International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

bChat: A Decentralized Chat Application

Sourabh[1], Deepanker Rawat[2], Karan Kapkoti[3], Sourabh Aggarwal[4], Anshul Khanna[5]

[1-4]Student, Dept. of Information Technology, Inderprastha Engineering College, Uttar Pradesh, India
[5]Professor, Dept. of Information Technology, Inderprastha Engineering College, Uttar Pradesh, India

---

Abstract - Sending messages always been a security concern over insecure channels. Although, there are numbers of techniques to encrypt the messages but still, there are possibilities to attack on the messages like Eavesdropping, MITM, EFAIL, etc. Many countries, like China and South Korea, Citizens are being tracked by their government (e.g., what are they chatting, sharing, etc). Moreover, the traditional application manages their data on centralized database which is also a concern. We present BChat, which ensures decentralization, immutability, censorship resistance and data security. The data which send by the users will directly added to the blockchain and creates the global copy of data in each node. Only the legitimate users can access that data by their private key on the blockchain. It eliminates the need for trusted intermediaries. The system is entirely decentralized and allows users to exchange message securely.

Key Words: Eavesdropping, Smart Contracts, Blockchain, INFURA, Solidity, SPF and Ethereum.

1. INTRODUCTION

Blockchain was designed to make transactions more secure. It is an inherent approach to achieve confidentiality, data integrity, authorization and relying on cryptography to achieve tamper-resistance.

It can be used to achieve the secure communication and integrity of data. It provides the feature of zero participation of 3rd party in the transaction. All network participants must reach a consensus to validate transactions in a secure way, and previous records cannot be changed. A very high cost must be spent if someone wants to alter previous records. External attackers would have to gain access to every computer in the network that hosts the blockchain database at the same time to manipulate it, which is as practically impossible.

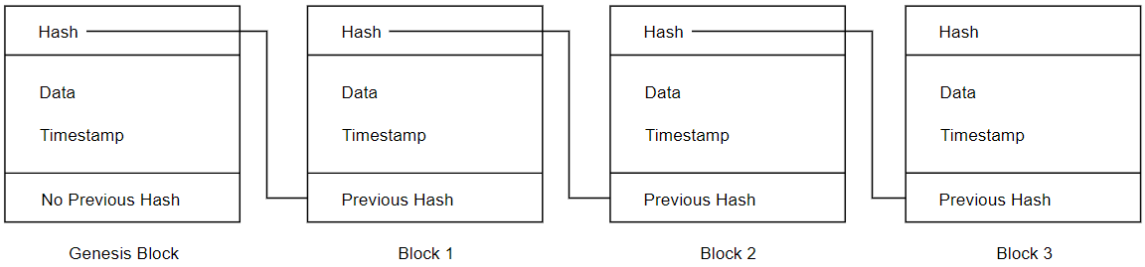

Fig 1. 1: Continuous sequence of blocks in a blockchain

The benefits of blockchain-based communications and data storage technology are numerous:

a. the stored data can be encrypted making it impervious to theft

b. the data is stored in a distributed manner, eliminating single-point failures

c. communications can be end-to-end encrypted making them spy-proof

d. the system is decentralized and less expensive than central storage of data

e. transactions and communications can be subject to incorruptible rules enforced by the blockchain itself

f. the system maintains a tamper and repudiation-proof history of past transactions and communications.

In our application, we are using the approach of decentralized application (DApp). All the user data is stored on a block which is connected to other blocks forming a chain. It is a peer-to-peer network. And, tampering the data which is stored on the blockchain is quite impossible because, of the encryption algorithm. If malicious user tries to make changes to the information in block then, he/she will have to make changes to all the copies of that block on whole blockchain network and that can be quite impossible. Though blocks are on all nodes, they cannot access the information in it, only the person for whom the information is concerned, they can only access.

The most important features are:

1. Easy and quick communication: The communication among the users is easy and quick. They can connect to the network just by their private keys.

2. Data immutability: The inability to make adjustments to the data after they are recorded in a distributed database.

3. Censorship resistance: It implies that everyone can transact with the network on the same terms, regardless of their personal identifying characteristics. If true censorship-resistance is to be achieved, then users should not be able to exclude others from information.

4. Decentralized storage: It is a model of network online storage where data is stored on multiple nodes (computer), which is hosted by the participants in the cloud.

5. Data Security: It means protecting digital privacy measures that are applied to prevent unauthorized access to the nodes.

2. BACKGROUND AND RELATED WORK

Over the last decade, WhatsApp, WeChat, etc, these traditional applications have taken all over the internet. There is a centralized server which stores all the information including identity to chats. Generally, these chat applications based on the following:

Centralized Management: In this management system, entire correspondence goes through the company's server which can govern its rules. Messages can be blocked on a certain subject or restrictions can be applied on the certain files.

Centralized Architecture: In this architecture, there is only single server which is maintaining all the services. It is allowing us to block access to a certain service for the whole country whereas, the problems on the management servers may lead to inadequacy of the service for all or a significant part of users.

Confidentiality: Confidentiality of a user can be compromised on the request of government.

Single Point of Failure (SPF): If a single node fails then whole application can be compromised.

The above encouraged us to build an application where, we can have all the features like: Decentralized storage, Censorship resistance, Data security and Data immutability.

3. IMPLEMENTATION

The decentralized application is implemented on Ethereum blockchain network with ReactJS.

Refer code: https://github.com/samcracker/bChat

A. Ethereum: A platform for decentralized applications

Ethereum is an open source, public, blockchain based distributed computing platform and operating system featuring smart contract functionality. It supports a modified version of Nakamoto consensus.

Fig 3. 1: Ethereum

Proof of Stake:

• This algorithm which aims to achieve distributed consensus in a blockchain.

• A stake is value (money), we bet on a certain end product. This process is called as staking.

Why Proof-of-Stake?

Before this, the most popular way to achieve distributed consensus was through Proof-of-Work which is implemented in Bitcoin. But Proof-of-Work based consensus mechanism states that if it has more computation resources then the chances of mining a new block by the miner is very high.

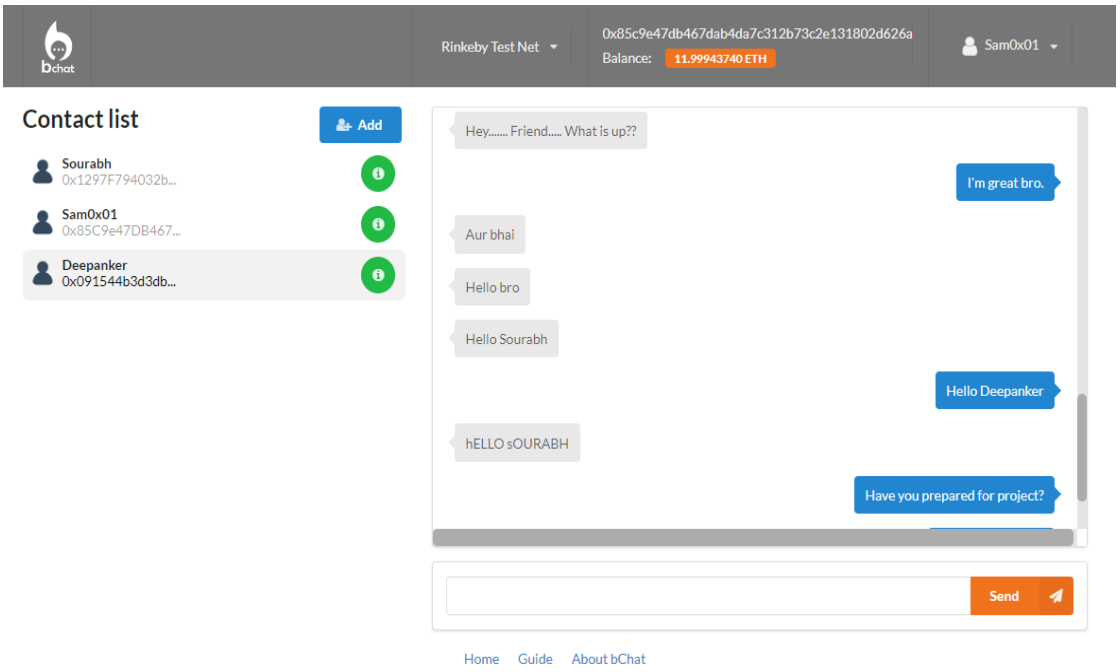

Fig 3. 2: bChat

B. INFURA

Infura is a hosted Ethereum node cluster that lets your users run your application without requiring them to set up their own Ethereum node or wallet.

INFURA

Fig 3. 3: Infura

Why Infura?

There are lots of pain points being faced by blockchain which may be solved by Infura or other InterPlanetary File System (IPFS), to some extent. These are the main challenges:

1. It's expensive to store data on the Ethereum blockchain.

2. It's tough to configure an Ethereum geth client.

3. It's tough to scale the infrastructure.

C. SOLIDITY

Solidity is an object-oriented, high-level language for implementing smart contracts. Smart Contracts are programs which govern the behaviour of accounts within the Ethereum state.

It's an agreement or set of rules that govern a business transaction. It's stored on the blockchain and is executed automatically as part of a transaction. It allows transaction to be carried out without the need for a governance, legal system, central authority or external enforcement mechanism.

SOLIDITY

Fig 3. 4: Solidity

Features:

1.Trust 5.Savings

2. Autonomous 6. Speed

3. Security 7. Transparency

4. Redundancy 8. Precision

D. ETHERSCAN

Etherscan allows to explore and search the Ethereum blockchain for transactions, addresses, tokens, prices and other activities.

Fig 3. 5: Etherscan

E. Decentralized Application Architecture

- The front-end of Decentralized application is generally built on ReactJS. It's an open source and component -based JavaScript library used for creating dynamic and interactive user interfaces, especially for single page application.
- The backend of the decentralized application is Ethereum network. The blocks of network contain the information about transactions. The smart contract which is written in Solidity is deployed on Rinkeby test network.
- When the user sends the message to the particular user's address. The message block will be added to the Ethereum network according to the given prescribed contract.

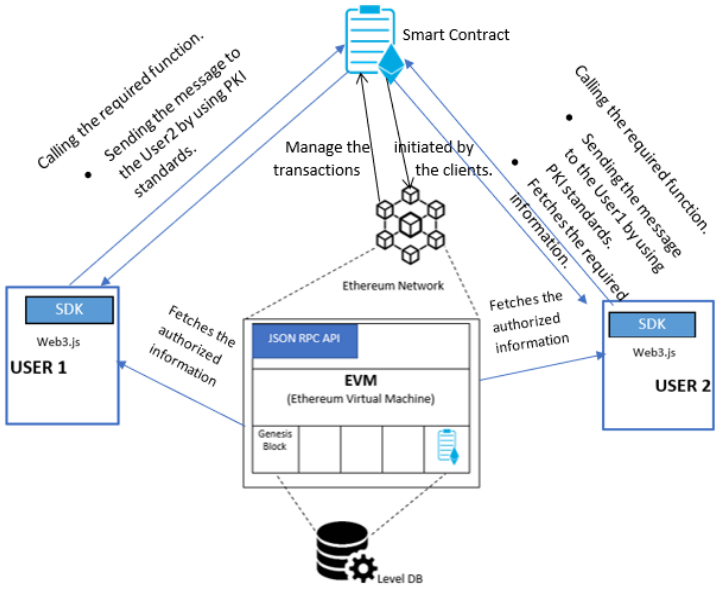

Fig 3. 6: Architecture of bChat

F. Cryptography: Encryption and Decryption of messages

- The algorithm is used "AES-256" for encryption and decryption using CryptoJS. It's available in the npm package manager.
- The inbuilt library which is used for encrypting the messages. First the messages are encoded into hex and then passing through the encryption function with the algorithm and secret key. And, the same process is used in decryption.

G. SMART CONTRACT COMPILATION

- The smart contract is built on the solidity version 0.4.22.
- It contains mainly 5 functions: join contract, send message, add address, accept address and block address.
- The smart contract is tested using the JavaScript. Testing of each function is very necessary. Smart can't be deployed without testing the contract's functions or modules.
- If it's done and smart contract having some logical errors or security errors then it can compromise the whole application.

```
function addContract(address addr) public onlyMember{
    require(relationships[msg.sender][addr] == RelationshipType.NoRelation);
    require(relationships[addr][msg.sender] == RelationshipType.NoRelation);

    relationships[msg.sender][addr] = RelationshipType.Requested;
    emit addContactEvent(msg.sender, addr);
}
function acceptContactRequest(address addr) public onlyMember {
    require(relationships[addr][msg.sender] == RelationshipType.Requested);
    relationships[msg.sender][addr] = RelationshipType.Connected;
    relationships[addr][msg.sender] = RelationshipType.Connected;

    emit acceptContractEvent(msg.sender, addr);
}
function join(bytes32 publicKeyLeft, bytes32 publicKeyRight) public {
    require(members[msg.sender].isMember == false);
    Member memory newMember = Member(publicKeyLeft, publicKeyRight, "", "", 0, true);
    members[msg.sender] = newMember;
}
```

Fig 3. 7: Snippet of smart contract

H. DEPLOYMENT

- When the compilation is done successfully. Now, it's the time to deploy the smart contract on blockchain network.
- This application uses the INFURA API for deployment of smart contract.
- The smart contract is deployed on Rinkeby test network.

4. ADVANTAGES

The decentralized chat application will have desirable and incredible advantages like:

1. Military Professionals: It'll be very useful for the government security officials where they can send messages direct to their intenders without any 3rd party involvement. They don't need to trust any application.

2. Censorship resisted: It'll also helpful where citizens can't have Right to Speech. Even, where their government is tracking and monitoring them.

5. CONCLUSION

The messages are being sent over insecure channels using end-to-end encryption. When the message block is added to the blockchain then, it'll never be changed. If malicious user tries to make changes to the information in block then, he/she will have to make changes to all the copies of that block on whole blockchain network and that can be quite impossible. Though blocks are on all nodes, they cannot access the information in it, only the person for whom the information is concerned, they can only access.

REFERENCES

[1] Decentralized Chat Application using Blockchain Technology Abhishek P. Takale, Chaitanya V. Vaidya, Suresh S. Kolekar Rajendra Mane College Of Engineering And Technology, Ambav, India.
[2] Cooperative Storage Cloud https://en.wikipedia.org/wiki/Cooperative_storage_cloud
[3] Secure Peer-to-Peer communication based on Blockchain Kahina Khacef, Guy Pujolle.
[4] Ethereum Developer Resources: https://ethereum.org/developers/
[5] Ethereum documentation: http://www.ethdocs.org/en/latest/index.html.

[6] Hands On: Get Started with Infura and the IPFS on Ethereum
[7] Solidity documentation: https://solidity.readthedocs.io/en/v0.6.5/
[8] Infura documentation: https://infura.io/docs