CartelChain: A Secure Communication Mechanism for Heterogeneous Blockchains

Riya Tapwal, Sudip Misra, *Senior Member, IEEE*, and
Surjya Kanta Pal
Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur
Kharagpur 721302, India
Centre of Excellence in Advanced Manufacturing Technology, Indian Institute of Technology Kharagpur
Kharagpur 721302, India
Email: tapwalriya@kgpian.iitkgp.ac.in, sudipm@iitkgp.ac.in, skpal@mech.iitkgp.ernet.in

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838600

***Abstract*—In this work, we propose – "CartelChain" – for the secure communication of blockchains (BCs) and achieving optimal throughput. With the advancement of BC technology, many industries are adopting it to maintain a secure, immutable, and decentralized system. Industries such as IoT, supply chain, and finance apply BC technology to maintain a decentralized database and automate different activities using smart contracts. However, storing data of different scenarios from the Industrial Internet of Things (IIoT) in separate BCs (multi-chains) leads to isolated data islands. This results in difficulty for these multi-chains to interact with one another efficiently and credibly. For the seamless operation of industries, it is of significant importance to achieve interoperability among different BCs. Toward achieving this, we propose a solution that utilizes smart contracts for enabling data exchange among various BCs. Further, for secure and reliable communication, we use encryption and an access control mechanism that makes the same more credible and reduces the latency compared with multi-chains sharing the data sequentially. Through experimental results, we demonstrate that the proposed method can utilize the resources more efficiently and reduce CPU as well energy usage by 8% and 6%, respectively. Apart from this, the throughput of the proposed method is 900 tps at 200 requests.**

***Index Terms*—BC, Consensus algorithms, Smart Contract, Industrial Internet of Things.**

I. Introduction

The application of blockchain (BC) technology in the different industries has helped to make a distributed, safe and secure environment for all kinds of data, thereby letting everybody participate and share data in a secure environment. However, due to the structural complexity of these industries' computing and storage systems, it is not possible to use a single BC. To achieve this, there is a need for multiple BCs to store the data so that each BC works in its own local environment [1]. All these chains often differ in their architecture and consensus mechanisms which prevent the data sharing as well as exchange and, further, block the potential data over various applications. To prevent this situation and unlock the potential data, it is necessary to exchange data among various BCs. Meanwhile, it is also necessary to protect the data from various attacks and enable user privacy. To ensure this, we require a secure and robust method of communication to share data between these heterogeneous BCs. Further, all nodes in a BC might not have access to the different kinds of data. We require an access control mechanism to ensure that a node can only access certain data if it has access rights.

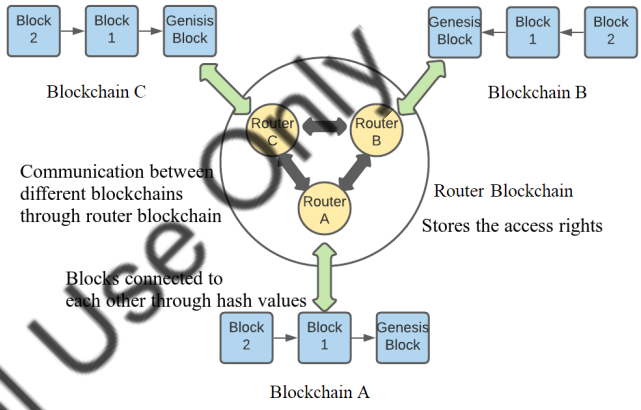


Figure 1: An overview of *CartelChain* architecture.

In this work, we propose a method for communicating between multiple heterogeneous BCs (BCs with different consensus mechanisms) and sharing data between them in a secure manner. To achieve this, we propose a smart contract that is capable of providing access rights to the BC nodes in the system and enabling communication between them. As shown in Fig. 1, any node from a BC that attempts to communicate with other BCs first participates in a network called the routing network. Then the node from sender BC sends a simple request to the smart contract deployed in the routing BC. The smart contract checks the authorization access for the node, and based on that, the routing BC collects the required data from the BC and encodes it in a specified format. Further, it sends the data to the nearest node of target BC based on routing information that is stored in the form of a smart contract in the routing BC. The data packet then gets decoded according to the protocol stored in the target BC and gets added to the BC as a new block.

Example Scenario: Consider a network of IoT devices in the manufacturing industry with various kinds of sensor nodes.

For Personal Use Only

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on December 22,2022 at 12:34:54 UTC from IEEE Xplore. Restrictions apply.

Each such industry consists of various subsections, each with its own network of nodes. Each subsection has its own BC network with different consensus algorithms based on the network architecture. In order to make communication between various BCs, CartelChain provides a secure and robust framework.

A. Motivation

The diverse application scenarios utilize separate BCs for handling their data and result in a multi-chain environment. The multi-chain environment consists of independent BCs, which prevent data sharing as well as exchange and, further, blocks the potential data over various applications. To prevent this situation and unlock the potential data, it is necessary to exchange data among various BCs. Meanwhile, it is also necessary to protect the data from various attacks and enable user privacy. To ensure this, we propose "CartelChain", a mechanism using smart contract and Advanced Encryption Standard (AES) algorithm to provide secure communication [2].

B. Contribution

We enable secure and access-controlled communication between various heterogeneous BCs. To achieve this, the major contributions of this paper are as follows:

- ***CartelChain:*** We propose a smart contracts-based mechanism for handling the sharing of data across heterogenous BCs, each with its own consensus mechanism. This helps in the prevention of data islands and unlocks data over various applications.
- ***Secure Communication:*** We use the AES encryption algorithm for establishing a secure connection between various BCs and preventing the data from various attacks.
- ***Algorithmic Independence:*** CartelChain is independent of the consensus mechanisms as well the cryptographic algorithms to provide security. As a proof of concept, we use the AES encryption algorithm to ensure secure communication.
- ***Robustness:*** CartelChain is a robust mechanism for data sharing among various heterogeneous BC communication as it can handle the communication between various BCs irrespective of their consensus algorithm.
- ***Evaluation:*** We discuss our observations by performing an extensive experiment by deploying CartelChain on the network of Raspberry Pis. We also demonstrate the advantages of utilizing CartelChain over conventional BC.

It may be noted that, in this paper, we propose a smart contract-based solution to solve the access control problem and use symmetric cryptography for providing a secure communication protocol. We do not have a comparative study with other security mechanisms suitable for CartelChain. In the future, we plan to do a comparative study of various security mechanisms suitable for CartelChain rather than AES.

The organization of the rest of the paper is as follows. Section II contains the various researches related to BC, smart contracts, and multi-chains. Section III describes the system model, which consists of network architecture, preliminaries, and proposed solutions. Section IV demonstrate the experimental setup followed by conclusion in Section V.

II. Related Work

A. BC Architecture

Nakomoto *et al.* [3] first introduced the BC architecture through the bitcoin BC in this whitepaper. He proposed a peer-to-peer decentralized electronic cash system that is fault-tolerant and secure. Leible *et al.* [4] described the applications of BC technology in various domains. They explained how BC could be used for the benefits of issues like trustability, collaboration, organization, identification, credibility, and transparency. Crosby *et al.* [5] described the architecture of BC and created a detailed analysis of how BC can be applied in both financial and non-financial. Syed *et al.* [6] presented a detailed analysis of the core BC architecture and its applications in the areas of the Internet of Things (IoT), healthcare, business, and vehicular industry. Frauenthaler *et al.* [7] introduced a new relay scheme that implements a method based on the validation on-demand mechanism combined with different economic incentives, thereby reducing the cost of operation of a relay between different ethereum based BCs by upto 92%. Chen *et al.* [8] and Dai *et al.* [9] proposed that the BC technology can also be utilized by the Industrial Internet of Things for storing, sharing, and processing of data. Further, Mistry *et al.* [10] integrated BC with 5G-enabled Industrial Internet of Things and also discusses various issues that hinder the growth of BC in Industry 4.0. The authors in [11] proposed a customized BC to reduce the storage spaces and computing power. Also, in BC, a heuristic algorithm is proposed to reduce the time to acquire the hash values.

B. Smart Contracts

Negara *et al.* [12] described the applications of smart contracts in various domains. They provided a review of the technological developments that had been done previously and the implementation of smart contracts in multiple domains. Further, they created detailed data of the frameworks, methods, and simulations of smart contracts in these domains. Andesite *et al.* [13] proposed a new method to test smart contracts for bugs and other run time errors. Since a smart contract cannot be changed after being deployed to the BC, it is necessary to run proper tests on the contract. The proposed testing mechanism is based on mutation testing applicable for Solidity smart contracts. Zhang *et al.* [14] proposed an approach for increasing the efficiency of ethereum smart contracts and decreasing the complexity of these smart contracts. They used a public dataset of faulty and buggy smart contracts from Github to evaluate the proposed solution. Kemmoe *et al.* [15] described the latest advancements and updates in developing more efficient smart contracts. They also created a survey and divided it into four categories based on their purposes in the following manner - access management, cryptography, social application, and smart contract structure. Kongmanee *et al.*

For Personal Use Only

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on December 22,2022 at 12:34:54 UTC from IEEE Xplore. Restrictions apply.

[16] proposed a technique for building the basic functional model checking that helps to identify all possible executions that may lead to a breach in the security of the BC. The models proposed in this work are platform and language independent. Further, Alahmadi *et al.* [17] used smart contract-based BC for enabling secure and fair supply chain between various merchants and suppliers, which prevents malicious behavior by imposing penalties.

C. Multi Chain Communication

Luo *et al.* [18] proposed a new component-based framework for exchanging information between multiple heterogeneous BCs by creating a dynamic network using the nodes of the BCs as routers to act as the bridge between the BCs. They used the Escrow transfer protocol and the three-phase commit protocol to maintain atomicity and consistency for inter-chain transactions. The nodes are selected based on the availability of resources and the distance to the nearest node; however, there is no consensus for the data to be shared. McQuillian *et al.* [19] proposed a new routing algorithm that has lower reaction time to different changes in the network and also uses fewer resources, and helps solve the problem of long term loops. The proposed ARPANET routing algorithm worked by maintaining a description of the network topology and delay on each connection in each node. They created a tree that defines the delay from the root node to any other node in the network. Wu *et al.* [20] proposed a new novel method for cross-BC communication based on the periodical committee rotation mechanism that helps to communicate between multiple heterogeneous BCs. They also proposed a verification system based on a messaging service that improved the rate of the number of trusted communication visits among multiple heterogeneous chains. Back *et al.* [21] proposed a pegged sidechain method for multi-chain communications that supported the transactions of ledgers and other assets between multiple BCs and allowed the users to access other BC systems using the bitcoin cryptocurrency. Sidechains are secure, as any breakdown in a sidechain would not affect the primary BC. Mohammed *et al.* [22] proposed a secure method of storing IoT data that is immune to DOS attacks based on the ethereum proof of work consensus protocol. Also, they proposed a communication mechanism between multiple heterogeneous Bcs based on a network of BCs that uses secp256k1 encryption.

D. Synthesis

Communication between heterogeneous multi-chains is an open problem, and many solutions have been proposed to solve this problem. However, very few of them have worked on providing secure communication between the BCs. These solutions, as shown in Table I did not consider security, access control as well as computation together. To address these lacunae, we propose a system CartelChain which provides an access control mechanism as well as a secure connection between the BCs with very low computation and high throughput.

Table I: Difference of *CartelChain* with some existing works.

Paper	Multi-chain data communication	Security	Access Control	Less computationally expensive
Luo *et al.* [18]	✓	×	×	×
McQuillian *et al.* [19]	✓	×	×	×
Wu *et al.* [20]	×	×	×	×
Back *et al.* [21]	×	×	×	×
Mohammed *et al.* [22]	×	✓	×	×
CartelChain (proposed)	✓	✓	✓	✓

III. System Model

A. Network Architecture

As shown in Fig. 2, we consider a heterogeneous multi-chain system with a set of BCs, $B = \{b_1, b_2, b_3, \ldots, b_n\}$ with different consensus mechanisms, $C = \{c_1, c_2, c_3, \ldots, c_n\}$ for data communication, routing nodes and routing BC. The node from each of the BCs that has access right to all the data in the BC participates in the routing BC. It is essential for the routing node to have administrative access to all the data in the BC it's a part of. The routing BC is a network with one or more nodes from each of the BC. It contains the routing information regarding all the nodes. On the updation of routing information in the routing table, all the nodes in the network achieve consensus and update the routing information as a block in the routing BC. Whenever a node in BC wants to send some information to another BC, it sends a request to a smart contract which invokes the same in the routing BC, which checks the access authorization and further encodes the data in a specific format. The sender routing node encodes the packet using the AES algorithm and sends it to the corresponding target routing node of the other BC. The sender routing nodes utilize a three-phase commit protocol for the exchange of data. In the first phase, the sender routing node sends the packet to the target routing node, which decodes it without minting to the BC. In the next phase, the target routing node sends a pre-commit request back to the sender routing node, confirming the transaction and consenting in the routing BC. In the third phase, the sender routing node sends a commit signal back and closes the transaction.

B. Preliminaries

Blockchain: A BC is a decentralized network of nodes, each storing the same records or a ledger. It is a fault-tolerant distributed system. Any transaction between two nodes in a BC system first needs to be consented to by each node participating in the network. On achieving a consensus, the transaction record is added in the form of a block in the ledger or database stored in each of the nodes. Every block in a BC contains a hash value formed from the previous node's hash value, which makes the BC an immutable ledger of records. The first public BC network created was the bitcoin BC which uses the POW (Proof of Work) consensus mechanism to achieve consensus.

Smart contracts: Smart contracts are simple computer programs or transaction protocols written in code that is deployed to a specific address in the BC that can be called by any node

For Personal Use Only

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on December 22,2022 at 12:34:54 UTC from IEEE Xplore. Restrictions apply.

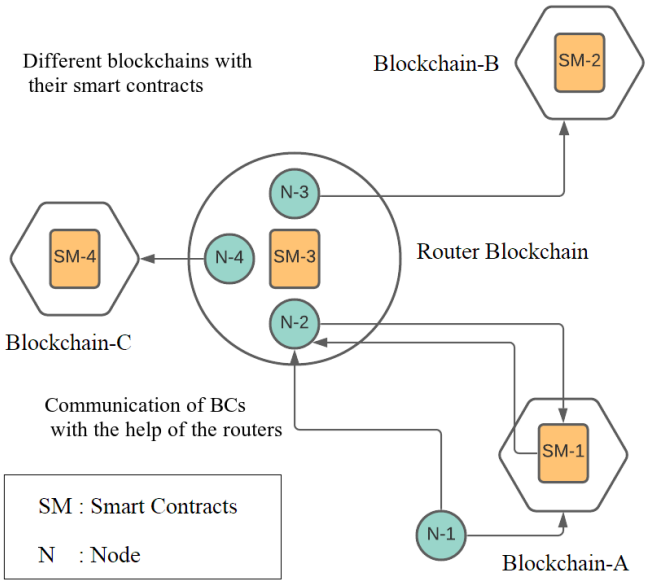


Figure 2: Network architecture of CartelChain.

in the BC. A smart contract, once deployed, cannot be updated or changed. In order to update a smart contract, a new contract has to be deployed, which makes them a secure program that runs on the BC.

Three phase commit protocol: The three-phase commit protocol is a networking protocol used in network communication to make a robust communication between two nodes. Unlike a two-phase commit where the data sent is received and a confirmation signal is sent, in a three-phase protocol, a confirmation is sent by the receiver node, and then in the third phase, a final confirmation is sent again by the sender node to make sure that the data sent was correct and was not transacted due to any error.

C. Proposed Solution

In this work, we solve the problem of access control and secure communication by proposing a mechanism called CartelChain. We propose a method in which a BC communicates with another BC with a completely different architecture and consensus mechanism through the routing BC system using a smart contract deployed in it. As shown in Algorithm 1, any node who wants to send data to another BC first calls the smart contract stored at a specific address in the BC. The smart contract checks if the node has access right to the specific data, and if the node has the same, the smart contract sends a request to the corresponding router node of the BC to call a smart contract stored in the BC. The routing node then transmits the package and uses the three-phase commit mechanism for exchanging data. Further, to make the communication between two BCs secure, we use a cryptographic encryption algorithm AES. So, when a node requests to send data to another BC, the router nodes first encrypt the data packet using the AES algorithm, which is then communicated across the other nodes. This makes the communication tolerant to brute force and other attacks.

For instance, as shown in Fig. 2, there are three BCs connected to the network, namely BC-A, BC-B, and BC-C. Now, suppose BC A wants to communicate with BC B. So, N-1 in BC A sends a request to the N-2 (router node) to access a certain data [23]. N-2 then calls SM-1 with the address of N-1, and if N-1 has the access rights as computed by SM-1, smart contract based on the access rights of the node then permits access and sends a request to the SM-3 in the router BC, which then access the data from the BC and encodes it according to the universal data format between the BCs. Then another layer of encoding goes on the data packet based on the AES algorithm for secure communication by a function in SM-3. The data packet is then sent to the nearest node in BC B according to the routing information. Suppose N-3 is the nearest node in the network. The packet is then received by N-3, and the first later is decoded based on the AES key and the next layer according to the data architecture of the BC.

Algorithm 1: Algorithm for data exchange between heterogeneous BCs.

Input: Data from various BCs
Output: Data exchange between heterogeneous BCs.
Procedure:
for *each* b_i *from various BCs B wants to request data* d_i **do**
  Invoke the smart contract s_i
  s_i invokes the smart contract S present in router BC
  S checks for the access rights
  if b_i *has right for* d_i **then**
    Sender Routing node encodes the data using AES
    Sends the data to the routing nodes using three phase commit mechanism
  end
  // Ensures the atomicity of the data exchange.
  else
    Access denied
  end
end

IV. Performance Evaluation

In this section, we present our experimental setup, deployment architecture, and observations while executing the proposed scheme.

A. Experiment Setup

We use Python 3 to execute CartelChain and utilize three different BCs as a proof of concept. BC-A utilizes Proof of Work, BC-B utilizes Proof of Stake, and BC-C utilizes Practical Byzantine Fault Tolerance. Further, we perform controlled

For Personal Use Only

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on December 22,2022 at 12:34:54 UTC from IEEE Xplore. Restrictions apply.

transmission of the data of different BCs to test the access control mechanism of CartelChain.

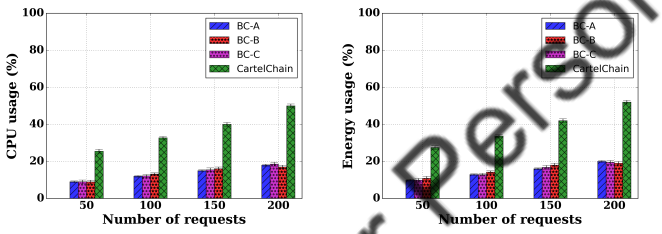
B. Deployment Architecture

We generate a trio BC architecture for the implementation of CartelChain. We have different consensus mechanisms for each BC. The individual BC continuously requests for and sends data to another BC using the router BC, which stores the complete access control information. Further, by using that information, BC sends data to others in a secure fashion.

C. Results

In this section, we discuss the resource utilization, access time, mining time, and the throughput of the CartelChain.

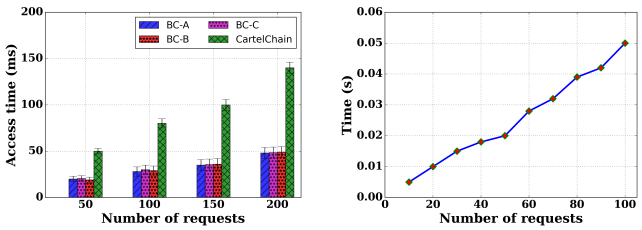
1) Resource Utilization: We observe CPU and energy usage by CartelChain required for accessing data from it. We perform 40 iterations to record the resource utilization and observe that CartelChain utilizes more CPU and energy as compared to the individual BC, as shown in Fig. 3. However, the overall CPU and memory usage by all the individual BCs (the summation of usage by all the three BCs) is 8% and 6% more as compared to CartelChain. This decrease in resource usage is because of the utilization of smart contracts, which manages the access authorization of data and results in the decrease of resource usage as compared to the data access from separate BCs. Further, there is variation in the resource utilization of BC-1, BC-2, and BC-3 because of the varying computational requirements of the consensus utilized in these BCs. We imply that the usage of smart contracts for managing access rights decreases resource utilization.



(a) CPU usage by different BCs for transferring data. (b) Energy usage by different BCs for transferring data.

Figure 3: CPU and energy usage by different BCs for transferring data.

2) Access Time: We observe the time to access data from different BCs with varying numbers of requests. We perform 30 iterations to observe the access time of different BCs at a different number of requests. From Fig. 4(a), we observe that the time to access data from CartelChain is 14ms which is 66% (approx) more from the individual BC. However, there is an overall decrease in access time when we have to access data from the BCs, and it is not clear that the data belongs to which BC. It is because we individually have to check all the BCs for the data. However, in CartelChain, smart contracts manage all this information which prevents us from checking all the BCs individually and reduces the overall access time. Further, from Fig. 4(b), we observe that with the increase in the number of requests, the time to access the data also increases. The access time of CartelChain for 100 requests is 0.06s. We infer that the access time increases linearly with the increase in requests.



(a) Access time of data from various BCs with varying number of requests. (b) Access time of data with varying number of requests.

Figure 4: Access time of data from CartelChain with varying number of requests.

3) Throughput: We observe the throughput of CartelChain with varying numbers of BCs interacting with each other. We perform 40 iterations to observe the throughput with security and without security. From Fig. 5, we observe that there is a 10% decrease in the throughput in CartelChain as compared to the BC system without security. We attribute this decrease to the implementation of AES as it results in a decrease in the number of transactions per second. This is because it also takes some time to implement AES. Further, we also observe that the throughput does not vary much with the increase in the number of BCs. We imply from our observations that the implementation of security mechanisms decreases the throughput. However, the decrease is diminutive in nature and is acceptable.

V. Conclusion

In this work, we proposed a method – "CartelChain" – for the secure communication of BCs and achieving optimal throughput. Many industries adopted BC technology to maintain a

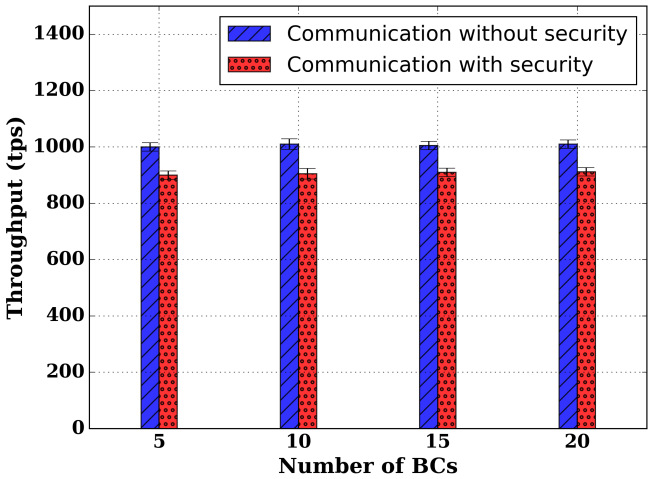


Figure 5: Throughput of CartelChain with varying number of BCs.

For Personal Use Only

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on December 22,2022 at 12:34:54 UTC from IEEE Xplore. Restrictions apply.

secure and immutable decentralized system. However, storing data of different scenarios from the Industrial Internet of Things in separate BCs led to isolated data islands. This results in difficulty for these multichains to interact with each other efficiently and credibly. For the seamless operation of industries, it is of significant importance to achieve interoperability among different BCs. Toward achieving this, we proposed a solution that utilizes smart contracts for enabling data exchange among various BCs. Further, for secure and reliable communication, we used encryption and an access control mechanism that makes the same more credible and reduces the latency compared with multichains sharing the data sequentially. We observed that CartelChain is efficient in reducing the overall resource usage when compared to the summation of resource usage by different BCs. Further, Cartelchain also supports secure communication by reducing the throughput. However, the reduction in the throughput is negligible, which is acceptable.

In this paper, we ceased our research to transmit data from one BC to the other in a secure fashion using AES and did not have a comparative study with other security mechanisms. In the future, we plan to do a comparative study of various security mechanisms suitable for CartelChain rather than AES.

VI. Acknowledgement

The authors are grateful to the Centre of Excellence in Advanced Manufacturing Technology, Indian Institute of Technology, Kharagpur, India for lending their data and industrial platform for implementing the work reported in this work.

References

[1] A. Ahmad, M. Saad, L. Njilla, C. Kamhoua, M. Bassiouni, and A. Mohaisen, "BlockTrail: A Scalable Multichain Solution for Blockchain-Based Audit Trails," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[2] I. Hammad, K. El-Sankary, and E. El-Masry, "High-Speed AES Encryptor With Efficient Merging Techniques," *IEEE Embedded Systems Letters*, vol. 2, no. 3, pp. 67–71, 2010.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[4] S. Leible, S. Schlager, M. Schubotz, and B. Gipp, "A Review On Blockchain Technology and Blockchain Projects Fostering Open Science," *Frontiers in Blockchain*, vol. 2, p. 16, 2019.

[5] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain Technology: Beyond Bitcoin," *Applied Innovation*, vol. 2, no. 6-10, p. 71, 2016.

[6] T. A. Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, and T. Alghamdi, "A Comparative Analysis of Blockchain Architecture and Its Applications: Problems and Recommendations," *IEEE access*, vol. 7, pp. 176 838–176 869, 2019.

[7] P. Frauenthaler, M. Sigwart, C. Spanring, M. Sober, and S. Schulte, "ETH Relay: A Cost-efficient Relay for Ethereum-based Blockchains," in *Proceedings of the IEEE International Conference on Blockchain (Blockchain)*, 2020, pp. 204–213.

[8] W. Chen, M. Ma, Y. Ye, Z. Zheng, and Y. Zhou, "IoT Service Based On Jointcloud Blockchain: The Case Study Of Smart Traveling," in *Proceedings of the IEEE Symposium on Service-Oriented System Engineering (SOSE)*, 2018, pp. 216–221.

[9] H. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[10] I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain For 5G-enabled IoT For Industrial Automation: A Systematic Review, Solutions, And Challenges," *Mechanical Systems and Signal Processing*, vol. 135, p. 106382, 2020.

[11] D. Wu and N. Ansari, "A Cooperative Computing Strategy For Blockchain-Secured Fog Computing," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6603–6609, 2020.

[12] E. S. Negara, A. N. Hidayanto, R. Andryani, and R. Syaputra, "Survey of Smart Contract Framework and Its Application," *Information*, vol. 12, no. 7, p. 257, 2021.

[13] E. Andesta, F. Faghih, and M. Fooladgar, "Testing Smart Contracts Gets Smarter," in *Proceedings of the* 10^{th} *International Conference on Computer and Knowledge Engineering (ICCKE)*, 2020, pp. 405–412.

[14] M. Zhang, P. Zhang, X. Luo, and F. Xiao, "Source code obfuscation for smart contracts," in *Proceedings of the* 27^{th} *Asia-Pacific Software Engineering Conference (APSEC)*, 2020, pp. 513–514.

[15] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son, "Recent Advances in Smart Contracts: A Technical Overview and State of the Art," *IEEE Access*, vol. 8, pp. 117 782–117 801, 2020.

[16] J. Kongmanee, P. Kijsanayothin, and R. Hewett, "Securing Smart Contracts in Blockchain," in *Proceedings of the* $34th$ *IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW)*, 2019, pp. 69–76.

[17] A. Alahmadi and X. Lin, "Towards Secure and Fair IIoT-Enabled Supply Chain Management via Blockchain-Based Smart Contracts," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[18] L. Kan, Y. Wei, A. Hafiz Muhammad, W. Siyuan, L. C. Gao, and H. Kai, "A Multiple Blockchains Architecture on Inter-Blockchain Communication," in *Proceedings of the IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2018, pp. 139–145.

[19] J. McQuillan, I. Richer, and E. Rosen, "The New Routing Algorithm For The ARPANET," *IEEE Transactions on Communications*, vol. 28, no. 5, pp. 711–719, 1980.

[20] Z. Wu, Y. Xiao, E. Zhou, Q. Pei, and Q. Wang, "A Solution to Data Accessibility Across Heterogeneous Blockchains," in *Proceedings of the IEEE* 26^{th} *International Conference on Parallel and Distributed Systems (ICPADS)*, 2020, pp. 414–421.

[21] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling Blockchain Innovations With Pegged Sidechains," *Blockstreams*, vol. 72, 2014.

[22] M. H. Salih Mohammed, "A Hybrid Framework for Securing Data Transmission in Internet of Things (IoTs) Environment using Blockchain Approach," in *IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2021, pp. 1–10.

[23] J. Guo, M. Wang, B. Chen, S. Yu, H. Zhang, and Y. Zhang, "Enabling Blockchain Applications Over Named Data Networking," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

For Personal Use Only

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on December 22,2022 at 12:34:54 UTC from IEEE Xplore. Restrictions apply.