

An Effective Framework of Private Ethereum Blockchain Network for Smart Grid

Do Hai Son*
dohaison1998@vnu.edu.vn

Tran Thi Thuy Quynh*
quynhhtt@vnu.edu.vn

Tran Viet Khoa*[†]
khoatv.uet@vnu.edu.vn

Dinh Thai Hoang[†]
hoang.dinh@uts.edu.au

Nguyen Linh Trung*
linhtrung@vnu.edu.vn

Nguyen Viet Ha*
hanv@vnu.edu.vn

Dusit Niyato[‡]
dniyato@ntu.edu.sg

Diep N. Nguyen[†]
diep.nguyen@uts.edu.au

Eryk Dutkiewicz[†]
eryk.dutkiewicz@uts.edu.au

* AVITECH, VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

[†] School of Electrical and Data Engineering, University of Technology Sydney, Australia

[‡] Nanyang Technological University, Nanyang, Singapore

Abstract—A smart grid is an important application in Industry 4.0 with a lot of new technologies and equipment working together. Hence, sensitive data stored in the smart grid is maliciously vulnerable to modification and theft. This paper proposes a framework to build a smart grid based on a highly effective private Ethereum network. Our framework provides a real smart grid that includes modern hardware and a smart contract to secure data in the blockchain network. To obtain high throughput but a low uncle rate, the difficulty calculation method of the Ethereum consensus mechanism is modified to adapt to the practical smart grid setup. The performance in terms of throughput and latency are evaluated by simulation and verified by the real smart grid setup. The modified private Ethereum-based smart grid has performance significantly better than the public one. Moreover, this framework can be applied to any system used to store data in the Ethereum network.

Index Terms—Smart grid, private Ethereum network, throughput, latency.

I. INTRODUCTION

In Industry 4.0, a huge amount of data is created from smart applications. This leads to a serious security threat to many entities, including users, service providers and network operators [1]. Blockchain technology [2] has recently been introduced as a promising solution. It stores data on a distributed ledger and a consensus mechanism among the nodes to avoid the ledger being manipulated maliciously.

In 2009, Bitcoin was introduced by Satoshi Nakamoto and hailed as a radical development in money, being the first example of a digital asset. Since Bitcoin only intended to serve as a decentralized payment, the one's transaction will show up in a long time. That is why after Bitcoin, blockchain technologies are blooming with a famous project called "Ethereum" – a decentralized platform. This technology not only possesses

advantages of Bitcoin's technologies, i.e., decentralization, transparency, immutability, and security-and-privacy, but also has some great improvements, i.e., smart contract and GHOST (Greedy Heaviest Observed Subtree) protocol [3]. It takes only 15 seconds to confirm a new block, approximate 2.5% of Bitcoin [4]. As a result, Ethereum network is applied in many impactful applications, such as smart agriculture [5], [6], Internet-of-vehicles [7], healthcare [8], [9].

Industry 4.0 also sees the smart grid an attractive application, which stores, manages, and exploits the electric system [10]. A traditional grid is described in [11] with only one centralized server, controlled by the energy company, interacts with customers with symmetric or asymmetric schemes. A serious risk of the system is centralization, which leads to increased latency and loss of data when cyber-attacks occur. The Ethereum network can be applied to overcome the risk [12]–[14]. A typical Ethereum-based smart grid is illustrated in Fig. 1. It is a potential system for the future electric network where users and energy markets are connected.

Several studies related to Ethereum-based smart grid are as follows. Zhuang *et al.* [12] reviewed the blockchain technology and showed the architecture and platform of a blockchain-based smart grid for cyber-security. Huang *et al.* [13] presented smart grid's protocols in theory and implemented a communication system using Sigfox devices, but did not apply the Ethereum network. Gao *et al.* [14] introduced a smart contract for their smart grid, but practical experiments were not investigated. Besides Ethereum, many studies applied the smart grid into other blockchain networks, for examples: [15] used the Hyperledger Fabric 1.0 without optimization; [16] proposed a structure of their SmartChain framework, which used the concept of Proof-of-Time (PoT), instead of Proof-of-Work (PoW), for good computational and propagation time versus conventional blockchain network in simulation, but without verification; and [17] proposed an energy trading (ET)

This work is the output of the ASEAN IVO http://www.nict.go.jp/en/asean_ivo/index.html project "Cyber-Attack Detection and Information Security for Industry 4.0" and financially supported by NICT <http://www.nict.go.jp/en/index.html>.

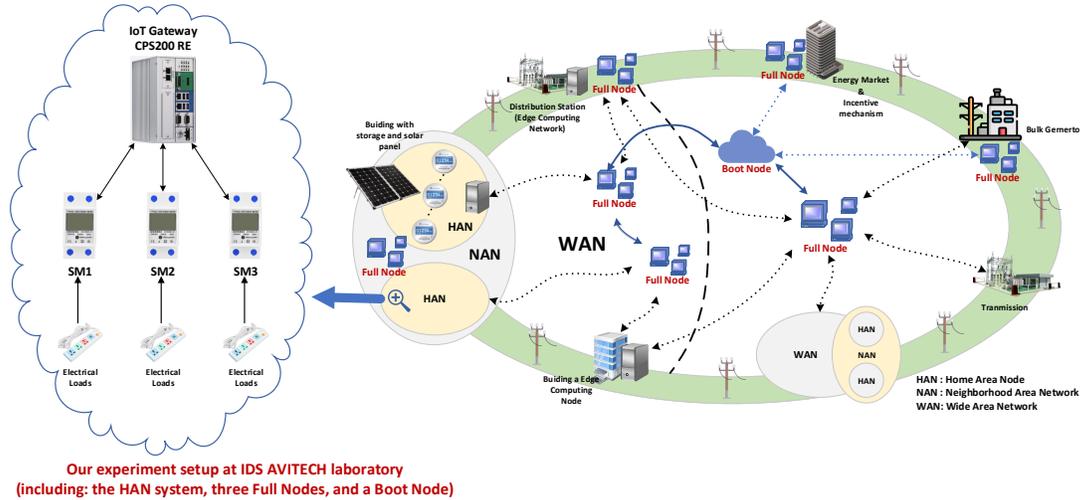


Fig. 1. Model of a smart grid in a smart city.

framework but did not target a specific blockchain network, also the performance was evaluated only in terms of the number of Hash functions to execute; thus, it is unclear if this framework can be implemented in a real blockchain network. Generally, most of the studies reviewed above proposed system models or algorithms of the smart contracts for the smart grid or implementation of a testing system without the Ethereum network. This motivates us to focus, in this work, on developing a novel framework to implement a smart grid with secure data, and improve the efficiency of the private Ethereum network.

As shown in Fig. 1, a smart grid can be separated into three layers: Home Area Node (HAN) which is the home electrical system, Neighborhood Area Network (NAN) which includes several HANs, Wide Area Network (WAN) which is a network of NANs. In this paper, a prototype of an Ethereum-based smart grid is implemented at the HAN layer. This prototype includes the essential components of a smart grid, e.g., smart meters, an IoT Gateway. To secure data inside the blockchain network, encryption methods are required. However, asymmetry schemes would make user reveal the secret key to the nodes [18]. So, a symmetric pre-encryption technique and a simple smart contract are considered to prevent the encrypted data from being duplicated and to avoid revealing the secret key.

The system is based on a private Ethereum network instead of a public Ethereum network. In a small-scale network, the mining time can be reduced while still ensuring security in the Ethereum network. By modifying the difficulty calculation method of the Ethereum consensus mechanism [19], the performance can be improved, with higher throughput, smaller latency, but while keeping an uncle rate as same as that in the main Ethereum network. To experiment this method, we use BlockSim – a recently proposed framework for blockchain systems by Alharby and van Moorsel [20].

The input parameters for performance study by simulation are measured from the real system. Once we have obtained a suitable threshold of block interval in the consensus layer, the trade-off between latency and uncle rate, these parameters are then applied to the prototype to verify the system performance.

The remainder of the paper is organized as follows. In Section III, the Ethereum blockchain technology is briefly described. Section III presents the implementation of the proposed private Ethereum blockchain network for smart grid and the proposed modification on the method of difficulty calculation for improving the performance. Section IV performances improvement experiment on a real smart grid. Finally, Section V concludes the paper.

II. OVERVIEW OF ETHEREUM BLOCKCHAIN TECHNOLOGY

A. Ethereum Blockchain Technology

The Ethereum blockchain network was introduced by Vitalik Buterin in 2015. Typically, it can be divided into seven protocol layers: Storage, Data, Network, Protocol, Consensus, Contract, and Application. The Contract Layer and the GHOST protocol [3] in the Consensus Layer are greatly upgraded, in comparison with those in the Bitcoin network.

The smart contract in the Contract Layer is the first highlight of Ethereum. A smart contract, being simply a piece of codes running on Ethereum, can be built with the Solidity language. The Solidity Compiler compiles the smart contract into Bytecode and Application Binary Interface (ABI). Both of them are packaged into a transaction and deployed into the Ethereum network. Bytecode is an executable code on Ethereum Virtual Machine (EVM) and Contract ABI is an interface to interact with EVM Bytecode.

GHOST is a PoW blockchain protocol like in Bitcoin, except the way it resolves the correct blockchain. Instead of using the longest chain consensus rule in Bitcoin, GHOST follows the path of the sub-tree with the combined hardest proof

of work/difficulty. The sub-tree is created because Ethereum allows us to reintroduce orphaned blocks to the chain as “uncles”. These uncles in the chain allow the network to reduce the mining time while avoiding multiple forking of the ledger (51% attacks). But the uncle has no role in data storage, so if the uncle rate is too high, it will lead to unnecessary storage effort.

B. Types of Ethereum Nodes

A node is a device/program that communicates with the Ethereum network, also known as a client. In this prototype, there are two node types of the Ethereum network. A *full* node keeps a ledger, receives or broadcasts transactions to other nodes. Any full node can be used to confirm blocks and transactions and get rewards. In this case, it is also called “miner”. A *boot* node keeps Ethereum Node Records (ENR) of many full nodes and is not responsible for keeping the ledger, mining, or broadcasting transactions. Any node that connects with a boot node would discover peers in the network.

III. PROPOSED PRIVATE ETHEREUM NETWORK FOR SMART GRID

A. Private Ethereum Network and Hardware Implementation

At the present, 1 Ether (ETH) is approximately 2000 USD. If the smart grid is deployed in the public Ethereum network, at least transaction fees will be charged. There are two ways to deploy the system and send transaction fees in the Ethereum network; one uses Ethereum test-nets and the other creates an own private network.

In the former, test-nets are for free by given ETH coins in some vaults, and the benefit of this way is that we do not necessarily run our miners. However, the ETH coins obtained from vaults are limited. So for a large-scale system, Ethereum test-nets are not compatible.

In the latter, a private Ethereum network is preferred because it can overcome the disadvantage of the former. To deploy a private network, the Ethereum developer team provides a powerful open-source software named Go-ethereum [22] (Geth). The software includes a number of functions to make private nodes, make boot nodes, create new accounts, run full nodes, and so on. Geth v1.10.4 is used in our setup. The private Ethereum network, as shown in Fig. 2, consists of three full nodes which are personal computers with processor Intel® Core™ i7-4800MQ @2.7 GHz, RAM of 16 GB. The network has been setup by a Cisco switch Catalyst 2950 with 100 Mbps bandwidth.

As shown in Fig. 1, the proposed prototype of the HAN layer includes the essential components, i.e., electrical loads, smart meters, and an IoT gateway CPS 200RE.

XTM35SC is the next generation of electricity meters (smart meters). It measures how much electricity has been used, and displays information on a handy in-home display. Furthermore, data collected from the smart meter can be exploited by other IoT devices which use the Modbus-RTU protocol. In the system, data collected from the smart meter will be exploited,

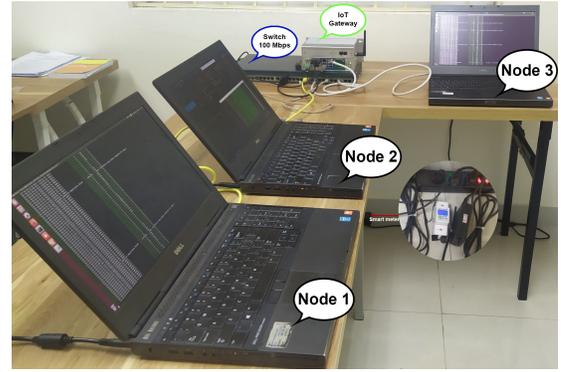


Fig. 2. Our experiment system.

decoded, encrypted and transmitted through an IoT gateway. For simplicity, only consumed energy data will be collected.

CPS 200RE is an edge IoT gateway. It is fully integrated with fieldbus accessibility, Modbus TCP/RTU, PROFINET® or EtherNet/IP™, and so on, for extremely easy deployment of both centralized/decentralized field data implementation in the automation process. The IoT gateway is responsible for the collection of device identification, collection time, and value of consumed energy.

B. Security Enhanced Smart Contract

After collecting raw data from smart meters, the gateway encrypts the data to avoid malicious tapping of the data. This work is necessary because the mechanism of the blockchain makes all data public, requiring pre-encryption before transmission. Both symmetry and asymmetry schemes are considered. But in the blockchain, a classical symmetry scheme named AES-256-CTR [23] is used in the system because asymmetry schemes would make a user reveal his/her $key_{private}$ to Ethereum nodes [18]. In the blockchain network, a pair of key_{pub} and key_{pri} are provided when a user creates a new account. In this paper, the $key_{private}$ used for AES-256-CTR is the same as the key_{pri} . At this stage, privacy of the raw data is guaranteed.

The smart contract is given by Algorithm 1 in which the inputs are encrypted identification/collection time/value of consumed energy $\mathcal{F}(\dots, key_{pri})$ and the outputs are stored inputs, saving the key_{pub} of the user account that has deployed the smart contract as the primary account. This account has permission to add/remove other accounts from the account list, allowing the added/removed account to push data in smart contract or not. Then, the contract is deployed in Remix - Ethereum IDE, and transaction cost per function is captured and shown in Table I. The data flow of every smart meter to the private Ethereum network is summarized in Fig 3.

C. Performance Improvement via Throughput and Latency

The latency of the original Ethereum network is more than 12 seconds because of global scalability [4]. This is not suitable for low-latency applications. When applied to a smart

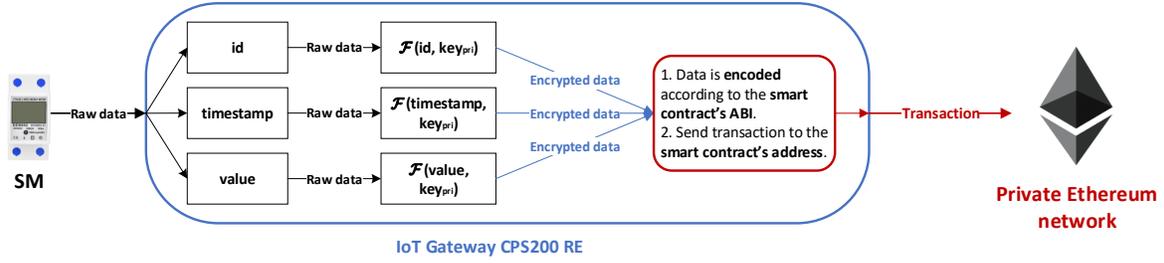


Fig. 3. Data flow from a smart meter to the private Ethereum network.

Algorithm 1

Input: msg.sender (address sending transaction)

$_id \leftarrow \mathcal{F}(_id, key_{pri})$
 $_time \leftarrow \mathcal{F}(_time, key_{pri})$
 $_value \leftarrow \mathcal{F}(_value, key_{pri})$

Output: push data to the Ethereum network

```

1: Class SmartFac {
2:   init_addr  $\leftarrow$  None
3:   total_of_reco  $\leftarrow$  0
4:   struct Reco { id, time, value }
5:   reco[uint][Reco]
6:   trusted_acc[address][bool]
7:   Function constructor( ) {
8:     trusted_acc[msg.sender]  $\leftarrow$  true
9:     init_addr  $\leftarrow$  msg.sender
10:  }
11:  Function add_acc( _addr ) {
12:    if msg.sender  $\neq$  init_addr
13:      return Error
14:    trusted_acc[_addr]  $\leftarrow$  true
15:  }
16:  Function rm_acc( _addr ) {
17:    if msg.sender  $\neq$  init_addr  $\parallel$  _addr == init_addr
18:      return Error
19:    delete trusted_acc[_addr]
20:  }
21:  Event added_reco(addr, _id, _time, _value)
22:  Function new_reco( _id, _time, _value ) {
23:    if trusted_acc[msg.sender]  $\neq$  true
24:      return Error
25:    total_of_reco  $\leftarrow$  total_of_reco + 1
26:    reco[total_of_reco]  $\leftarrow$  Reco {_id,_time,_value}
27:    emit added_reco(msg.sender, _id, _time, _value)
28:  }
29: }
```

grid, with smaller scalability, the system can be improved to obtain higher throughput and smaller block intervals (T).

Based on the analysis of 10,000 consecutive blocks in the Bitcoin network, on average, the transmission time of a block which has just been produced from a miner to 50% and 95%

TABLE I
TRANSACTION COST PER FUNCTION IN THE SMART CONTRACT.

Function	Transaction cost (gas)
Deploy smart contract	573 581
add_acc	26 960
rm_acc	29 197
new_reco	94 408

of all nodes are 6.5 seconds and 40 seconds respectively, and the mean delay is around 12.6 seconds [24]. For a private Ethereum network with small scalability, the mining time can be reduced subject to block propagation, which is measured in this private Ethereum network.

Firstly, we consider the case when the public Ethereum network keeps the stable block interval described in the Ethereum yellow paper [19]:

$$D_i = \begin{cases} D_0 = 131072, & \text{if } i = 0, \\ \max(D_0, P_D + x \times \zeta + \epsilon), & \text{otherwise,} \end{cases} \quad (1)$$

with

$$x = \left\lfloor \frac{P_D}{2048} \right\rfloor, \quad (2)$$

$$\zeta = \max \left\{ y - \left\lfloor \frac{T}{9} \right\rfloor, -99 \right\}, \quad (3)$$

$$y = \begin{cases} 1, & \text{if } \|P_U\| = 0, \\ 2, & \text{otherwise,} \end{cases} \quad (4)$$

$$\epsilon = \left\lfloor 2^{\lfloor \max(i-5000000, 0) \div 100000 \rfloor - 2} \right\rfloor, \quad (5)$$

where D is “difficulty” which is a scalar value corresponding to the difficulty level of this block, P is the parent of this block, P_D , P_s and P_U are “difficulty”, “time_stamp”, and “the number of uncles” of P , respectively, $\lfloor \cdot \rfloor$ denotes the integer division operator, the index i indicates the current block number, T is the block interval, given by $T = \text{current_block_time_stamp} - P_s$.

We focus on the case in which $\|P_U\| = 0$. By this constraint, we consider the effect of the propagation time to T . Simultaneously, when the number of uncles is reduced, the size of the ledger is decreased while still keeping all transactions. Moreover, reducing the number of uncles in the ledger also

TABLE II
THE PARAMETERS IN THE NETWORK ARE PRE-SETUP BASED ON THE MEASURED AND OUR EXPERIMENTS.

Parameters	Values	Notes
Block gas limit	15,000,000 gas	Same as main Ethereum network at June-6-2021
Average transaction size	0.759808 kB	Based on our smart contract
Average block size	60 kB	Same as main Ethereum network at June-6-2021
Average block propagation delay	0.25 seconds	Based on our experiment measured
Sync mode	light	Just sync block header
Number of transactions created per second	100 transactions per second	
Number of node	3 miners	Each miner has 33.33% of the total computing power

avoids “selfish mining” in the network when some miners are simply to mine uncles instead of blocks extending the best chain [25].

We can observe that the Ethereum consensus does not vary T directly, but indirectly through D and a threshold λ in replacement of value 9 in (3). In detail, Eq. (1) depends on Eq. (3), so unless the current block is the first one (genesis block), if $T < \lambda$ then D is adjusted upwards by $(x + \epsilon)$, if $\lambda \leq T < 2\lambda$ then D is unchanged, and if $T \geq 2\lambda$ then D is adjusted downwards proportional to the timestamp difference by from $(-x + \epsilon)$ to $(-99 \times x + \epsilon)$. It can be seen that T is always desired between λ and 2λ seconds, subject to λ in (3). It should be well noted that this threshold is rooted from [24].

However, in a private Ethereum network of limited size, T can be reduced while ensuring that the block propagates through 95% of the nodes. This leads to our modification in the consensus mechanism to improve the block interval. We propose to modify the source code of Geth [22] by restoring the use of the threshold λ in [24] as in

$$\zeta = \max \left\{ y - \left\lfloor \frac{T}{\lambda} \right\rfloor, -99 \right\} \quad (6)$$

instead of fixing it to 9 as in (3), and obtain the value of λ based on the practical smart grid setup.

IV. EXPERIMENTS

A. Performance Study by Simulation

Recently, a simulation framework called BlockSim [20] is introduced. It is used to evaluate a Bitcoin/Ethereum blockchain network depending on input parameters. This paper focuses on evaluating three parameters, i.e., block interval, throughput (transactions per second) and uncle rate. The others based on either the main (public) Ethereum network or measured data from the prototype, and are given in Table II.

The simulation results are shown in Fig. 4. The line with marker of triangle and that of diamond present uncle rate and throughput versus block interval, respectively. The results are averaged over 100 experiments.

Generally, the throughput and the uncle rate both decrease as the block interval time increases. The system has maximum performance with throughput of more than 80 transactions per second (tx/s) and block interval of less than 2 seconds. However, 7-18% of uncle rate is not good because of increasing storage. Comparing with the main Ethereum network of 4.81% on average [26], the proposed network has $T = 3$ seconds.

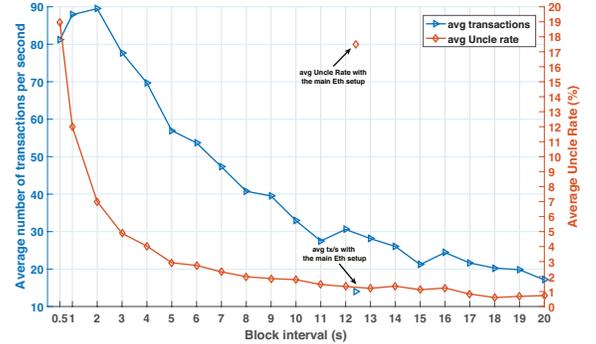


Fig. 4. Performance by simulation: Throughput and uncle rate versus block interval.

At this value of block interval, throughput and uncle rate are 77.72 tx/s and 4.88%, compared with 29.95 tx/s and 1.47% at $T = 12$, respectively.

Besides, other simulation results of a main Ethereum network [24] with 3 full nodes are two scatter points as shown in Fig. 4 with throughput and uncle rate are 14.05 tx/s and 17.48%, respectively. This throughput is suitable with the real main Ethereum network.

B. Verification by Real Data

From the experiment by simulation, to obtain the block interval of approximately 3 seconds and the uncle rate of approximately 4.81%, as shown in Section IV-A, the threshold λ in Eq. (6) is found to be 3.

The experimental results from the real prototype are presented in Fig. 5 with 7,000 consecutive blocks. The top line and bottom line show the average of number of transactions per second in a block time and the block interval, respectively. Here, because the mining process depends on probability, so in this figure, we can see that both are not stable at a specific value. The average value of both on 7,000 blocks, compared to that of the main Ethereum network (found at www.etherscan.io/chart), are shown in Table III. According to the real experiment, the proposed framework can handle 50.08 tx/s and 2.7 seconds between two blocks while those of the main Ethereum network are 16.25 tx/s and 13.48 seconds. Moreover, the obtained uncle rate is 3.03%, is smaller than that of the main Ethereum network.

The throughput of the real experiment is not as good as that of simulated experiment by BlockSim but better than that

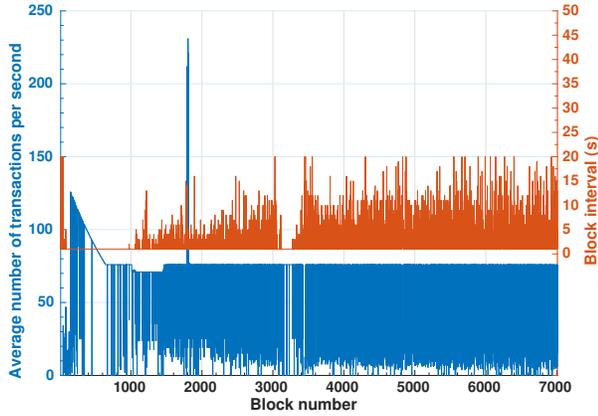


Fig. 5. Real experiment: Throughput and block interval in 7,000 blocks at $\lambda = 3$.

TABLE III
EXPERIMENT RESULTS VERSUS THE MAIN ETHEREUM NETWORK

Parameters	Avg values in the private Eth	Avg values in the main Eth
Transactions per second	50.08 tx/s	16.25 tx/s
Uncle Rate	3.03%	4.81%
Block interval	2.7 seconds	13.48 seconds

of the main Ethereum network. While, the latency is much smaller than that of the main Ethereum network. It is clear that the proposal of changing the consensus mechanism did give better performance.

However, decreasing T may cause risky scenarios, for example, a significantly faster miner joins the network and takes all mining jobs. It can lead to a 51% attack by confirming dishonest transactions. In a private network, there are a few technical methods provided by Geth to solve this problem, e.g., whitelist IP, set maxpeers; more details in [22].

V. CONCLUSION

We have proposed an effective framework to build a private Ethereum network for smart grid with an own private Ethereum network and essential hardware of a smart grid at the home electrical system. The AES-256-CTR standard is applied to pre-encrypt raw data and a smart contract for authentication has been proposed. Then, we have showed how to improve the efficiency of a practical smart grid setup and our verification system can obtain throughput of 50.08 tx/s and latency of 2.7 seconds at an uncle rate of 3.03%. These results clearly show that our proposed framework can outperform the original setup for a private Ethereum network. Moreover, this framework can be applied to any system used to store data in the Ethereum network with any scale.

REFERENCES

[1] V. Alcácer and V. Cruz-Machado, "Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems," *Engineering Science and Technology, an International Journal*, vol. 22, no. 3, pp. 899-919, 2019.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2018. [Online serial]. Available: bitcoin.org/bitcoin.pdf

[3] Sompolinsky and A. Zohar, "Secure High-Rate Transaction Processing in Bitcoin," in *Financial Cryptography and Data Security. FC 2015. Lecture Notes in Computer Science*, Böhme R., Okamoto T. Eds. Springer, Berlin, Heidelberg, 2015, vol. 8975, pp. 507-527.

[4] VitalikButerin, "Toward a 12-second BlockTime," 2014. [Online]. Available: <https://blog.ethereum.org/2014/07/11/toward-a-12-second-block-time/>

[5] G. Mirabelli and V. Solina, "Blockchain and agricultural supply chains traceability: Research trends and future challenges," *Procedia Manufacturing*, vol. 42, pp. 414-421, 2020.

[6] Shyamala Devi, R. Suguna, et al., "Design of IoT Blockchain Based Smart Agriculture for Enlightening Safety and Security," *Comm. in Computer and Information Science*, vol. 985, no. 5, pp. 7-19, 2019.

[7] R. Jabbar, M. Kharbeche, K. Al-Khalifa, et al., "Blockchain for the Internet of Vehicles: A Decentralized IoT Solution for Vehicles Communication Using Ethereum," *Sensors*, vol. 20, no. 14, pp. 1-27, 2020.

[8] Drew Ivan, "Blockchain-based Method for Secure Storage of Patient Records," *ONC Blockchain Ideation Challenge*, no. 8, p. 11, 2016.

[9] W. Hersh, A. Totten, et al., "Health Information Dissemination from Hospital To Community Care: Current State And Next Steps In Ontario," *Journal of Medical Systems*, vol. 63, no. 50, pp. 425-432, 2016.

[10] El-Hawary, Mohamed E., "The smart grid - State-of-the-art and future trends," *Electric Power Components and Systems*, vol. 42, no. 3-4, pp. 239-250, 2014.

[11] Guan, Z., Zhang, Y., Zhu, L. et al, "EFFECT: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid," *Sci. China Inf. Sci.*, vol. 62, p. 32103, 2019.

[12] P. Zhuang, T. Zamir, and H. Liang, "Blockchain for Cyber Security in Smart Grid: A Comprehensive Survey," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 3-19, 2021.

[13] Huang, K. Suankaewmanee, J. Kang, D. Niyato, et al., "Development of reliable wireless communication system for secure blockchain-based energy trading," *16th International Joint Conference on Computer Science and Software Engineering (IJCSE)*, Thailand, 2019, pp. 126-130.

[14] J. Gao, K. O. Asamoah, E. B. Sifah, A. Smahi, et al., G. Dong, "GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid," *IEEE Access*, vol. 6, pp. 9917-9925, 2018.

[15] K. Gai, Y. Wu, L. Zhu, M. Qiu, M. Shen, "Privacy-Preserving Energy Trading Using Consortium Blockchain in Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3548-3558, 2019.

[16] G. Bansal, A. Dua, G. S. Aujla, M. Singh and N. Kumar, "SmartChain: A Smart and Scalable Blockchain Consortium for Smart Grid Systems," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, China, 2019, pp. 1-6.

[17] A. Kumari, R. Gupta, S. Tanwar, S. Tyagi, N. Kumar, "When Blockchain Meets Smart Grid: Secure Energy Trading in Demand Response Management," *IEEE Network*, vol. 34, no. 5, pp. 299-305, 2020.

[18] M. Raikwar, D. Gligoroski, and K. Kravlevska, "SoK of Used Cryptography in Blockchain," *IEEE Access*, vol. 7, pp. 148 550-148 575, 2019.

[19] G. Wood, "Ethereum: A secure decentralised generalised transaction-ledger, petersburg version 41c1837," 2021. [Online serial]. Available: <https://ethereum.github.io/yellowpaper/paper.pdf>

[20] M. Alharby, A. van Moorsel, "BlockSim: An Extensible Simulation Tool for Blockchain Systems," *Frontiers in Blockchain*, vol. 3, p. 28, 2020.

[21] Vitalik Buterin, "Ethereum Whitepaper." [Online]. Available: <https://ethereum.org/en/whitepaper>

[22] Ethereum, "Official Go implementation of the Ethereum protocol." [Online]. Available: <https://github.com/ethereum/go-ethereum>

[23] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Methods and Techniques," National Institute of Standards and Technology, Gaithersburg, MD, Tech.Rep. 20899-8930, 2001.

[24] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*, Italy, 2013, pp. 1-10.

[25] H. Kang, X. Chang, R. Yang, J. Mišić and V. B. Mišić, "Understanding Selfish Mining in Imperfect Bitcoin and Ethereum Networks with Extended Forks," *IEEE Transactions on Network and Service Management* (Early Access), 2021.

[26] Ethereum Statistics, "Ethereum Uncle Rate", [Online]. Available: https://ycharts.com/indicators/ethereum_uncle_rate