International Journal of Scientific Research in Engineering & Technology
Volume4, Issue6 (November-December 2024), PP: 23-26.
www.ijsreat.com ISSN No: 2583-1240

Decentralized Security Solutions for IoT Using Ethereum

Mohit Agrawal[1], Monika Jareda[2], Pradhum Malhotra[3], Shreya Meena[4], Bharat Modi[5]
[1,2,3,4] Undergraduate Students, Department of Computer Science Engineering, Swami Keshvanand Institute of Technology Management and Gramothan, Jaipur, Rajasthan, India.
[5]Associate Professor, Department of Computer Science Engineering, Swami Keshvanand Institute of Technology Management and Gramothan, Jaipur, Rajasthan, India.

To Cite this Article: *Mohit Agrawal[1], Monika Jareda[2], Pradhum Malhotra[3], Shreya Meena[4], Bharqt Modi[5], "Decentralized Security Solutions for IoT Using Ethereum", International Journal of Scientific Research in Engineering & Technology, Volume 04, Issue 06, November-December 2024, PP: 23-26.*

Abstract: *The current IoT system has a security risk since all data is kept in a single location and all computing activities are done through a central server. It is possible for data tampering and a single point of failure in a centralized IoT system. When the primary server fails, the centralized IoT paradigm creates a single point of failure because a centralized system manages all IoT data from several linked devices. It is a clear target for privacy and security concerns. To overcome these problems, the Ethereum Blockchain Technology is considered in this chapter for securing the data through distributed and decentralized ways. The pre-processing for Ethereum node creation from a genesis block, smart contract deployment, and performance metrics are presented*

Key Word: *Ethereum Blockchain, IoT, genesis block, smart contract deployment*

I.INTRODUCTION

Ethereum is a popular blockchain implementation that serves as a platform for smart contract execution .These smart contracts can help with financial transactions and data storage on a distributed ledger .Gas and ether are two forms of Ethereum currency. Gas is the ether that smart contract nodes must pay to run them. Ether is the Ethereum crypto-currency that may send money and pay for gas to operate smart contracts.

Similar to traditional paper contracts, smart contracts define the conditions of an agreement between two parties. They automatically execute when the terms are met and eliminate the need for either party to know who is on the other side of the transaction or for an intermediary. Ethereum introduces several concepts to enable, sending and receiving ether, executing smart contract transactions, and reading data to and from the blockchain in general. To overcome the drawback of the IoT issues and challenges, Ethereum based Blockchain network has been implemented. The Smart Contracts were created in Solidity 0.6.0., Ethereum Nodes are connected with the Geth (Go Ethereum) v1.10.2 CLI tool and Truffle (Node.js app) 5.1.30 is used to build, compile and deploy Solidity-based Smart Contracts.

II.GENESIS BLOCK

Figure 1 Genesis Block in Ethereum Network

The genesis block is the first block of the entire chain and is named block 0 in the Ethereum network. Each block in the blockchain is linked to the preceding block. Since the genesis block is the beginning of a block, there are no links behind it. It has to be created manually for ethereum networks and contains a hash of zeros. This block is represented in the form of a JSON file

The Ethereum network was built using a genesis file, as shown in Figure 1. The genesis block contains all the essential information to configure the network as well as find related peers. It is the config file for the Ethereum network. The genesis file is a simple JSON file that contains config thresholds.

Some of the fields in the genesis state file are as follows:

Config:

The file starts with the "config" block, which contains all the config parameters and thresholds that control the networks basic operations.

Chain ID:

It protects the network from a replay attack. It acts as an offset to prevent attackers from deciphering continuous values in network.

Homestead Block:

Homestead is the second major release of Ethereum (the first release is Frontier). If the value is 0, Homestead is used.

EIP150Block:

EIP stands for Ethereum Improvement Proposal. Ethereum is open-source so that proposals can be made in discussions and code, and EIP150 is one such proposal that was accepted. This EIP took effect on block 2463000 and mainly increased gas prices in response to denial-of-service concerns.

EIP150Hash:

The hash of the EIP150Block, which is needed for fast sync.

Difficulty:

This determines how difficult it is to mine in the Ethereum network. It is set to the lowest possible value when developing on a test network, so waiting time is reduced for mining blocks.

Nonce & mix hash:

Nonce and mix hash are values, it allows to verify that a block has been cryptographically mined, and it is validated. The mix hash is a 256-bit hash which proves when combined with the 64-bit nonce that a sufficient amount of computation has been carried out on this block: the Proof-of-Work (PoW). Mix hash is added with the block header hash to form the complete block hash. It is not relevant to a new private network, so it is set to 0.

III. PUBLIC ETHEREUM NETWORK

Genesis file is used to start a public Ethereum network. For a blockchain to operate, each Ethereum node must have the same network ID.

Table 1 Public Ethereum Networks

Network	Type	Network ID	Network Status
main	Main	1	Online
morden	Test	2	Retired
ropsten	Test	3	Online
rinkeby	Test	4	Online

Table 1 shows that network ID 1 is reserved for the Ethereum main chain. In contrast, network ID 2 is reserved for the late Morden test chain, network ID 3 is reserved for the Ropsten test chain, and network ID 4 is reserved for the Rinke by test chain.

IV. SMART CONTRACT

The smart contract is designed and implemented using the Remix IDE and Geth Console, an extensive, open development toolset and framework to facilitate the implementation of blockchain applications.

As shown in Figure 2, the Smart Contract has various features that allow users to connect with the ledger, combining the state database and the Blockchain. Smart contracts are implemented in a blockchain with unique addresses, which means that transactions are signed off by nodes and addressed to smart contracts themselves to invoke a feature written in a smart contract. The block contains the transaction hash value and the previous block hash value to ensure data consistency in the ledger. If the ledger hosted by one peer tampers, it will not convince all the other peers; because the ledger is distributed across the network. The transaction is appended to the block, and the ledger state is updated. In the end, the updated result for the ledger is returned as a response to the question.

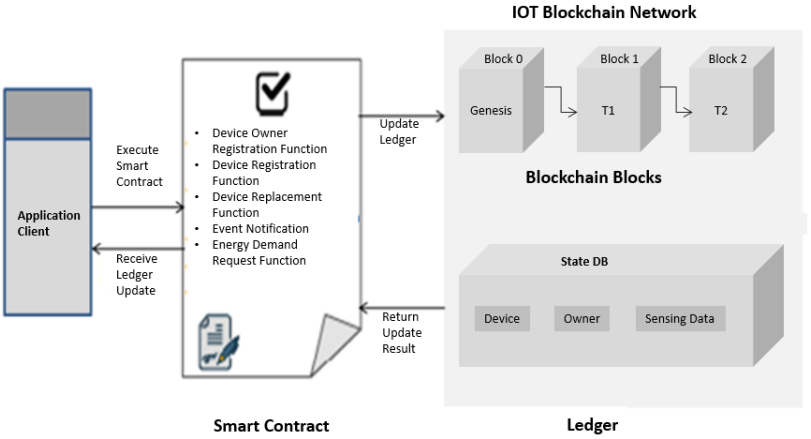


Figure 2 Smart contract network interactions

V. TOOLS USED

Geth

Geth (Go Ethereum) is a Go-based command-line interface for running an Ethereum node. Geth allows to join the Ethereum network and send ether between accounts. For external listening, Geth uses port 30303. 8545 is the default port for internal communication, such as between wallet and Geth. Geth 1.10.13-stable version was used to connect to the Ethereum network through the command line

Web3J

Web3J is a highly flexible, reactive, and type-safe Java and Android toolkit for dealing with smart contracts and interfacing with Ethereum network clients (nodes). Web3J 4.8.2 is used to communicate with Ethereum nodes. Table 3.2 describes a few commands that allow geth to attach peer nodes and start the mining process.

Solidity

Solidity is a statically typed programming language used to create smart contracts on the Ethereum Virtual Machine (EVM). Smart contracts are decentralized programs that operate on a peer-to-peer network with no central authority and allow for the implementation of value tokens, ownership, voting, and other logic.

Remix IDE

Smart contracts written in the Solidity language are simulated using the Remix IDE. The remix is a browser-based IDE that can create and test smart contracts and aids to create and execute a smart contract. Moreover, several popular IDEs and text editors have plugins and frameworks for creating and testing Smart Contracts.

Truffle

Truffle is a smart contract deployment tool based on Ethereum. Truffle 5.1.50 was used to deploy the smart contract on the Ethereum Network. Truffle is also used to get the smart contract address and specify the ABI. A few commands that allow truffle to compile and deploy smart contracts are listed

MetaMask

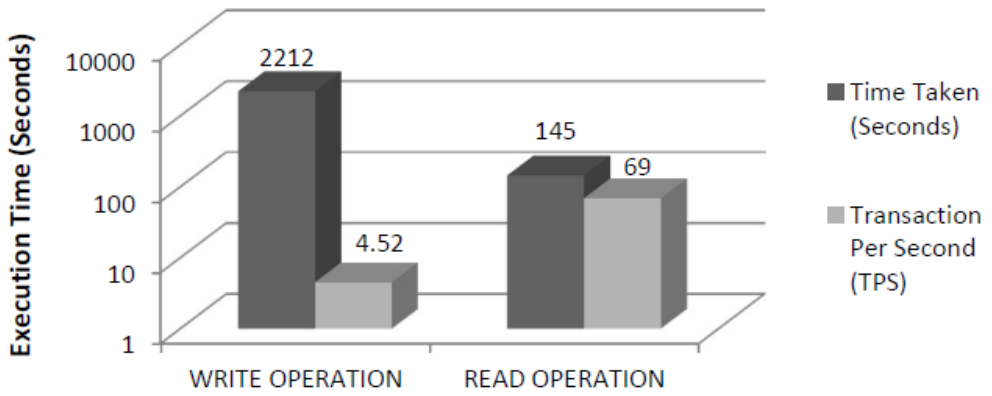
MetaMask is a cryptocurrency wallet that works with the Ethereum network through software. It lets users utilize a browser extension or a mobile app to access their Ethereum wallet, which can subsequently engage with decentralized apps. MetaMask 8.1.3 browser extension was used for the Truffle with Remix link.

VI. RESULTS

Data security in IoT systems using Ethereum Network with Geth console. As shown in Figure 5, accounts and private keys are created in Ethereum Network using the Test RPC Ethereum emulator. The Genesis File initialization and account address creation in Ethereum Network are shown in Figure 6 and Figure 7. The block information and transaction finally, the Performance metrics of the Write/Read operation for 10k transactions in the Ethereum Network captured are shown in Figure

VII. PERFORMANCE EVALUATION

The performance of the ethereum network was captured with the write and read test operations 10k transactions. The performance metrics are verified in Transactions Per Second (TPS) and Time Taken (seconds) for device registration and data storage. Transaction duration was measured in seconds, while transaction costs were computed in Ether. The system conducted the performance tests with Ethereum Network,



Test Operation - Write/Read for 10k Transactions

VII. CONCLUSION

One of the primary concerns among IoT adopters is the security and privacy of data. Ethereum Blockchain Network is a feasible solution for IoT data management and added encryption standards, safeguarding IoT data from unauthorized access. By replacing the central authority, it provides an unchangeable ledger to ensure the reliability of transactions between linked nodes on a distributed blockchain network. All transactions on the Ethereum network are public, allowing anybody to see the transaction details. For mining the vast volume of data in this network, additional processing time and power consumption are required. To overcome these problems in IoT with Ethereum Network, Quorum Blockchain Network (QBN)

REFERENCE

1. *Verma, G.; Prakash, S. Emerging Security Threats, Countermeasures, Issues, and Future Aspects on the Internet of Things (IoT): A Systematic Literature Review. In Advances in Interdisciplinary Engineering; Kumar, N., Tibor, S., Sindhwani, R., Lee, J., Srivastava, P., Eds.; Lecture Notes in Mechanical Engineering; Springer: Singapore, 2021; pp. 59–66.*
2. *Patnaik, R.; Padhy, N.; Raju, K.S. A Systematic Survey on IoT Security Issues, Vulnerability and Open Challenges. In Intelligent System Design; Satapathy, S.C., Bhateja, V., Janakiramaiah, B., Chen, Y.-W., Eds.; Advances in Intelligent Systems and Computing; Springer: Singapore, 2021; pp. 723–730.*
3. *Roy, R.; Dheeba, J. Survey on Methodological Model of IoT in Digital Forensic. In Proceedings of the 2023 International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC), Silchar, India, 3–4 February 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.*
4. *Suny, M.F.I.; Fahim, M.M.R.; Rahman, M.; Newaz, N.T.; Akhund, T.M.N.U. IoT Past, Present, and Future a Literary Survey. In Proceedings of the Information and Communication Technology for Competitive Strategies (ICTCS 2020); Jaipur, India, 11–12 December 2020, Kaiser, M.S., Xie, J., Rathore, V.S., Eds.; Lecture Notes in Networks and Systems; Springer Nature: Singapore, 2021; pp. 393–402.*