SEBDA: A Secure and Efficient Blockchain Based Data Aggregation Scheme

Sehrish Shafeeq and Mathias Fischer
Universität Hamburg, Germany

Keywords: Secure Data Aggregation, Blockchain, End-to-End Integrity.

Abstract: Data aggregation plays a vital role in collecting and summarizing data in the Internet of Things (IoT). Data aggregation results are used to make a critical decision; therefore, the end-to-end integrity of the data aggregation result is of utmost importance. Recently, blockchain-based data aggregation approaches have been proposed that mainly focus on data confidentiality. However, existing approaches ignore two important requirements 1) the end-to-end integrity of data aggregation result and 2) the system's scalability. This paper proposes a blockchain-based data aggregation scheme to detect end-to-end integrity of aggregation results and identify malicious aggregators. Furthermore, we improve the efficiency and scalability of the system by leveraging a sidechain. Our simulation results indicate that the proposed system improves efficiency and scalability compared to conventional blockchain-based data aggregation.

1 INTRODUCTION

The proliferation of the Internet of Things (IoT) is driven by the promise to provide innovative solutions to make life more efficient, integrated, and productive (Catarinucci et al., 2015; Jeong and Lee, 2014). These solutions are equipped with numerous devices that collect and exchange data to make intelligent decisions. By 2023, the number of IoT devices is expected to reach 29.3 billion (Cisco, 2020). In many scenarios, data consumers only require access to *aggregated data*, rather than fine-grained data.

Data aggregation (Liu et al., 2019) is a process of summarizing data obtained from multiple data providers for statistical analysis, typically using a distributed approach called decentralized data aggregation. In this process, data providers send their raw data to *intermediate nodes* that perform the aggregation function and then transmit the aggregation result to the next intermediate node, continuing the process until it reaches the querier. The intermediate nodes are called aggregators.

Decentralized data aggregation reduces the number of data transmissions and improves scalability (Krishnamachari et al., 2002). However, it has some limitations. First, end-to-end integrity cannot be enforced easily, as malicious aggregators may not correctly perform aggregation operations and submit incorrect aggregation results. *End-to-end integrity* implies that the aggregation result is the output of the correct computation of an aggregation function f performed on the data values of a set of known data providers. Second, the process lacks traceability and transparency. The data submitted by data providers and operations performed by aggregators are hidden which makes difficult to find malicious aggregator.

In recent years, blockchain has attracted a lot of attention for enhancing the security of IoT. However, the current literature on the integration of blockchain and data aggregation still faces some challenges. One of the challenges is the scalability of the blockchain network (Xie and Chen, 2021; Xie and Chen, 2021). Furthermore, high transaction fees make microtransactions infeasible and unattractive for data providers to contribute to the process. Another issue is that existing approaches assume the aggregator to be "honest but curious", meaning that it correctly performs computation on the data which is not realistic assumption.

Therefore, there should be a mechanism that takes advantage of blockchain technology in the data aggregation process and simultaneously overcomes the previously mentioned issues. This paper proposes a decentralized data aggregation scheme with a 3-tier architecture. The first tier comprises data providers that send their data to sidechains representing the second tier. A sidechain is a secondary blockchain that helps to improve the system's scalability. Sidechains perform aggregation on the data values and submit ag-

Shafeeq, S. and Fischer, M.
SEBDA: A Secure and Efficient Blockchain Based Data Aggregation Scheme.
DOI: 10.5220/0012077900003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 369-376
ISBN: 978-989-758-666-8; ISSN: 2184-7711
Copyright © 2023 by SCITEPRESS – Science and Technology Publications, Lda. Under CC license (CC BY-NC-ND 4.0)

gregation results to the mainchain. The mainchain is the third tier and performs data aggregation on the intermediate aggregation results submitted by the sidechain. Following are the main contributions of the proposed Secure and Efficient Blockchain based Data Aggregation Scheme (SEBDA) scheme.

- We propose a scheme that leverages the blockchain and smart contracts for decentralized data aggregation. The proposed scheme protects the end-to-end integrity of the aggregation result in the presence of malicious aggregators.
- Our solution improves scalability by coupling on-chain computation with sidechain computation.
- The proposed scheme improves efficiency compared to conventional blockchain-based data aggregation.
- The proposed scheme encourages honest participation in the network by giving economic incentives and discourages malicious behaviors by penalty mechanism.

The rest of the paper is organized as follows: Section 2 summarizes the literature. Section 3 describes the system model and the background. Section 4 presents the proposed solution. Section 5 provides security analysis and comparision with state-of-the-art solutions. In Section 6, we performed performance analysis. Finally, Section 7 concludes the paper.

2 RELATED WORK

Authors of (He et al., 2008) ensure integrity through redundancy by constructing disjoint aggregation trees. Each data provider sends its data to two aggregation trees. The base station accepts aggregation results only if they agree with each other. In (He et al., 2009), authors enhance (He et al., 2008) with a method to identify malicious nodes. In this scheme, data providers split their data into pieces and send them to cluster members. The cluster head performs data aggregation, and cluster members monitor the aggregation result to identify malicious behaviour.

In (Li and Luo, 2012), authors proposed a homomorphic signature scheme to ensure the correctness of the aggregation result. In (Leontiadis et al., 2015), authors uses authentication tag to produce proof of correct aggregation computation. (Cui et al., 2018) proposed a data aggregation scheme that achieves end-to-end integrity using Homomorphic MAC (Agrawal and Boneh, 2009). However, in this scheme, a single private key is shared among all data providers for Homomorphic MAC tag generation, which can be disastrous in the compromise of any node. (Vinodha and Mary Anita, 2021) uses a concatenation based data aggregation function which enables BS to recover individual data values and verifies integrity. However, using concatenation-based data aggregation leads to high communication costs.

(Fan et al., 2020) focus on removing a trusted third party in smart grid data aggregation by using blockchain. Each smart meter encrypts its data using a Paillier cryptosystem (O'Keeffe, 2008) and sends it to the mining node along with Boneh-Lynn-Shacham short signature (Boneh et al., 2001). The mining node aggregates the data, decrypts it and then publishes result on the blockchain. However, the mining node is assumed to be honest but curious. (Loukil et al., 2021) leverage smart contracts to group IoT devices and choose an aggregator. Each data provider first encrypts its data using the consumer public key and then encrypts it using the aggregator public key. However, double encryption increases computational complexity for IoT devices. (Lu et al., 2021) propose a fault-tolerant secure data aggregation scheme. Smart meters submit their data reports to Edge Server (ES), which aggregates local data. ES then sends a transaction that contains the aggregation result to a master node for global data aggregation. (Xie and Chen, 2021) leverages differential privacy and token-based encryption to protect data confidentiality. In this work, smart contracts perform data aggregation and device identification. In summary, existing aggregation schemes have limitations in terms of focusing solely on data confidentiality, relying on a trusted aggregator, and not considering scalability.

3 SYSTEM MODEL AND BACKGROUND

3.1 System Model and Attacker Model

The system model consists of four entities: querier, data providers, primary cluster head, and secondary cluster head. The querier initiates the data aggregation process by issuing a query. The data provider provides its data values $\{v_1, \ldots, v_n\}$ upon request of the querier. Let $\mathbb{D} = \{d_1, \ldots, d_n\}$ be a set of n data providers, where each data provider has a unique identifier. Each primary cluster head collects data values and performs intracluster data aggregation. The data provider can also act as a primary cluster head. Each secondary cluster head is also responsible for performing aggregation function f_{agg} and challenges primary cluster head if it submits incorrect aggregation results.

The querier and data providers are assumed to be honest. An adversary $\mathcal{A}$ can compromise a primary

cluster head or secondary cluster head. A compromised primary cluster head or secondary cluster head is called a malicious aggregator. A malicious aggregator may selectively drop the data received from its cluster members or manipulate the aggregate.

3.2 Background on Blockchain

Blockchain is a distributed ledger technology in which a group of trustless actors works together to maintain a record of transactions (Nakamoto, 2008). The consensus is used to ensure that majority of the network agree on a state in order to achieve security and correctness guarantees. One of the significant features of blockchain is a smart contract., which is a set of promises and protocols agreed among parties for the execution of the promises (Wood et al., 2014). Participants in the network validate the execution outcome of a smart contract. To deploy a smart contract on the blockchain, it is compiled to get Ethereum Virtual Machine (EVM) bytecode, and then a contract creation transaction that contains bytecode is sent to the network. Authorized participants can interact with a contract using the application binary interface (abi). A transaction sender needs to pay gas to invoke contract functions; gas is a measure of computational effort to execute EVM operations.

For a successful IoT application on a blockchain, a blockchain needs to handle a large number of devices. However, the rate at which blockchain handles transactions is limited. Recently, many approaches for solving the blockchain scalability issue have been proposed (Chauhan et al., 2018). One of the notable approaches is a sidechain; a sidechain is a secondary blockchain connected to the mainchain. The protocols of the sidechain are designed to meet application-specific goals. The user can offload some of the tasks to the sidechain to achieve high transaction processing speed and save high transaction fees. The integration of sidechains to the mainchain increases scalability; however, the consensus algorithm is critical to maintaining the security of the sidechain.

4 SEBDA SOLUTION

This section describes the architecture of the SEBDA scheme.

4.1 SEBDA Overview

In this section, we introduce our SEBDA scheme at an abstract level. First, a querier initiates the data aggregation process by publishing a smart contract

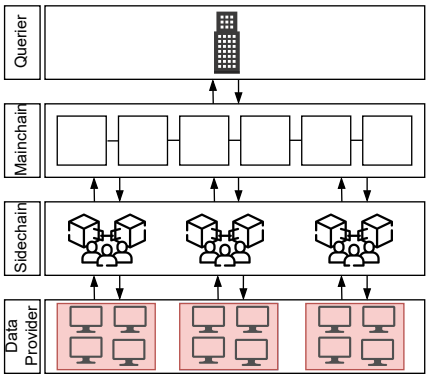

Figure 1: System overview.

`mainchainAgg` on a mainchain. Next, each participant register themselves by sending a transaction to the `mainchainAgg` contract. Then, data providers are divided into multiple clusters, and each cluster is assigned a primary cluster head and secondary cluster heads, respectively. The primary cluster head computes data aggregation and reports the result to the mainchain. This opens a challenge period during which secondary cluster head can challenge the submitted aggregation result. After the challenge period expiration, the `mainchainAgg` contract performs intercluster aggregation. A detailed explanation of the SEBDA is described in Section 4.2.

4.2 SEBDA Protocol Details

The proposed data aggregation protocol consists of the following five phases: system initialization, registration, intracluster data aggregation, dispute resolution, and intercluster data aggregation.

4.2.1 System Initialization

Each participant has a digital wallet, a piece of software enabling interaction with the blockchain. A wallet has a master private key that can essentially generate unlimited public key pk and private key sk pairs. A participant's public key pk is visible to the network. A private key sk is known only to the participant and is used to create a digital signature of a transaction using the Elliptic Curve Digital Signature Algorithm (ECDSA) (Breitner and Heninger, 2019).

4.2.2 Registration

The querier initiates the aggregation process by deploying a smart contract, called `mainchainAgg` contract to the mainchain by sending a transaction. After a successful deployment of the `mainchainAgg` contract, it is assigned a unique address $\mathsf{add}_{\mathsf{main}}$ that

is used for further interactions. In addition to the `mainchainAgg` contract, a querier also publishes a contract called `sidechainAgg` contract on InterPlanetary File System (IPFS). IPFS is a peer-to-peer distributed file-sharing system. Further details of these contracts are mentioned in subsequent sections.

Next, each data provider interested in participating registers itself by sending a transaction that invokes a function `regDp` of the `mainchainAgg` contract. The `mainchainAgg` contract mainly handles participants' registration, clusterization, challenges, and intercluster aggregation functions. As a result of successful registration, each device is assigned a unique identifier d_i. Participants interested in acting as aggregators also register themselves on the mainchain. To do so, each aggregator sends a transaction that invokes a function `regCh` of `mainchainAgg` contract. To thwart submission of incorrect aggregation results, aggregators are required to submit tokens called deposits during registration. The deposit act as leverage against malicious aggregators. Aggregators are also incentivized to participate and follow the rules in the network by rewarding them.

Next, querier partitions data providers $\mathbb{D}$ into mutually disjoint non-empty n sets whose union is $\mathbb{D}$ and each set is called a cluster c_i. Then, each cluster c_i is assigned an aggregator called primary cluster head. Each cluster c_i is also assigned multiple watchdogs, called secondary cluster heads. A secondary cluster head verifies the aggregation result submitted by primary cluster head. Both primary cluster head and secondary cluster head are randomly selected. They can also be chosen by aggregator election algorithm such as LEACH protocol (Heinzelman et al., 2000; Heinzelman et al., 2002); but this is not in the scope of this research.

4.2.3 Intracluster Data Aggregation

In this phase, each cluster c_i creates its sidechain. The sidechain stores data values of data providers of a cluster, and aggregators are responsible for managing it. After the sidechain creation, the primary cluster head fetches the pointer to `sidechainAgg` contract from the mainchain. Then, using the pointer, the primary cluster head gets the `sidechainAgg` contract from IPFS and publishes it on the sidechain. As a result of successful `sidechainAgg` contract publication, it is assigned a unique address $\mathsf{add}_{\mathsf{sidecontract}}$. Next, the cluster head sends the address $\mathsf{add}_{\mathsf{sidecontract}}$ and abi of `sidechainAgg` contract to the data providers of its cluster.

Each data provider d_i in the cluster encapsulates its data value v_i in a transaction and sends it to the sidechain within a specified period. The cluster aggregator verifies the transaction and executes it if it recognizes the data provider as an authenticated cluster member. The state of the intracluster aggregate changes as a result of successful transaction execution. After computing the intracluster aggregation result, the primary cluster head submits it to the mainchain.

When the mainchain receives the submitted intracluster aggregation result, it emits an event to notify secondary cluster heads. This opens a challenge period during which secondary cluster head of a cluster can open a dispute. Each secondary cluster head compares the submitted intracluster aggregation result with the intracluster aggregation result computed on the sidechain; if they are not equal, the secondary cluster head challenge it within the challenge period. If the intracluster aggregation result has not been challenged by any secondary cluster head during the challenge period the mainchain optimally accepts it, rewards the primary cluster head for good behavior, and proceeds to Section 4.2.5.

4.2.4 Dispute Resolution

This phase executes only when one of the secondary cluster heads challenges the intracluster aggregation result submitted by primary cluster head. The `mainchainAgg` contract emits an event about the intracluster aggregation result submitted by challenger secondary cluster head and asks to vote within a specified period t. If the majority of secondary cluster head votes for the challenger aggregation result it is accepted as a valid result; otherwise, the system accepts the intracluster aggregation result submitted by the primary cluster head.

The dispute resolution phase ensures that the incorrect aggregation result submitted by malicious primary cluster head is not accepted by the system as long as the majority of aggregators are honest. Similarly, a malicious secondary cluster head cannot falsely win the challenge. To penalize a malicious primary cluster head, a part of its deposit is burned, meaning tokens are removed from circulation by sending them to the wallet address that cannot be used by anyone.

4.2.5 Intercluster Data Aggregation

Once all clusters have submitted their aggregation result and the challenge period is over, the querier initiates intercluster aggregation by sending a transaction that invokes a function `computeInterclusterAgg` of `mainchainAgg` contract. The intercluster aggregate is computed on the mainchain, and a network of miners ensures transparent and correct computation of the in-

Algorithm 1: Challenge.

Input: cluster id c_i,
cluster head address ch_i,
claimed value $claimedAgg_i$
Modifier: is challenge period active,
is challenger valid;
1: start challenge against cluster head ch_i with claimed value $claimedAgg_i$;
2: increment vote v against cluster head ch_i by 1;
3: **if** total votes v is greater than majority of aggregators **then**
4: $agg_i \leftarrow claimedAgg_i$
5: a part of ch_i deposit is burned
6: **if** $ch_i.deposit <= 0$ **then**
7: remove cluster head ch_i from aggregators list
8: **end if**
9: **end if**

tercluster aggregate. As a result of successful execution, an event is emitted to notify the querier about the aggregation result.

5 SECURITY ANALYSIS

5.1 End-to-end Integrity

To show that SEBDA ensures end-to-end data integrity, we analyze it in 2 scenarios 1. incorrect intracluster data aggregation 2. incorrect intercluster data aggregation.

For the first scenario, assume there is a malicious primary cluster head $\mathcal{A}$ which submits incorrect intracluster aggregation result agg'_i to the mainchain. Once the mainchain receives the result, it notifies secondary cluster heads of the cluster c_i and opens the challenge period. An honest secondary cluster head challenges the malicious primary cluster head $\mathcal{A}_i$ and submits correct aggregation result $claimedValue$, according to Algorithm 1. Other honest secondary cluster heads belonging to cluster c_i votes in favor of the challenger. The intracluster aggregation result agg'_i submitted by $\mathcal{A}$ is replaced by $claimedValue$ when $claimedValue$ receives $v/sch_i > \tau$ votes, where v is a total number of votes in favor of $claimedValue_i$, sch_i is a total number of secondary cluster head in the cluster i and τ is threshold whose value must be greater than 50%. The more the secondary cluster head there are, the more challenging it becomes for a malicious primary cluster head to compromise the aggregation result by colluding with a significant portion of cluster heads. According to the assumption, the majority of secondary cluster head are honest; thus, intracluster aggregation result agg'_i submitted by malicious primary cluster head $\mathcal{A}$ is replaced by correct intracluster aggregation result agreed by the honest majority.

Now, let's consider the second case in which adversary $\mathcal{A}$ tries to manipulate intercluster aggregation result. Our scheme leverage smart contract to compute intercluster aggregation result. The miners of the main network are responsible for verifying transactions and computing the new state of the smart contract. Therefore, an adversary $\mathcal{A}$ can only manipulate intercluster aggregation results if its hash power is greater than half of the total hash power, which is hard to achieve in practice.

5.2 Authentication and Non-repudiation

All participants sign their transactions using ECDSA. Other participants use the public key to verify the transaction's integrity (single-hop data integrity) and authenticity. Single-hop integrity ensures that an external entity does not modify the aggregation result but does not ensure that the aggregation result is computed correctly. ECDSA has been proven existentially unforgeable under the hardness of the Discrete Logarithm Problem (DLP) and collision-resistant hash function. Furthermore, since all participants sign their transaction using their private keys; therefore a participant cannot deny a transaction. Thus, the proposed scheme ensures single hop integrity, authentication and non-repudiation properties.

5.3 False Data Injection

In a false data injection attack, an external adversary $\mathcal{A}$ attempts to inject false data as data values or aggregation results. It is essential to ensure false data injection because the blockchain is a public ledger, and any member of a blockchain can send a transaction to submit data value and aggregation results. In our scheme, each primary cluster head uses ECDSA to signature a transaction that invokes an intracluster aggregation result function `subIntraclusterAgg`. The modifier of a function `subIntraclusterAgg` ensures that only authenticated primary cluster head can submit intracluster aggregation results. Modifiers check the prerequisites of a function and execute the function of a smart contract only if prerequisites are satisfied. Thus the modifier of a `subIntraclusterAgg` function rejects the intracluster aggregation result if an adversary $\mathcal{A}$ tries to submit an intracluster aggregation result. Similarly, the `sidechainAgg` contract ensures that only authenticated data providers of a cluster can submit data values. Also, the `mainchainAgg` contract ensures that only members of a cluster can challenge primary cluster head. Hence, by utilizing ECDSA and modifiers, our scheme filters out false data submitted by an external adversary $\mathcal{A}$.

Table 1: Comparision between SEBDA and related work.

Paper	Single hop data Integrity	False Data Injection	Scalable	End-to-end data integrity	Smart contracts
(Fan et al., 2020)	✓	✗	✗	✗	✗
(Wang et al., 2020)	✓	✗	✗	✗	task publication
(Xie and Chen, 2021)	✓	-	✗	✗	data aggregation
(Loukil et al., 2021)	✓	✓	✗	✗	group formation, aggregator selection
(Lu et al., 2021)	✓	✓	✗	✗	✗
SEBDA	✓	✓	✓	✓	data aggregation , dispute resolution

5.4 Comparision with Related Work

Comparision with other blockchain-based data aggregation schemes (Fan et al., 2020; Wang et al., 2020; Xie and Chen, 2021; Loukil et al., 2021; Lu et al., 2021) shows that they provide single-hop data integrity and do not consider malicious aggregator; thus fails to achieve end-to-end integrity. The main goal of (Xie and Chen, 2021) is to protect the data confidentiality in the presence of an untrusted aggregator and leader. However, a leader responsible for calling the aggregation function of a smart contract can submit an incorrect or incomplete database $\mathcal{DB}$ where $\mathcal{DB}$ is collected data of devices. Additionally, the system is not scalable enough to handle many IoT devices.

6 EVALUATION

In this section, we compare the performance of our solution with a Conventional Blockchain-based Data Aggregation (CBDA) system that performs all operations on the mainchain.

6.1 Experimental Setup and Procedures

We chose Ethereum blockchain to conduct our experiments, and Ganache is used to simulate the Ethereum blockchain. To make Ganache close to the main Ethereum network, we set the block time to 12 sec. It is important to note that other factors, such as gas price and the number of pending transactions, also influence the transaction processing time on Ethereum main network. However, these factors are challenging to incorporate into the experimental setup. The smart contracts are written in Solidity, and the web3.js library is used to interact with the Ethereum node. The data values of data providers are randomly generated, and a sum aggregation query is used for performance evaluation. We considered 3 sidechains connected to the mainchain, and each sidechain is assigned one primary cluster head and two secondary cluster heads. The experiments are repeated 30 times and each data value represents the average. The experiments are conducted on the physical machine equipped with Intel(R) Xeon(R) CPU $E5-2660$ v4, 131.7964 GB RAM, CPU Frequency 2.00GHz, and docker container is used for virtualization.

6.2 Results

We have evaluated the performance of our scheme based on gas, transactional cost, and computational cost. Table 2 shows the gas consumed by different functions of a solidity smart contract. Ethereum sets the minimum gas units for any transaction to $21,000$. This mainly covers the cost of running an elliptic curve algorithm. In addition to this, a transaction also incurs gas that is dependent on the used opcodes in the smart contract. It can be seen that the cluster head registration function `regCh` requires the highest gas. This is because when a node sends a transaction to register as cluster head, the `regCh` function triggers the change of Ethereum accounts state as a result of deposit transfer. Then, it also updates the smart contract's state to reflect the newly added cluster head.

Table 2: Gas unit of different functions.

Number	Function	Gas units
1	regDp	95413
2	regCh	179334
3	subData	27631
4	subIntraclusterAgg	61093
5	compInterclusterAgg	76791
6	challenge	76652

Next, in Figure 2, we compare the cost of the data aggregation process in terms of transaction fees. The data aggregation cost is the sum of each transaction fee executed during the aggregation process, including the registration fee. The results are compared with

the CBDA system. The transaction fee is calculated as $transaction\ fee = gas\ units * gas\ price\ per\ unit$. The gas used by different aggregation functions can be seen in Table 2. Gas price is the amount of ether (ETH) per gas unit that must be paid to execute a network transaction and fluctuates over time. ETH is a digital coin used in the Ethereum network, and one of the denominations of ETH is Gwei. At the time of running experiments (26 Sep 2022), the gas price was 14 Gwei in the Ethereum network; thus, we have also used 14 Gwei as the gas price for our network. In Figure 2, it can be observed that our proposed SEBDA results in overall lower transaction fees as compared to the CBDA system. This is because intracluster aggregation computation is performed on the sidechain in our SEBDA scheme. Hence, it saves transaction fees and encourages participation of data providers in the network.

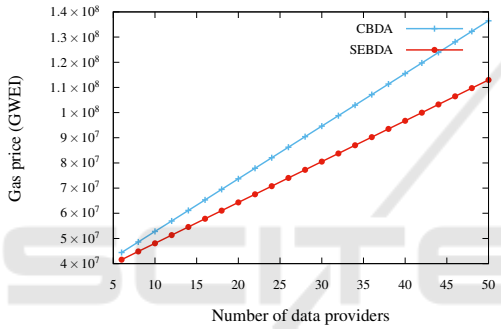

Figure 2: Transaction cost comparision.

Figure 3 displays the computational cost of data submission and intercluster aggregation in our scheme. Our SEBDA scheme takes 139 ms for data submission and intracluster aggregation of 100 data providers. Both operations are handled on the sidechain, improving efficiency and removing transactional costs on the data provider. In Figure 4, we illustrate the scalability of our scheme and analyze the effect of an increasing number of data providers on the total data aggregation computation time. The inside smaller chart we zoomed in on the y-axis of the original graph. In our SEBDA scheme, most of the total time is consumed by intracluster aggregation result submission and intercluster cluster aggregation because these operations are done in the mainchain, which takes 12 sec to process a single transaction as we can see that our SEBDA scheme shows superiority than the CBDA system with an increasing number of data providers. For example, our SEBDA scheme reduces 91.9% of the total computational cost for 50 data providers. Thus, it can be concluded that our scheme can handle a large number of data providers without affecting overall performance, as its computational time increases linearly with the increasing number of data providers.

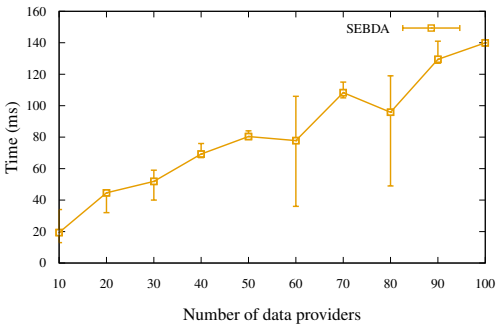

Figure 3: Computational cost of data submission and intracluster aggregation.

7 CONCLUSION AND FUTURE WORK

We propose a 3-tier blockchain-based data aggregation scheme that use sidechains for data submission and intracluster aggregation. To protect the end-to-end integrity, SEBDA utilizes challenge and dispute resolution protocols. The reward and punishment protocols encourage participation and discourage malicious behaviors. Our simulation results prove that SEBDA improves efficiency and scalability. To the best of our knowledge, this is the first blockchain-based data aggregation scheme that addresses integrity and scalability requirements. In the future, we will investigate the effect of an increasing number of sidechains on security and find a sweet spot between security and scalability of the blockchain-based data aggregation scheme. Moreover, we also aim to deploy the proposed solution on low power IoT devices and conduct experiments in a practical setting. This will involve creating a small-scale IoT network where IoT devices submit data values to the sidechain.

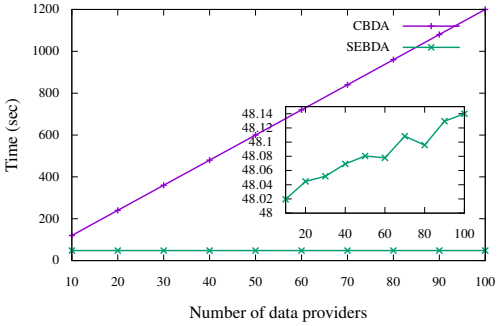

Figure 4: Computation cost of data aggregation.

REFERENCES

Agrawal, S. and Boneh, D. (2009). Homomorphic macs: Mac-based integrity for network coding. In *Applied Cryptography and Network Security: 7th International Conference, ACNS 2009, Paris-Rocquencourt, France, June 2-5, 2009. Proceedings 7*, pages 292–305. Springer.

Boneh, D., Lynn, B., and Shacham, H. (2001). Short signatures from the weil pairing. In *International conference on the theory and application of cryptology and information security*, pages 514–532. Springer.

Breitner, J. and Heninger, N. (2019). Biased nonce sense: Lattice attacks against weak ecdsa signatures in cryptocurrencies. In *International Conference on Financial Cryptography and Data Security*, pages 3–20. Springer.

Catarinucci, L., De Donno, D., Mainetti, L., Palano, L., Patrono, L., Stefanizzi, M. L., and Tarricone, L. (2015). An iot-aware architecture for smart healthcare systems. *IEEE internet of things journal*, 2(6):515–526.

Chauhan, A., Malviya, O. P., Verma, M., and Mor, T. S. (2018). Blockchain and scalability. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 122–128. IEEE.

Cisco, U. (2020). Cisco annual internet report (2018–2023) white paper. *Cisco: San Jose, CA, USA*.

Cui, J., Shao, L., Zhong, H., Xu, Y., and Liu, L. (2018). Data aggregation with end-to-end confidentiality and integrity for large-scale wireless sensor networks. *Peer-to-Peer Networking and Applications*, 11:1022–1037.

Fan, H., Liu, Y., and Zeng, Z. (2020). Decentralized privacy-preserving data aggregation scheme for smart grid based on blockchain. *Sensors*, 20(18):5282.

He, W., Liu, X., Nguyen, H., and Nahrstedt, K. (2009). A cluster-based protocol to enforce integrity and preserve privacy in data aggregation. In *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, pages 14–19. IEEE.

He, W., Nguyen, H., Liuy, X., Nahrstedt, K., and Abdelzaher, T. (2008). ipda: An integrity-protecting private data aggregation scheme for wireless sensor networks. In *MILCOM 2008-2008 IEEE Military Communications Conference*, pages 1–7. IEEE.

Heinzelman, W. B., Chandrakasan, A. P., and Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, 1(4):660–670.

Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pages 10–pp. IEEE.

Jeong, J. and Lee, E. (2014). Vcps: Vehicular cyber-physical systems for smart road services. In *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pages 133–138. IEEE.

Krishnamachari, L., Estrin, D., and Wicker, S. (2002). The impact of data aggregation in wireless sensor networks. In *Proceedings 22nd international conference on distributed computing systems workshops*, pages 575–578. IEEE.

Leontiadis, I., Elkhiyaoui, K., Önen, M., and Molva, R. (2015). Puda–privacy and unforgeability for data aggregation. In *International Conference on Cryptology and Network Security*, pages 3–18. Springer.

Li, F. and Luo, B. (2012). Preserving data integrity for smart grid data aggregation. In *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pages 366–371. IEEE.

Liu, X., Yu, J., Li, F., Lv, W., Wang, Y., and Cheng, X. (2019). Data aggregation in wireless sensor networks: from the perspective of security. *IEEE Internet of Things Journal*, 7(7):6495–6513.

Loukil, F., Ghedira-Guegan, C., Boukadi, K., and Benharkat, A.-N. (2021). Privacy-preserving iot data aggregation based on blockchain and homomorphic encryption. *Sensors*, 21(7):2452.

Lu, W., Ren, Z., Xu, J., and Chen, S. (2021). Edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid. *IEEE Transactions on Network and Service Management*, 18(2):1246–1259.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260.

O'Keeffe, M. (2008). The paillier cryptosystem. *Mathematics Department April*, 18:1–16.

Vinodha, D. and Mary Anita, E. (2021). Discrete integrity assuring slice-based secured data aggregation scheme for wireless sensor network (dia-ssdas). *Wireless Communications and Mobile Computing*, 2021:1–17.

Wang, X., Garg, S., Lin, H., Kaddoum, G., Hu, J., and Hossain, M. S. A Secure Data Aggregation Strategy in Edge Computing and Blockchain empowered Internet of Things. pages 1–1.

Wood, G. et al. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32.

Xie, X. and Chen, Y.-C. Decentralized Data Aggregation: A New Secure Framework based on Lightweight Cryptographic Algorithms. In *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–2.

Xie, X. and Chen, Y.-C. (2021). Decentralized data aggregation: a new secure framework based on lightweight cryptographic algorithms. *Wireless Communications and Mobile Computing*, 2021.