International Journal of Internet, Broadcasting and Communication Vol.16 No.4 116-123 (2024)
http://dx.doi.org/10.7236/IJIBC.2024.16.4.116

IJIBC 24-4-13

Blockchain-Based Pseudonymization Method for Enhanced Data Privacy Management

Youn-A Min

Professor, Applied Software Engineering, Hanyang Cyber University, Korea
yah0612@hycu.ac.kr

Abstract

In this paper, I study the application of blockchain technology in environments that require accurate handling of large-scale data, such as artificial intelligence, to enhance prediction accuracy and data performance. To address data privacy concerns and to strengthen trust in Data Privacy and security, I have researched the application-based performance of zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) using formulated approaches. For performance evaluation, I designed and developed a smart contract based on the proposed content to ensure the implementation of zk-SNARKs. The results indicate that when compared to traditional pseudonymization algorithms like Pseudonymization and tokenization, zk-SNARKs improve confidentiality by 5-10%, data privacy by over 10%, and security by more than 20%.

Keywords: *zk-SNARKs, Pseudonymization, Data privacy, Blockchain*

1. Introduction

Blockchain technology is widely adopted by various platforms for data management due to its advantages, including transparent data management and the preservation of data accuracy [1]. Recently, interest in applying blockchain technology has increased as a way to enhance data accuracy and prediction rates when integrating deep learning technologies such as generative AI [1-2]. However, while blockchain technology ensures data immutability and accuracy, it may pose challenges in terms of data privacy, as data recorded on the blockchain cannot be modified. Table 1 summarizes the advantages and disadvantages of blockchain. As shown, blockchain offers benefits such as decentralization, transparency, prevention of double transactions, and ensuring data immutability. However, its disadvantages include the irreversible nature of the data structure due to the immutability of stored block contents [1-3].

Table 1. Blockchain Advantages and disadvantage

Factor	The details
Advantages	● Securing transparency through decentralization ● False representation and double dealing are prohibited. ● Guarantee data immutability

Manuscript Received: September. 14, 2024 / Revised: September. 19, 2024 / Accepted: September. 25, 2024
Corresponding Author: yah0612@hycu.ac.kr
Tel: +82-2-2290-0872, Fax: +82-2-2290-0872
Professor, Applied Software Engineering, Hanyang Cyber University, Korea

disadvantage	● Irreversibility problem due to inability to change data ● Storage capacity and processing speed problems due to continuous increase in storage space

Artificial intelligence technology is increasingly being applied for prediction and data analysis in environments that handle large-scale data. To improve prediction accuracy, it is essential to have an accurate dataset, and blockchain technology can enhance the accuracy, quality, and stability of artificial intelligence services. However, despite its many advantages, blockchain technology has the characteristic of data irreversibility, which can create challenges in maintaining data privacy. Recently, there has been growing interest in the convergence of blockchain and artificial intelligence technologies, leading to the need for research on strengthening data privacy and implementing Pseudonymization to effectively leverage blockchain technology [1,3-5].

In this paper, I apply and process zero-knowledge proofs based on the zk-SNARKs algorithm to address data privacy issues that may arise from the data irreversibility inherent in blockchain technology, especially in environments that handle large-scale data, such as those involving artificial intelligence. I propose a Pseudonymization method and demonstrate its superiority in performance compared to traditional pseudonymization algorithms through a mathematical formula. Additionally, the robustness of the research is evaluated by assessing its stability in scenarios involving variations in the number of nodes.

2. Related work

Pseudonymization methods applicable in the blockchain environment include general pseudonymization methods such as Pseudonymization and tokenization, as well as advanced methods like zk-SNARKs [5-6].

Pseudonymization is a technique used to protect sensitive data by altering it so that it cannot be read or guessed. This method is often employed during data analysis in environments where the use of actual data is either restricted or unnecessary. Pseudonymization can be divided into two main types: static Pseudonymization and dynamic Pseudonymization [6-9].

Static Pseudonymization is a method of protecting sensitive data by permanently altering it to prevent unauthorized access. This technique is commonly used in replicated or test databases to ensure that the original data remains confidential. On the other hand, dynamic Pseudonymization masks data in real-time and is primarily used to protect sensitive information in active databases [10].

Tokenization is another technique that replaces sensitive data with unique tokens, which are then stored securely and analyzed. In this approach, the original data is stored in a secure environment, and only tokens are used within the actual system. When necessary, the token can be matched with the original data and restored. Simple tokenization involves converting data into tokens using string conversion, while advanced tokenization applies cryptographic algorithms to achieve a higher level of security. Tokenization effectively protects sensitive information since the original data is securely stored, and data restoration can be performed flexibly across the system. However, the increased complexity of the system required for tokenization implementation can lead to performance degradation, particularly in large-scale systems where token generation and management are necessary [10-11].

zk-SNARKs is a cryptographic technique frequently used in blockchain technology. It enables the verification of specific information without revealing the information to the verifier, maintaining the

privacy of the prover. The characteristics of zk-SNARKs are summarized in Table 2.

Table 2. zk-SNARKs Features [12]

Features	The details
Succinctness	Validity of multiple transactions can be verified at once
Non-Interactivity	Proof of non-interactivity with a single message
Arguments of Knowledge	Prove that the prover actually has specific knowledge

zk-SNARKs can facilitate transactions between users through Zcash, and they also enhance scalability and data privacy when used with Ethereum [10-12].

3. Blockchain-Based Pseudonymization Method for Enhanced Data Privacy Management

In this paper, Set up an environment where customer information is pseudonymized using zk-SNARKs on the Ethereum platform. To evaluate the performance of this setup, I used indicators such as Data Privacy, data privacy, and Data Privacy. The performance was measured based on previously established formulas.

The performance evaluation environment simulates a product transaction scenario between customers. The number of nodes corresponding to each institution is set to five or more, and each node includes participation from at least 100 customers. The pseudonymization process is applied to personal customer information, such as customer number, name, and transaction details, using zk-SNARKs. The details of the traded products are disclosed to ensure anonymous and transparent transactions. Figure 1 outlines the steps for creating a smart contract that pseudonymizes specific customer information using zk-SNARKs on Ethereum. The process involves installing and setting up ZoKrates to create proof with ZoKrates, followed by creating a smart contract to generate verification and deliver data.

After compiling and generating proof using the ZoKrates code, a smart contract is created as outlined in Table 3. The contents described in Table 3 primarily involve transaction addition, transaction disclosure, and pseudonymization operations. The smart contracts are then deployed, and transactions are added using the proofs generated through ZoKrates. In the process outlined in Table 3, the zk-SNARK proof is verified through the function that adds the transaction, and the pseudonymized transaction is subsequently stored. Afterward, the product information and transaction amount can be disclosed through a function that retrieves and manages the transaction data.

Table 3. Pseudonymization Process

```
// Create contract Test_Verifier
// mapping Varibales and Create event TransactionAdded and addTransaction(
- define data
- Verify the zk-SNARK proof
- Store transaction
// Create getTransaction funtion to process Transaction
```

Table 4 details part of the process and the smart contract (chaincode) used for pseudonymizing customer information.

Table 4. Part of smart contract code

```
#process
Class Definition -> Create InitLedger Method -> Create Transaction Method -> Query Transaction Method


# smart contract
const {Contract} = require('fabric-contract-api');
const crypto = require('crypto');
class TransactionContract extends Contract {
//Initializes the ledger asynchronously using ctx as a parameter}
//Creates a transaction asynchronously, using the contents in parentheses as parameters (ctx, customerId, customerName, product, amount) {...}
Creates a Query asynchronously, using the contents in parentheses as parameters(ctx, anonymizedId) {...}
```

In Table 4, the class inherits a smart contract, instantiates a chaincode, and then generates a pseudonymized ID and name. During this process, a SHA-256 hash is generated. Subsequently, a transaction object is created, the transaction is stored in the ledger with a specific key, and the details of the transaction are returned after processing.

To analyze the performance of the proposed method, general pseudonymization techniques such as Pseudonymization and tokenization were applied to assess Data Privacy (C), Data Privacy (P), and Data Privacy (S) based on existing research formulas. The performance of these general pseudonymization algorithms (G) was compared with that of zk-SNARKs (Z). Additionally, to evaluate the scalability of the pseudonymization method, a customer-related virtual dataset was created based on the proposed approach. An HTTP Request sampler was implemented using JMeter to compare TPS (Transactions Per Second) performance as the number of nodes increased and as the number of customers per node grew. Performance was measured and compared to confirm that the proposed solution operates stably within the network.

To analyze the performance of the proposed method, I conducted a comparison between general pseudonymization techniques, such as Pseudonymization and tokenization, and zk-SNARKs, focusing on key metrics like Data Privacy, data privacy, and Data Privacy. I also evaluated the scalability of the pseudonymization method by creating a virtual dataset and measuring TPS (Transactions Per Second) performance as the number of nodes increased. Each node was configured to manage 100 customers, resulting in a total of 1000 customers across the network. The libraries libsnark and snarkjs were used to compare the performance of Pseudonymization and tokenization, which are common pseudonymization methods, against the proposed zk-SNARKs approach. Additionally, considering that performance may vary with changes in the number of nodes, I also compared the performance metrics as the number of nodes was adjusted.

Table 5 provides details about the environment used for performance analysis. The blockchain network

was configured to connect five nodes, with each node managing data for 100 customers.

Table 5. Environment for performance analysis

● Blockchain network: Consists of 5 nodes (transactions proceed through 5 platforms)
● Number of customers per node: 1000
● zk-SNARKs implementation libraries: libsnark, snarkjs
● Common Pseudonymization methods: Pseudonymization, tokenization
● Hardware specifications: CPU on each node: 8-core 3.0GHz, RAM: 32GB, SSD: 1TB

The number of nodes was set to 1~2, 10, and 20, and the proposed algorithm was compared by applying the general pseudonymization algorithm and ZK-sNARKS to the data privacy items in each node environment, and the performance evaluation results are as shown in Fig. 1.

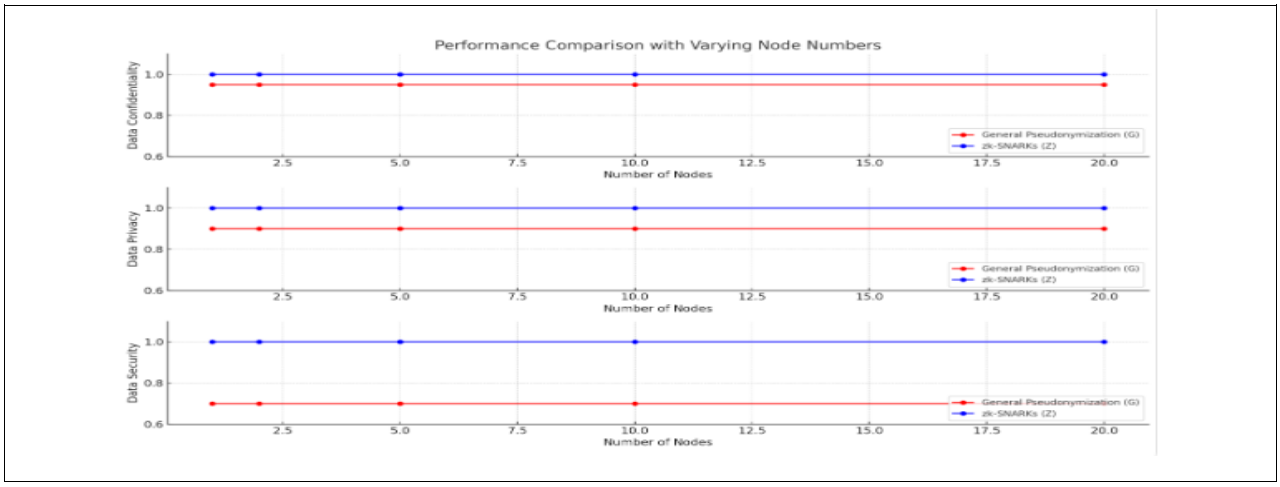

Fig. 1. Performance evaluation results

According to the performance evaluation, Data Privacy in general pseudonymization algorithms ranges between 0.9 and 0.98, while in zk-SNARKs, it ranges between 0.99 and 1, reflecting an expected performance improvement of approximately 10%. Data Privacy shows a distribution between 0.87 and 0.91 for general pseudonymization methods, whereas zk-SNARKs exhibits a distribution between 0.99 and 1, indicating a performance improvement of around 9 to 13%. Data Privacy ranges from 0.7 to 0.75 in general methods, but zk-SNARKs shows a distribution from 0.99 to 1, demonstrating a performance improvement of more than 20%. To verify that the proposed solution operates stably across various node environments, I prepared the same virtual customer dataset used in the previous performance evaluation. An HTTP Request sampler was added through JMeter, based on the Hyperledger Fabric network environment, to simulate an increase in the number of nodes. By measuring and comparing TPS (Transactions Per Second) performance as the number of nodes and customers per node increased, I confirmed that the proposed solution maintains stable operation despite changes in node quantity and data volume.

For the experiment, the number of threads was configured to increase the number of nodes from 10 to

350. The ramp-up period was set to 10 seconds for each thread start, and the number of test repetitions was set to 10. Each thread was configured to use different customer information. The main indicators for performance evaluation included response time, throughput, and error rate. The Hyperledger Fabric network was used for all experiments. The test environment consisted of hardware with 8 CPU cores, 32GB of RAM, and a 1TB SSD, and software versions Hyperledger Fabric v2.2 and JMeter v5.4.1. The network configuration allowed for up to 35 nodes in each organization.

After writing JSON code for each request and response from the listener, customer numbers, names, and addresses were pseudonymized. The code was designed to ensure that information about traded products was transmitted correctly. The summary report of JMeter test results, according to the specified environment, is presented in Table 6.

Table 6. Summary Report via Jmeter

N_node	Average responsetime(ms)	throughput(TPS)	error rate(%)
10	150	200	0.0
50	170	195	0.1
100	200	180	0.5
200	250	160	1.2
350	300	140	2.5

When the number of nodes is relatively small, such as 10 or 50 (denoted as SS), the average response time ranges between 150 and 170 milliseconds, throughput is between 195 and 200 TPS, and the error rate remains below 0.1%. However, as the number of nodes gradually increases to 100, 200, or 350 (denoted as BB), there is a noticeable delay in response time of approximately 170% to 200%, a decrease in throughput by 40% to 60%, and an increase in the error rate by over 200% compared to the smaller node configurations (SS). These performance changes are attributed to the increased number of nodes, which introduces additional complexity into the system. Despite these changes, the system remains in a relatively stable state.

Figure 3 provides a visualization of the test results. TPS (Transactions Per Second) and Response Time Over Time were measured, and it was observed that while performance deteriorates as the number of nodes increases, the system remains relatively stable overall.

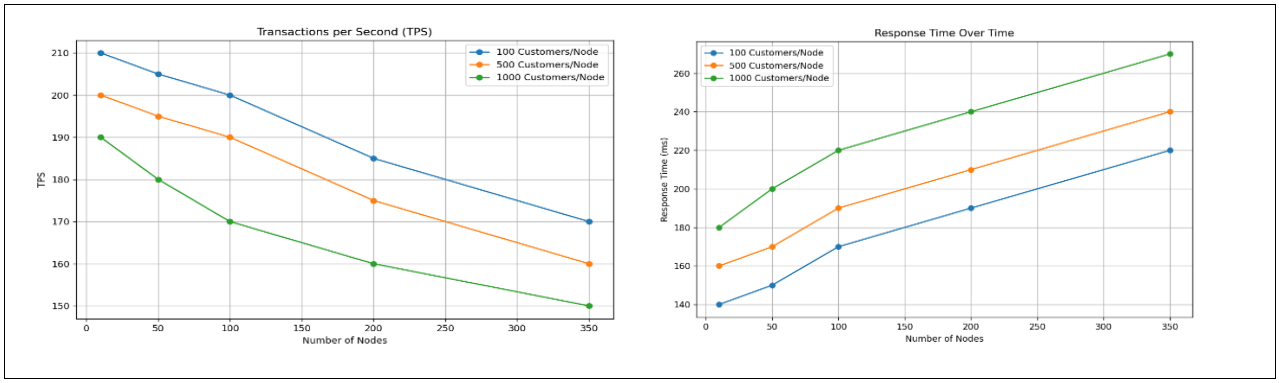

Fig. 2. Visualization by Summary Report

Although performance degrades as the number of nodes in the network increases, the proposed solution demonstrates relatively stable trends in response time, throughput, and error rate during node-to-node communication.

4. Discussion

The research demonstrated that zk-SNARKs offer superior performance in terms of Data Privacy, data privacy, and security compared to general pseudonymization algorithms. Performance improvements were observed across various metrics, with zk-SNARKs consistently outperforming traditional methods, particularly as the number of nodes increased. The analysis evaluated the degree of change in confidentiality, privacy, and security as the number of nodes varied, using predefined formulas. The results showed that, regardless of the number of nodes, Data Privacy (C) in general pseudonymization algorithms ranged from 0.9 to 0.98, while zk-SNARKs exhibited a range between 0.99 and 1, indicating a performance improvement of approximately 10%. Data Privacy (P) in general methods ranged from 0.87 to 0.91, whereas zk-SNARKs achieved a range between 0.99 and 1, confirming a performance improvement of about 9% to 13%. Data Privacy (S) ranged from 0.7 to 0.75 in general methods, while zk-SNARKs showed a range between 0.99 and 1, demonstrating a performance improvement of more than 20%.

Furthermore, performance improvements were observed across various node configurations. With a single node, performance factors improved by an average of 5% to 15%. When the number of nodes increased to two, performance factors improved by 5% to 20%. As the number of nodes increased from 10 to 20, performance factors continued to improve by an average of 5% to 20%. These results confirm that the reliability of data privacy factors is enhanced when using zk-SNARKs, regardless of the number of nodes.

Additionally, JMeter testing was conducted to verify the stable operation of the proposed solution as the number of nodes varied. It was confirmed that the average response time, throughput, and error rate remained relatively stable when the number of nodes increased from 10 to 350.

In this study, the virtual environment used for memory and performance analysis was limited in diversity. Therefore, future research will focus on constructing a virtual environment capable of supporting various memory and dataset configurations to ensure that scalable processing processes are maintained with stability across different platforms and environments.

Reference

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, http://www.bitcoin.org. Retrieved: April 2019.

[2] MuratKuzlu.,etal.,"PerformanceAnalysisofaHyperledgerFabricBlockchainFramework:Throughput,Latencyand Scalability",2019IEEEInternationalConferenceonBlockchain(Blockchain), pp.536-540, 2022, DOI:10.1109/Blockchain.2019.00003

[3] Hyperledger,https://www.hyperledger.org/

[4] S. Rouhani and R. Deters, "Performance analysis of ethereum transactions in private blockchain," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 70-74, 2017, DOI: 10.1109/ICSESS.2017.8342866

[5] Dongkun Hou., et al.,"Privacy-Preserving Energy Trading Using Blockchain and Zero Knowledge Proof", 2022 IEEE International Conference on Blockchain (Blockchain), pp.22-25 , 2022, DOI: 10.1109/Blockchain55522.2022.00064

[6] K. Gai, Y. Wu, L. Zhu, M. Qiu, and 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, DOI: 10.1109/TII.2019.2893433

[7] L. Liu, M. Du, and X. Ma, "Blockchain-based fair and secure electronic double auction protocol," IEEE Intelligent Systems, vol. 35, no. 3, pp.31–40, 2020, DOI: 10.1109/MIS.2020.2977896

[8] S. Zhang, M. Pu, B. Wang, and B. Dong, "A privacy protection scheme of microgrid direct electricity transaction based on consortiumblockchain and continuous double auction," IEEE access, vol. 7, pp.151746–151753, 2019, .DOI: 10.1109/ACCESS.2019.2946794

[9] Hamid Malik., et al.,"Performance Analysis of Blockchain based Smart Grids with Ethereum and Hyperledger Implementations",2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp.1-5, 2019, DOI: 10.1109/ANTS47819.2019.9118072

[10] Jun Wook Heo, et al.," Decentralised Redactable Blockchain: A Privacy-Preserving Approach to Addressing Identity Tracing Challenges", 2024 IEEE International Conference on Blockchain and Cryptocurrency, pp.215-219, May,2024, DOI: 10.1109/ICBC59979.2024.10634438

[11] Minita Samanta, et al., "Application of Ethereum Smart Contract in healthcare and health insurance using Zk-SNARKs in Zcash", 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation ,March,2024, DOI : 10.1109/IATMSI60426.2024.10502670

[12] E. Androulaki., et al.,"Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," , in Proceedings of the Thirteenth EuroSys Conference ACM, pp. 30-35, 2018, DOI : 10.48550/arXiv.1801.10228