USAGE OF BLOCKCHAIN TECHNOLOGIES AND SMART CONTRACTS FOR SECURED INTERNET BANKING

[1]BHAVNEET SINGH, [2]BHUPINDER KAUR

[1]Yes Bank YamunanagarIndia, [2]Meshed GroupSydney Australia
E-mail: [1]sbhavneet13@gmail.com, [2]kaurb2605@gmail.com

Abstract - With the increasing number of devices and gadgets using network based communication, the vulnerabilities are elevating to a huge levels. There is need to work out on the secured mechanisms so that the transactions in the network channels can be made secured. To cope with such scenarios, the usage of blockchain technology is quite prominent so that the peer to peer based secured communication can be done. In current scenario, the blockchain technology is more focused towards cryptocurrencies in which the distributed ledger is maintained for the transactions. The distributed ledger refers to the replicated, synchronized and shared digital asset to multiple locations and devices so that the third party manipulation cannot be possible. For example, if a bank follows the distributed database ledger with blockchain technology can enforce higher degree of security. If that bank is having one million customers then the records of the transactions will be stored on those one million devices. It refers to the fact that the hacker will have to hack one million devices in real time rather than a single server. This is the major advantage of using the decentralized blockchain technology. In case of centralized application, if hacker penetrates the server of a bank, then all the details and records of all the customers can be copied. That is the main reason because of which the government agencies should focus on decentralizing their web based applications.

Keywords - Blockchain Security, Data Security, High Availability Data, Secured Databases.

I. INTRODUCTION

In the present period, Blockchain Technology is one of the key zones of research just as execution explicitly in the space of Cryptocurrency. Presently days, various computerized digital currencies are very conspicuous and shared all through the world regardless of enormous analysis and contentions [1, 2]. These cryptographic forms of money incorporate BitCoin, Ethereum, LiteCoin, PeerCoin, GridCoin, PrimeCoin, Ripple, Nxt, DogeCoin, NameCoin, AuroraCoin, Dash, Neo, NEM, PotCoin, TitCoin, Verge, Stellar, VertCoin, Tether, Zcash and numerous others. These blockchain based digital currencies don't have any halfway bank or installment door to record the log of the exchanges [3, 4, 5]. That is the principle reason as a result of which numerous nations are not permitting the digital forms of money as legitimate cash exchange [6, 7, 8]. All things considered, these blockchain based digital currencies are celebrated and utilized as a result of immense security highlights [9, 10, 11]. The blockchain organize is having a square of records in which every single record is related with the dynamic cryptography so every one of the exchanges can be encoded with no likelihood of sniffing or hacking endeavors [12, 13, 14].

Using blockchian technology, the servers of government for land registry, citizen information, Permanent Account Number (PAN) and many others can be made secured using decentralized apps [15, 16, 17].

The blockchain based decentralized application can be used for following

- Digital Identity Management in of Government Documents
- (Birth, Marriage and Death) Certificates
- Asset and Land Registry
- Notarized Documents
- Incorporation Services
- Taxation and Financial Records
- Personalized Government Services
- Polling / Voting / Assembly Elections

Social Welfare and Benefits

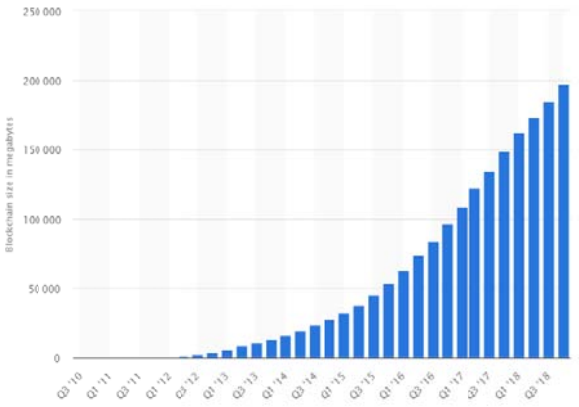


Figure 1: Blockchain Size from 2010 to 2019

As per the research analytics and reports from Statista.com, the size of BitCoin blockchain from year 2010 to current year 2019 and that is having enormous use throughout the globe.

II. DECENTRALIZED APPLICATIONS (DAPP) WITH SECURED BLOCKCHAIN

The decentralized application (dApp) refers to the software application that executes on the distributed channels so that the hacking of application will be near to impossible. In traditional centralized application, the application is deployed on a single

server [18, 19, 20]. The main limitation with centralized approach is that if that centralized server is hacked then everything can be damaged or copied from that server. In case of decentralized application, there is no single server rather storage is done on all the client devices so that the replication of the transaction can be done with maximum availability of transaction records.

In the scenario of decentralized application, the hacker will have to crack all the devices associated with that application and that will be very difficult in real time using smart contracts. In smart contracts based dApp, the dynamic token sharing is implemented so that the transactions will have maximum security measures [21, 22].

2.1. Smart Contracts

The smart contract programming is required for the globalization based transactions. It means that the transactions can be done across the people who can't communicate because of different continents, languages and traditions. Smart Contracts automatically validate the transactions and business dealings between the people who can't understand the language of each other.

The smart contract programming is required for the globalization based transactions. It means that the transactions can be done across the people who can't communicate because of different continents, languages and traditions. Smart Contracts automatically validate the transactions and business dealings between the people who can't understand the language of each other [23, 24].

Free and Open Source Tools for Blockchain Development

Hydrachain
URL: https://github.com/HydraChain/hydrachain
- Creation of Permissioned Distributed Ledgers
- Setup of Private Chain
- Fully Compatible with Ethereum Protocol

Multichain
URL: https://www.multichain.com/
- Compatible with Bitcoin
- Fully Customizable
- Fine-Grained Managed Permissions
- Rapid Creation and Deployment of New Blockchain
- Powerful Data Sharing and Encryption

OpenChain
URL: https://www.openchain.org/
- Digitally Signed transactions
- Custom rule definitions for ledgers
- Robustness and Fine Validation
- Client Server Architecture
- Module Design with Real Time Validations
- Immutability with Security

Ethereum
URL: https://www.ethereum.org/
- Smart Wallets and Smart Money
- Creation of Own Cryptocurrency Development
- Security against third party intervention or downtime
- Execution of Smart Contracts
- Virtual Shares with Crowd Fund and Crowd Sale

Corda
URL: https://www.corda.net/
- Platform for Blockchain and Distributed Ledger
- Smart Contracts
- Development of Distributed Apps
- Notary Infrastructure for Sequencing and Validation of Transactions
- Flow based Framework for Negotiation and Communication in the participants

Credits
URL: https://credits.com/
- Smart Contract Programming
- Real Time Monitoring of Network Transactions
- Web Wallet with Security using Private and Public Keys

BigChainDB
URL: https://www.bigchaindb.com/
- Big Data enabled Blockchain Database
- Decentralized Management and Control
- Dynamic Management of Digital Assets
- Byzantine Fault Tolerant (BFT) for high performance computing applications
- Rich Permissioning at each Transaction
- Integration with MongoDB NoSQL for fast transaction processing with unstructured data
- Resistance to Tamper and Faults for Security

Quorum
https://www.jpmorgan.com/global/Quorum
- Enterprise Level Smart Contract and Distributed Ledger Platform
- Peer Permissioning
- High Performance using Raft based Consensus
- Exchange of Private Messages with Secured Contracts
- Fully Customizable for Large Scale Business and Corporate Applications
- Integration with CakeShop Software Development Kit (SDK) for Graphical User Interface (GUI) enabled Smart Contracts, Quorum Networks and APIs.

Symbiont Assembly
URL: https://symbiont.io/
- Byzantine Fault-Tolerance

- Handling Thousands or more Transactions Per Second
- Elimination of data loss with storage of real time critical documents on network
- Sharing of data with advanced encryption and dynamic cryptography
- Maintenance of transaction logs and lifetime without third party intervention

Embark
URL: https://embark.status.im/
- Peer to Peer Secured Messaging
- Development and Distribution of Decentralized Apps with Decentralized Communication with Orbit and Whisper
- Integration with Web Technologies including Foundation, React, Angular and others
- Custom Framework Development
- Association with Simulated Blockchains including Ganache
- Automatic Smart Contract deployment
- Integration with EthereumBlockchains
- Testing Environment for Smart Contracts using Web3

Solidity
URL: https://github.com/ethereum/solidity
- High Level Programming Language for Smart Contracts
- Contract Oriented Statically Typed Programing Language
- Object Oriented with support to multiple blockchian platforms
- Compatible with Ethereum, Tendermint, Counterparty and ErisDB
- Creation of Smart Contracts for Crowdfunding, Muti-Signature Wallets, Voting, Blind Auctions and many others

Truffle
URL: https://truffleframework.com/
- Platform for Smart Contract Programming with compilation, linking and binary management
- Development Environment with Framework for Testing
- Testing of Contracts using Chai library and Mocha Framework
- NodeJS support for Mocha and Chai integration for unit testing

Solidity is one of the powerful and high performance programming language for writing smart contracts. It follows object oriented programming paradigm with higher degree of security and performance which can be integrated with assorted blockchain platforms.

The code of solidity is compiled and transformed to bytecode which is executed on Ethereum Virtual Machine (EVM). Solidity Programming is having the key base of multiple programming languages and scripts including Python, JavaScript, C++ so that it can be integrated to multiple environments and platforms for integration with blockchains. To work with Solidity Programming, there are many Integrated Development Environments (IDEs) and Editors which can be used including Remix, EthFiddle, JetBrains and many others [25, 26, 27].

To start with Solidity Programming, Remix is one of the powerful IDEs that is open source and also provides the web based interface. The web based interface of Remix IDE is easy for the developers to create the Smart Contracts with Blockchain Programming [28, 29, 30].

The URL of Web Based Remix IDE is remix.ethereum.org that can be accessed directly on the web browsers for writing, compiling and executing the smart contracts.

Following is the scenario of Blockchain integrated currency with the Internet Banking Transactions so that the overall communication and financial transaction will be secured and sniffers free.

```
pragma solidity ^0.4.18;
contractNewCryptoCurrency {
string public name = 'NewCryptoCurrency';
// Name of the New Currency
string public currencyName = 'Currency1.0';
// Select Currency
mapping (address =>uint) EBankingBalance;
// Key-Value Pair for Address-Account
event Transfer(address _sender, address _receiver,
uint256 _value);
// Log Recording
constructor() public {
// Constructor on Creating the Contract
EBankingBalance[msg.sender] = 100000;
// Balance Confirmation
}
functionsendAmount(address _receiver, uint
_amount) public returns(bool sufficient) {
if (EBankingBalance[msg.sender] < _amount) return
false;
// Authentication of the Transfer
EBankingBalance[msg.sender] -= _amount;
EBankingBalance[_receiver] += _amount;
emit Transfer(msg.sender, _receiver, _amount);
// Commit of Payment Transfer with Transaction
Recording
return true;
}
functiongetBalance(address _addr) public view
returns(uint) {
// Checking the Balance
returnEBankingBalance[_addr];
}
}
```

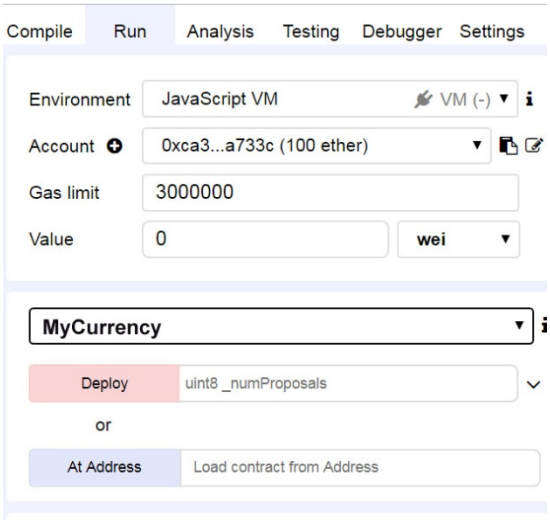


Figure 2: Compile and Run Code in Remix IDE

On click of Deploy option in Remix IDE, the code is executed and detailed logs of the transaction can be viewed.

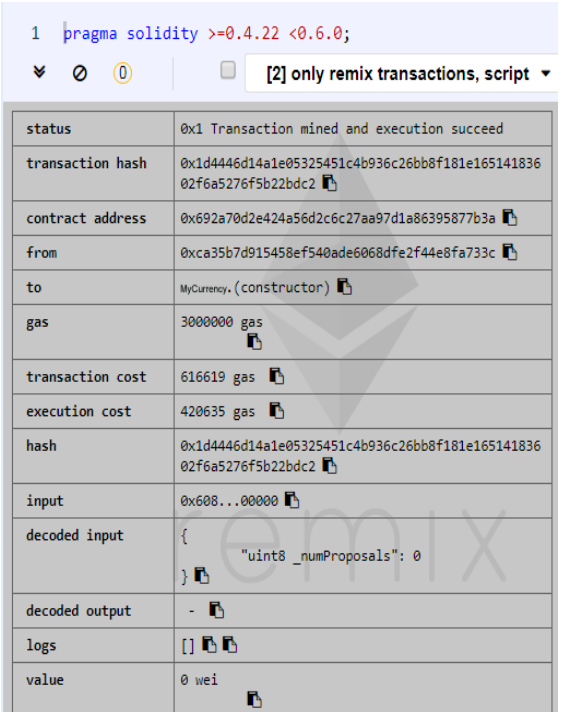


Figure 3: View Logs and Transaction Details

The detailed logs of the Blockchain Transaction can be analyzed associated with the transaction. There are multiple parameters in the transaction log including Gas Limit. The Gas Limit in Smart Contract Programming refers to the amount of work or throughput associated with the transaction [31].

2.2 NodeJS and Web3JS

NodeJS is a cross-platform open source platform for JavaScript based programing. It can be used for the development of multiple applications including Blockchain Development, Smartphone Applications, Distributed Web Applications, NoSQL Processing, Big Data Analytics, Machine Learning, Internet of Things (IoT) and many others related to advanced computing.

For blockchain programming, NodeJS is integrated with Web3JS. Web3JS refers to the set of libraries and tools for interaction with blockchain based connections [32].

The Web3JS Platform for blockchains and dApp can be integrated with following instructions

npm: npm install web3
purejs: link the dist/web3.min.js
meteor: meteor add ethereum:web3

After installation, the code in JavaScript and Server Side Scripts is written with the smart contracts and deployment for the secured applications.

Truffle is another programming suite for the development of blockchain based smart contracts that can be installed with NodeJS and can be executed on Ethereum Virtual Machine (EVM).

With the execution of following instruction, the Truffle Suite is associated with NPM

npm install -g truffle

After installation of Truffle, the versions of installed version can be checked in the terminaltruffle version
The new project EffBlockchain in Truffle can be mapped as
mkdirEffBlockchain
cdEffBlockchain
truffleEffBlockchain

In Truffle, the following directory structure is followed to code the application

- contracts/: Source Code for the Smart Contracts
- migrations/: Migration System and Handlers for the Smart Contracts
- test/: Tests and JavaScript Code
- truffle.js: Configuration File for Truffle
- EffBlockchain: Additional Folders and Files required for coding the blockchain

A new file <filename.sol> is created in contracts/ directory for the base coding of smart contracts in the following format

```
pragma solidity ^0.5.0;
contractMysmartcontract {
}
Sending Values
function adopt(uintMyVar) public returns (uint) {
require(MyVar>= 0 &&MyVar<= 15);
adopters[MyVar] = msg.sender;
returnMyVar;
}
```

The compilation of Truffle Code is done as followstruffle compile

The framework of Embark provides the tools and libraries for development of decentralized apps so that blockchain based implementation can be done. Embark can be used as an alternate to Truffle. To work with Embark, there is need to integrate Node Version Manager (NVM) having multiple versions of NodeJS.

Using "embark run", the dashboard of Embark is invoked

The token generation can be further written in code

```
pragma solidity ^0.4.25;
import "openzeppelin-solidity/contracts/token/ERC20/ERC20.sol";
contractCurrencyToken is ERC20 {
string public name = "CurrencyToken";
string public SymbolCurrency = "SYMBOLCURRENCYCURRENCY";
uint256 public decimals = 18;
constructor() public {
   }
}
```

As in the above example, the new currency can be mapped with the transaction with the secured communication with network based transactions. It can be anything as per the requirements of the smart contract associated with the blockchain.

CONCLUSIONS

Now days, the blockchain based implementations are used in many international banks and the related transactions so that the communication can be made secured. As blockchain based development is the emerging domain of research, there is need to work out different issues related to privacy and resource optimization. In blockchain and decentralized applications, the data is replicated to enormous devices and the issues of security and integrity arise. With the development and deployment of advanced algorithms, the performance of blockchain based implementations can be elevated.

REFERENCES

[1] Bogner, A., Chanson, M., &Meeuw, A. (2016, November). A decentralised sharing app running a smart contract on the ethereum blockchain. In Proceedings of the 6th International Conference on the Internet of Things (pp. 177-178). ACM.

[2] Bahga, A., &Madisetti, V. K. (2016). Blockchain platform for industrial internet of things. Journal of Software Engineering and Applications, 9(10), 533.

[3] Yuan, Y., & Wang, F. Y. (2016, November). Towards blockchain-based intelligent transportation systems. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) (pp. 2663-2668). IEEE.

[4] Warren, W., &Bandeali, A. (2017). 0x: An open protocol for decentralized exchange on the Ethereum blockchain. URl: https://github.com/0xProject/whitepaper.

[5] Zhang, P., Walker, M. A., White, J., Schmidt, D. C., & Lenz, G. (2017, October). Metrics for assessing blockchain-based healthcare decentralized apps. In 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom) (pp. 1-4). IEEE.

[6] Griggs, K. N., Ossipova, O., Kohlios, C. P., Baccarini, A. N., Howson, E. A., &Hayajneh, T. (2018). Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. Journal of medical systems, 42(7), 130.

[7] Sreehari, P., Nandakishore, M., Krishna, G., Jacob, J., &Shibu, V. S. (2017, July). Smart will converting the legal testament into a smart contract. In 2017 International Conference on Networks & Advances in Computational Technologies (NetACT) (pp. 203-207). IEEE.

[8] Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2017). A survey on the security of blockchain systems. Future Generation Computer Systems.

[9] Asharaf, S., &Adarsh, S. (Eds.). (2017). Decentralized Computing Using Blockchain Technologies and Smart Contracts: Emerging Research and Opportunities: Emerging Research and Opportunities. IGI Global.

[10] Huckle, S., Bhattacharya, R., White, M., &Beloff, N. (2016). Internet of things, blockchain and shared economy applications. Procedia computer science, 98, 461-466.

[11] Swan, M. (2015). Blockchain: Blueprint for a new economy.

[12] "O'Reilly Media, Inc.".

[13] Zhang, P., White, J., Schmidt, D. C., Lenz, G., & Rosenbloom, S. T. (2018). Fhirchain: applying blockchain to securely and scalably share clinical data. Computational and structural biotechnology journal, 16, 267-278.

[14] Rodrigues, B., Bocek, T., & Stiller, B. (2017). Enabling a cooperative, multi-domain DDoS defense by a blockchain signaling system (BloSS). In Proceedings of the 42nd IEEE Conference on Local Computer Networks.

[15] Karamitsos, I., Papadaki, M., & Al Barghuthi, N. B. (2018). Design of the blockchain smart contract: A Use Case for Real Estate. Journal of Information Security, 9(03), 177.

[16] Zhang, P., Schmidt, D. C., White, J., & Lenz, G. (2018). Blockchain technology use cases in healthcare. In Advances in Computers (Vol. 111, pp. 1-41). Elsevier.

[17] Zhang, P., White, J., Schmidt, D. C., & Lenz, G. (2017). Applying software patterns to address interoperability in blockchain-based healthcare apps. arXiv preprint arXiv:1706.03700.

[18] Swan, M. (2016, July). Blockchain temporality: smart contract time specifiability with blocktime. In International symposium on rules and rule markup languages for the semantic web (pp. 184-196). Springer, Cham.

[19] Bahga, A., &Madisetti, V. (2017). Blockchain Applications: A Hands-On Approach. VPT.

[20] Yuan, Y., & Wang, F. Y. (2018). Blockchain and cryptocurrencies: Model, techniques, and applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48(9), 1421-1428.

[21] Wohrer, M., & Zdun, U. (2018, March). Smart contracts: Security patterns in the ethereum ecosystem and solidity. In 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE) (pp. 2-8). IEEE.

[22] Swan, M. (2015, March). Blockchain thinking: The brain as a dac (decentralized autonomous organization). In Texas Bitcoin Conference (pp. 27-29). Chicago.

[23] McCorry, P., Shahandashti, S. F., &Hao, F. (2017, April). A smart contract for boardroom voting with maximum voter privacy. In International Conference on Financial Cryptography and Data Security (pp. 357-375). Springer, Cham.

[24] Voshmgir, S. (2016). Blockchains, Smart Contracts und das Dezentrale Web. Technologiestiftung Berlin, Blockchains, Smart Contracts und das Dezentrale Web, 17-35.

[25] Swan, M. (2015). Blockchain thinking: The brain as a decentralized autonomous corporation [commentary]. IEEE Technology and Society Magazine, 34(4), 41-52.

[26] Walsh, C., OReilly, P., Gleasure, R., Feller, J., Li, S., & Cristoforo, J. (2016). New kid on the block: a strategic archetypes approach to understanding the Blockchain.

[27] Dai, P., Mahi, N., Earls, J., &Norta, A. (2017). Smart-contract value-transfer protocols on a distributed mobile application platform. URL:https://qtum.org/uploads/files/cf6d69348ca50dd985b60425ccf282f3.pdf.

[28] Bashir, I. (2017). Mastering blockchain. Packt Publishing Ltd.

[29] Egbertsen, W., Hardeman, G., van den Hoven, M., van der Kolk, G., & van Rijsewijk, A. (2016). Replacing paper contracts with Ethereum smart contracts.

[30] Kounelis, I., Steri, G., Giuliani, R., Geneiatakis, D., Neisse, R., &Nai-Fovino, I. (2017, July). Fostering consumers' energy market through smart contracts. In 2017 International Conference in Energy and Sustainability in Small Developing Economies (ES2DE) (pp. 1-6). IEEE.

[31] Filipova, N. (2018). Blockchain–An Opportunity ForDeveloping New Business Models.

[32] Sylim, P., Liu, F., Marcelo, A., &Fontelo, P. (2018). Blockchain Technology for Detecting Falsified and Substandard Drugs in Distribution: Pharmaceutical Supply Chain Intervention. JMIR research protocols, 7(9).

[33] Rifi, N., Rachkidi, E., Agoulmine, N., &Taher, N. C. (2017, October). Towards using blockchain technology for eHealth data access management. In 2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME) (pp. 1-4). IEEE.

★★★