChainEd is the ability for users to test the resilience of their own blockchain by simulating an attack scenario. Users can attempt to change the information within a valid block, but BlockChainEd actively defends against such attacks. The application achieves this by recalculating the hash and subsequently breaking the chain from the corrupted block, ensuring the security and immutability of the blockchain.

In summary, BlockChainEd is an ambitious project that allows users to create, manage, and explore their own blockchain networks. By incorporating public-key cryptography using ECC and implementing the SHA256 hashing algorithm, BlockChainEd offers a secure and robust environment for users to engage in transactions while maintaining the integrity and privacy of their data.

3. User Manual

This user manual provides step-by-step instructions on how to obtain the BlockChainEd application from GitHub and run the RunBlockChainEd.cpp file. Please follow the guidelines below:

Prerequisites:

1. C++ Compiler: Ensure that you have a C++ compiler installed on your system, such as GCC or Clang.

2. Code Editor/IDE: Have a code editor or integrated development environment (IDE) installed to open and edit the RunBlockChainEd.cpp file.

Instructions:

1. Visit the BlockChainEd repository on GitHub at https://github.com/NusRAT-LiA/Blockchain_in_Cpp-SPL-1 .

2. On the GitHub repository page, click on the "Code" button and select the option to clone the repository. Copy the provided repository URL.

3. Open the Terminal/Command Prompt: Launch your system's terminal or command prompt to execute commands.

4. In the terminal or command prompt, navigate to the desired directory where you want to clone the BlockChainEd repository. Use the following command to clone the repository:
git clone https://github.com/NusRAT-LiA/Blockchain_in_Cpp-SPL-1

5. Compile the RunBlockChainEd.cpp file using the C++ compiler. Open your terminal or command prompt and navigate to the UserInterface directory where the RunBlockChainEd.cpp file is located. Then, run the following command:
g++ RunBlockChainEd.cpp -o BlockChainEd

6.Select From the Options :

Figure 5: Selected 1 for creating a blockchain

7.Enter your desired name for the blockchain and add mining difficulty

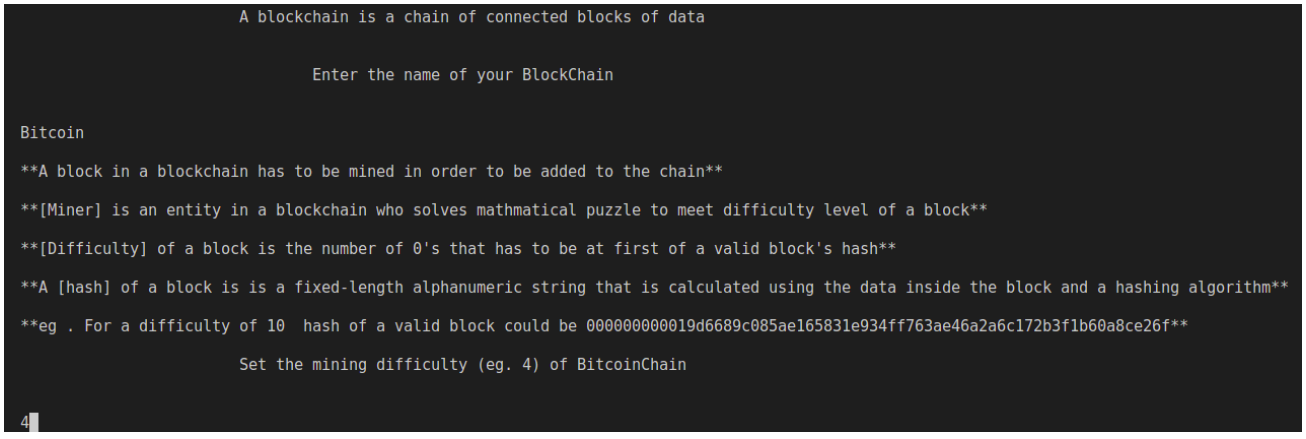


Figure 6 :Chosen name for the chain is “Bitcoin” , mining difficulty is “4”

8. A blockchain is initiated and genesis block is created .

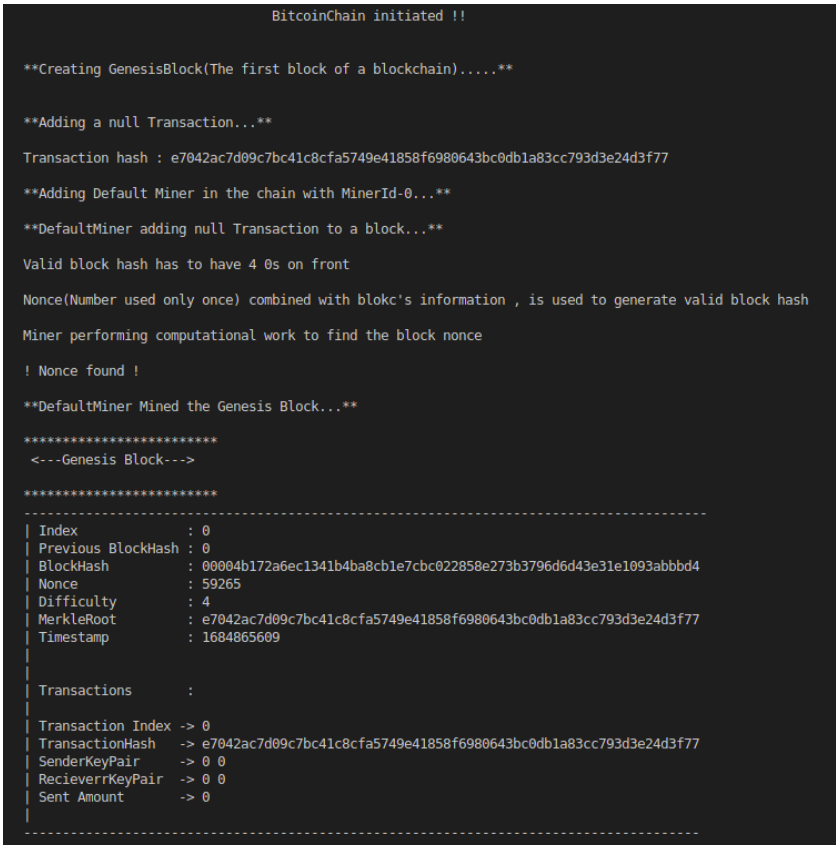


Figure 7 : Step by step process of creating genesis block

9. Create some wallets , wallet's key's will be generated through Elliptic-Curve-Cryptgraphic functionalities

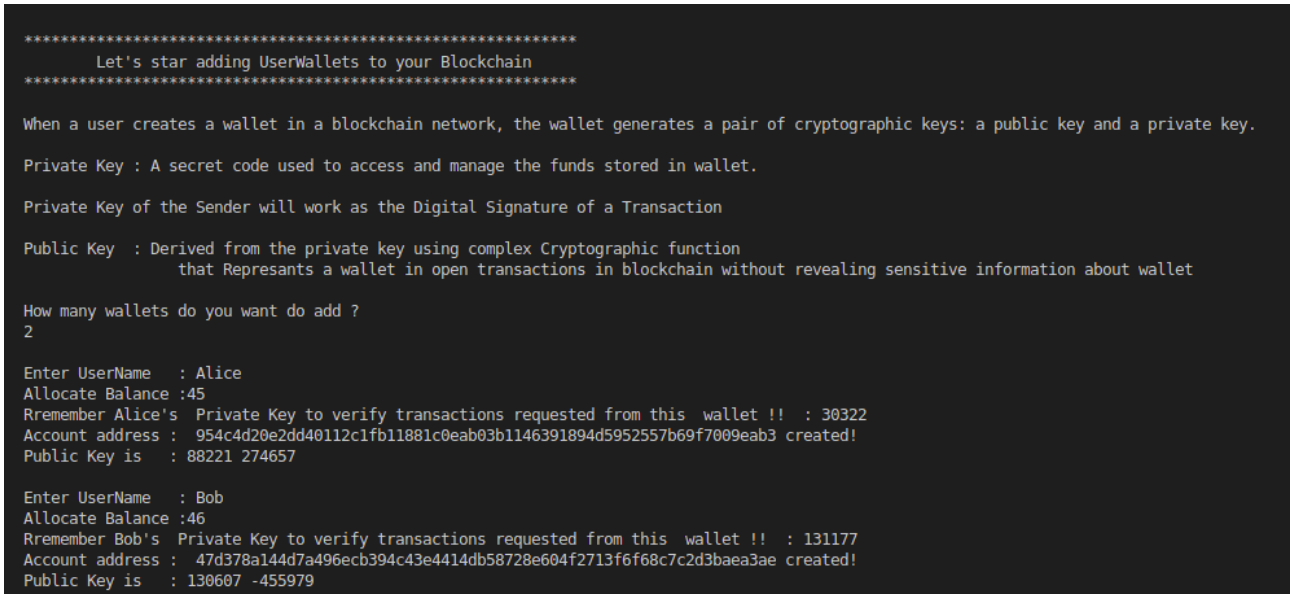


Figure 8 : Two wallets created with the names of "Alice" and "Bob" with respective 45 and 46 balance allocated

10.Create transactions using wallet’s keys

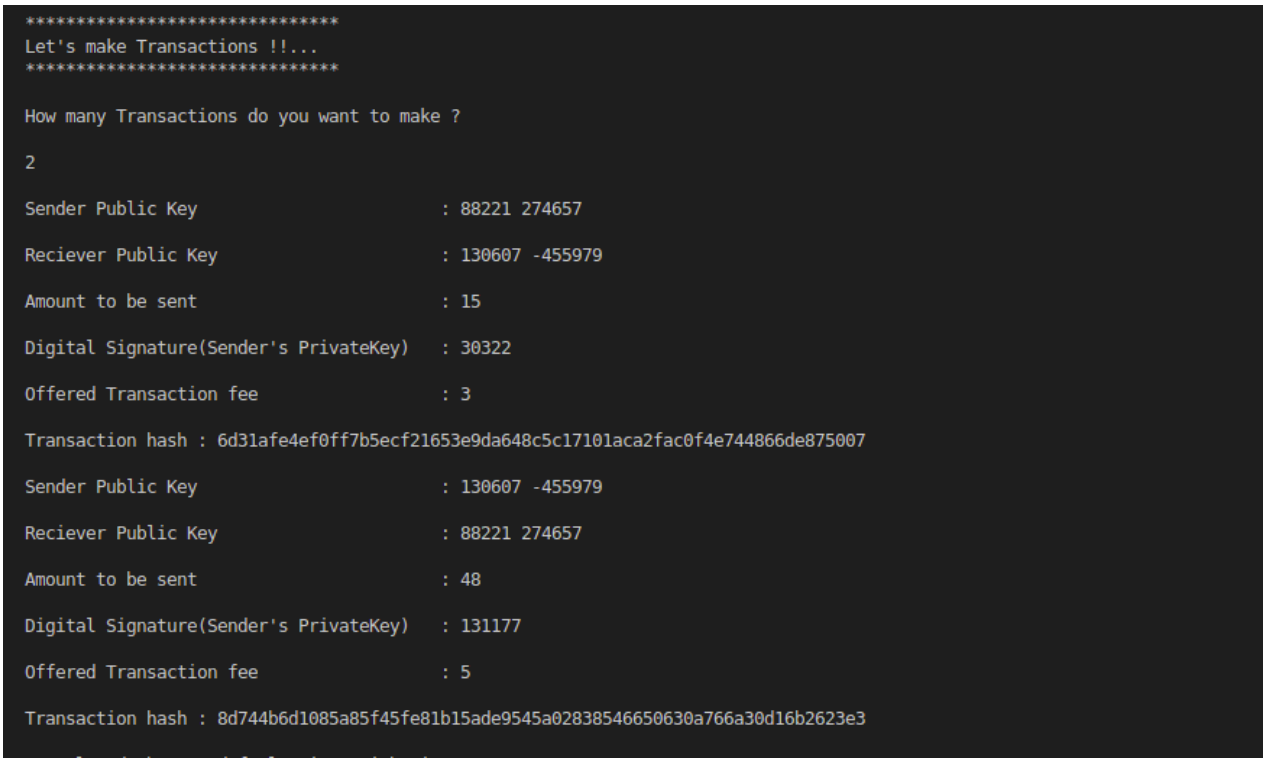


Figure 9 : First transaction is from “Alice” to “Bob” ; Second transaction is from “Bob” to “Alice”

11.If you want , add more miners along with default miner

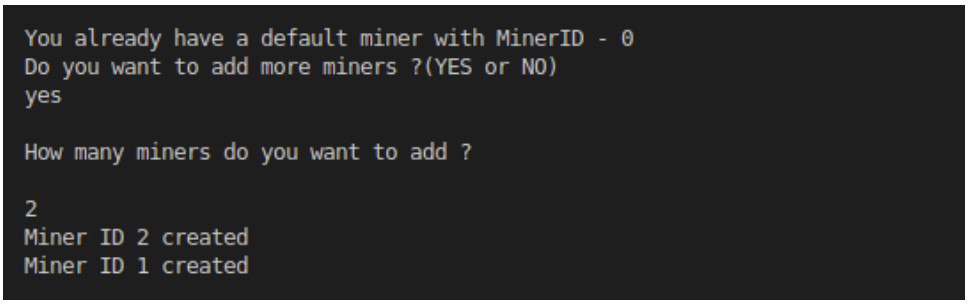
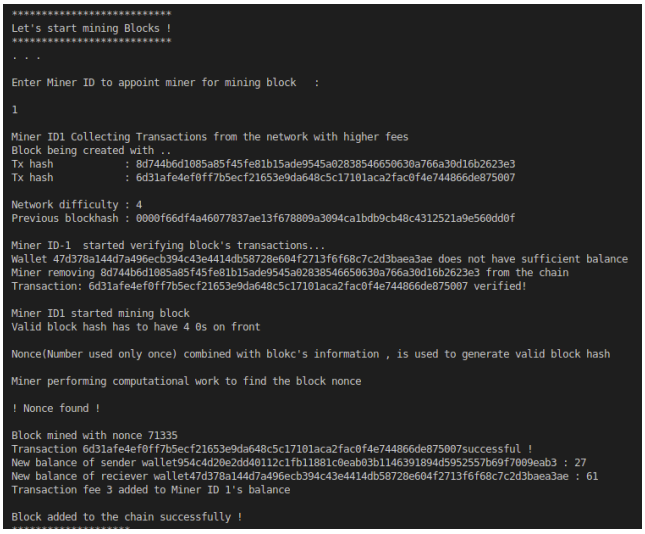


Figure 10 : Two miners created

12.Appoint miner to mine Blocks , miner will remove unverified transactions



```
Block added to the chain successfully !
**********
Here's your chain !
**********
--------------------------------------------------------------------------------
| Index              : 0
| Previous BlockHash : 0
| BlockHash          : 0000f66df4a46077837ae13f678809a3094ca1bdb9cb48c4312521a9e560dd0f
| Nonce              : 73846
| Difficulty         : 4
| MerkleRoot         : e7042ac7d09c7bc41c8cfa5749e41858f6980643bc0db1a83cc793d3e24d3f77
| Timestamp          : 1684866660
|
|
| Transactions       :
|
| Transaction Index -> 0
| TransactionHash   -> e7042ac7d09c7bc41c8cfa5749e41858f6980643bc0db1a83cc793d3e24d3f77
| SenderKeyPair     -> 0 0
| RecieverrKeyPair  -> 0 0
| Sent Amount       -> 0
|
--------------------------------------------------------------------------------
^
|
|
--------------------------------------------------------------------------------
| Index              : 1
| Previous BlockHash : 0000f66df4a46077837ae13f678809a3094ca1bdb9cb48c4312521a9e560dd0f
| BlockHash          : 0000355d0abe75dfa4747ea107904066e39703fdf379787c538312ad624fc421
| Nonce              : 71335
| Difficulty         : 4
| MerkleRoot         : 6d31afe4ef0ff7b5ecf21653e9da648c5c17101aca2fac0f4e744866de875007
| Timestamp          : 1684867152
|
|
| Transactions       :
|
| Transaction Index -> 1
| TransactionHash   -> 6d31afe4ef0ff7b5ecf21653e9da648c5c17101aca2fac0f4e744866de875007
| SenderKeyPair     -> 88221 274657
| RecieverrKeyPair  -> 130607 -455979
| Sent Amount       -> 15
|
--------------------------------------------------------------------------------
^
```

Figure 11 : Mined block with verified transaction added to your blockchain

13.Either keep creating wallets ,transactions , blocks or move on to next options(inspection /attacking)

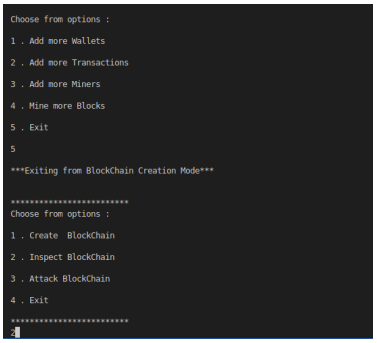


Figure 12 : Option choice

14 . Inspect elements of your blockchain

```
1 . Check Wallets of My Chain
2 . Check Transactions in My Chain
3 . Check for a Block
4 . Check Full chain
5 . Exit
1
Wallet's  Address     : 954c4d20e2dd40112c1fb11881c0eab03b1146391894d5952557b69f7009eab3
Wallet's  Public  Key : 88221 274657
Wallet's  Balance     : 45
Wallet's  Address     : 47d378a144d7a496ecb394c43e4414db58728e604f2713f6f68c7c2d3baea3ae
Wallet's  Public  Key : 130607 -455979
Wallet's  Balance     : 46
Choose from options :
1 . Check Wallets of My Chain
2 . Check Transactions in My Chain
3 . Check for a Block
4 . Check Full chain
5 . Exit
2
| TransactionHash   -> 6d31afe4ef0ff7b5ecf21653e9da648c5c17101aca2fac0f4e744866de875007
| SenderKeyPair     -> 88221 274657
| RecieverrKeyPair  -> 130607 -455979
| Sent Amount       -> 15
|
Choose from options :
1 . Check Wallets of My Chain
2 . Check Transactions in My Chain
3 . Check for a Block
4 . Check Full chain
5 . Exit
3
Enter the Block's Index :
1
-------------------------------------------------------------------------------------
| Index              : 1
| Previous BlockHash : 0000f66df4a46077837ae13f678809a3094ca1bdb9cb48c4312521a9e560dd0f
| BlockHash          : 0000355d0abe75dfa4747ea107904066e39703fdf379787c538312ad624fc421
| Nonce              : 71335
| Difficulty         : 4
| MerkleRoot         : 6d31afe4ef0ff7b5ecf21653e9da648c5c17101aca2fac0f4e744866de875007
| Timestamp          : 1684867152
|
|
| Transactions       :
|
| Transaction Index -> 1
| TransactionHash   -> 6d31afe4ef0ff7b5ecf21653e9da648c5c17101aca2fac0f4e744866de875007
| SenderKeyPair     -> 88221 274657
| RecieverrKeyPair  -> 130607 -455979
| Sent Amount       -> 15
|
-------------------------------------------------------------------------------------
```

Figure 13 : Wallets , Transactions and blocks the explored

15 . Generate an attack by changing transaction information

```
Choose from options :

1 . Create  BlockChain

2 . Inspect BlockChain

3 . Attack BlockChain

4 . Exit

************************
3
Enter the index of valid Block you want access to :

1
The block :


--------------------------------------------------------------------------------
| Index              : 1
| Previous BlockHash : 0000f66df4a46077837ae13f678809a3094ca1bdb9cb48c4312521a9e560dd0f
| BlockHash          : 0000355d0abe75dfa4747ea107904066e39703fdf379787c538312ad624fc421
| Nonce              : 71335
| Difficulty         : 4
| MerkleRoot         : 6d31afe4ef0ff7b5ecf21653e9da648c5c17101aca2fac0f4e744866de875007
| Timestamp          : 1684867152
|
|
| Transactions       :
|
| Transaction Index -> 1
| TransactionHash   -> 6d31afe4ef0ff7b5ecf21653e9da648c5c17101aca2fac0f4e744866de875007
| SenderKeyPair     -> 88221 274657
| RecieverrKeyPair  -> 130607 -455979
| Sent Amount       -> 15
|
--------------------------------------------------------------------------------
Enter the index(starting from 1) of valid Transaction in the block you want access to : 1

Choose from options :

1 . Change Sender's   Public Key

2 . Change Reciever's Public Key

3 . Change Sender's   Private Key

4 . Change sentAmout

5 . Exit

4
```

Figure 14 : Access to Block no.1's transaction no. 1 provided, because blockchain is a decentralized ledger

16 . Attack will be prevented and corrupted block will be removed

```
5 . Exit

4
Amount : 50

Choose from options :

1 . Change Sender's   Public Key

2 . Change Reciever's Public Key

3 . Change Sender's   Private Key

4 . Change sentAmout

5 . Exit

5
-------------------------------------------------------------------------------------
| Index              : 0
| Previous BlockHash : 0
| BlockHash          : 0000f66df4a46077837ae13f678809a3094ca1bdb9cb48c4312521a9e560dd0f
| Nonce              : 73846
| Difficulty         : 4
| MerkleRoot         : e7042ac7d09c7bc41c8cfa5749e41858f6980643bc0db1a83cc793d3e24d3f77
| Timestamp          : 1684866660
|
|
| Transactions       :
|
| Transaction Index -> 0
| TransactionHash   -> e7042ac7d09c7bc41c8cfa5749e41858f6980643bc0db1a83cc793d3e24d3f77
| SenderKeyPair     -> 0 0
| RecieverrKeyPair  -> 0 0
| Sent Amount       -> 0
|
-------------------------------------------------------------------------------------
^
|
|
^
|
X
|
Invalid block detected ! Connection broke !
|
X
|


***Exiting from Attack Mode***
```

Figure 15 : An attempt to make sent amount 50 instead of 15 is made , but change in verified data detected and prevented

Continue to explore !

4. Conclusion

In conclusion, the "BlockChainEd" project is a complex and challenging endeavor that requires a comprehensive understanding of blockchain technology, cryptography, consensus mechanisms, and security measures. The implementation of functionalities such as wallet creation, transaction handling, mining, block verification, and information inspection demands expertise in various programming languages and software development skills.

Throughout the project, several hurdles need to be overcome. These challenges include grappling with the intricacies of blockchain technology, ensuring robust security measures to protect againstattacks, addressing performance and scalability issues, designing a user-friendly interface, and conducting rigorous testing and debugging.

Despite these challenges, successfully implementing the "BlockChainEd" project can yield significant benefits. Users will be empowered to create their own blockchain, perform transactions securely through public key cryptography and ECC, and gain insights into decentralized environments. By allowing users to simulate and prevent attacks on their blockchain, the project facilitates a deeper understanding of blockchain vulnerabilities and countermeasures.

Overall, the "BlockChainEd" project serves as an educational tool, providing individuals with a hands-on experience in blockchain development. By gathering the necessary study and background knowledge and overcoming the associated challenges, developers can contribute to the advancement and adoption of blockchain technology, a transformative force with the potential to revolutionize various industries and foster a more decentralized and secure digital landscape.

References

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] Software Implementations and Applications of Elliptic Curve Cryptography, Kirill Kultinov/ Wright State University, 2019, 89

[3] Bitcoin: A Peer-to-Peer Electronic Cash System ,Satoshi Nakamoto/2008

[4] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In 20th Symposium on Information Theory in the Benelux, May 1999.

[5] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology, vol 3, no 2, pages 99-111, 1991.

[6] A. Back, "Hashcash - a denial of service counter-measure,"http://www.hashcash.org /papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980