Blockchain-based Privacy Preserving Linear Regression

Received: 30 January 2023; Accepted: 15 February 2023

Research Article

Zeynep Delal Mutlu
Dept. of Computer Engineering,
TOBB University
Ankara, Turkey
zmutlu@etu.edu.tr

Yesem Kurt Peker
TSTS School of Computer Science,
Columbus State University
Columbus, GA, USA
peker_yesem@columbusstate.edu

Ali Aydın Selçuk
Dept. of Computer Engineering,
TOBB University
Ankara, Turkey
aselcuk@etu.edu.tr

***Abstract*—In this study we propose a blockchain-based architecture that uses smart contracts and homomorphic encryption to allow statistical computations on confidential data by third parties. The use of blockchain provides the much-desired security properties of integrity and fault tolerance and homomorphic encryption preserves the privacy of the data. We present the design, implementation, and testing of our system. Our results show that a blockchain-based data sharing mechanism with homomorphic calculations via a smart contract is feasible and provides improvements in protecting the data from unauthorized users. Even though our work focused on linear regression, the architecture can be used for other statistical analysis and machine learning algorithms.**

Keywords—blockchain, homomorphic encryption, statistics, linear regression, ethereum

I. Introduction

Blockchain is a decentralized digital ledger consisting of blocks that are chained together via cryptographic hash functions. First introduced in 2008 as a transaction ledger for Bitcoin [1], blockchain technology has found applications not only in finance but also in healthcare, supply management, the Internet of Things (IoT), and other areas. Tamper resistance and fault tolerance, combined with the ability to execute code on the blockchain (i.e. smart contracts), blockchain technology became a very attractive solution for applications that require high integrity, high availability, and auditability of data and transactions [2].

One security property that is not natively included in blockchain is confidentiality [3]. Data submitted to the blockchain as part of a transaction or a smart contract is open to everyone for viewing. Unless encrypted, the privacy requirements of data shared on the blockchain cannot be satisfied. There are several studies that explore the use of encryption on blockchain. One of the encryption methods proposed for use for the confidentiality of data on blockchain is homomorphic encryption [4]. Homomorphic encryption allows computations to be done on the encrypted data (ciphertext) without having to decrypt it.

This work proposes a prototype for a blockchain-based data-sharing system that uses smart contracts and homomorphic encryption to allow statistical computations on confidential data by third parties. The advantages of the proposed system over systems that do not use homomorphic encryption include protecting the confidentiality of data during processing in addition to transmission and storage.

Our contributions in this work are as follows:

1) Design a prototype system for sharing and analysis of aggregated data respecting the privacy of the data.

2) Implementation and evaluation of the prototype system using the Ethereum platform for calculating linear regression equation for data that consists of pairs of values.

The remainder of this paper is organized as follows. In Section 2, we review the literature on privacy-preserving data sharing and analysis methods on blockchain. In Section 3, we provide a brief background of the proposed system. In Section 4, the proposed system is explained in detail, along with the tools used and its implementation. In Section 5, the results from our experiments are shared. Results are discussed in the Conclusion section, and avenues for improvements are included in the Future Work section.

II. Related Works

Privacy preserving data analysis over blockchain technology has been the topic of study in many articles in recent years. In the research [5], a theoretical perspective on the application of homomorphic operations in smart contracts has been developed, and an exemplary architecture has been presented. The consistency of this theoretical work reveals that different homomorphic cryptosystems in blockchain systems can be applied. There are also studies showing that blockchain systems are used in projects where a large amount of data needs to be processed. One of these studies, the article [6], offers a perspective on how machine learning algorithm calculations can be made in blockchain systems.

Similar studies were also conducted for IoT applications. Reference [7] is one of the articles that encrypts IoT data with homomorphic encryption and sends it to the blockchain. They conducted a study on the homomorphic aggregation of IoT data. This study, which also schematized the system-wide data collection method, implemented the project on a private Ethereum blockchain.

Blockchain is considered a suitable system for electronic voting due to the principle of protecting data integrity. However, the privacy of the data is not protected in blockchain systems. Reference [8] proposes to count the votes by protecting the confidentiality of the data with homomorphic encryption. The study focuses on the chain node structure for their specific use case.

In [9], a system is proposed in order to securely calculate the count, mean, variant and skewness of private health data using Paillier homomorphic cryptosystem in the Hyperledger Fabric blockchain system. The system is implemented on a consortium blockchain where only parties that are part of the consortium are allowed on the blockchain. The authors develop a REST application for user requests to blockchain. According to their experimental results, requests are made within a reasonable time and security needs are met, hence their work is feasible in real life applications. Our work builds on this study. We also use the Paillier cryptosystem on a blockchain platform for secure data analysis. We expand the study to the public blockchain network Ethereum allowing a wider range of users to have access to the system. We also include linear regression in our implementation. Our results are also promising for practical applications in terms of time needed for data sharing and analysis.

Our research shows that linear regression calculation is not a statistical equation previously calculated using homomorphic encryption in the Ethereum blockchain. Therefore, our study differs from all studies in terms of linear regression calculation in blockchain systems. Although related works contains homomorphic cryptosystems, in many articles, instead of public blockchain systems, the private or consortium systems are used. Hence, our research differs from previous studies in terms of the public Ethereum network and linear regression calculation.

III. Background

A. *Ethereum Blockchain and Smart Contract*

Blockchain is a digital ledger consisting of blocks that are chained to each other. There are some characteristic properties of blockchain systems, such as decentralization, persistency, anonymity and auditability [2]. The blockchain structure is based on the decentralization principle, means there is not any central administration. Each node holds the record of all transactions. If one transaction is approved by the blockchain system that means it is recorded to all nodes. Recorded transactions cannot be changed because each block is connected to the next block with its own hash value. If one transaction is changed, the hash value is damaged, so the chain architecture is broken. This solid structure in the blockchain is called tamper resistance or persistency. Since transactions in the blockchain are recorded in nodes and cannot be changed, they can be traced back through the records in nodes. This provides auditability. In blockchain, all transactions are made with an address. Every user who wants to communicate with can operate with more than one address value. Since there is not any connection between the user's real identity and the address values, it is said that blockchain provides anonymity.

The Ethereum blockchain has a structure that allows application development and can run smart contract codes. In the white paper of Ethereum, the statement “Smart contracts, cryptographic "boxes" that contain value and only unlock it if certain conditions are met ...” is declared [10, p. 1]. A smart contract acts like an account in the blockchain and has a bunch of functions run when they are called. Different software languages can be used to code an Ethereum smart contract, but the Solidity software language is the most widely used. There are two different outputs of Solidity compiler; ABI and byte code which are explained in detail under the Implementation section.

B. *Paillier Homomorphic Cryptosystem*

Homomorphic encryption allows certain mathematical operations to be performed on encrypted data without compromising it. It is especially suitable in cases where the data is processed by third parties who are not trusted or not authorized to see the data. The output of homomorphic operations is the encrypted form of the resulting data. In this study, we use the Paillier homomorphic cryptosystem [11]. Paillier system allows addition on encrypted data without having to decrypt the data. This property is formulated as

$$E\ (m_1 + m_2) = E\ (m_1) \times E\ (m_2) \qquad (1)$$

where E stands for encryption, m_1 and m_2 represent two plaintexts.

C. *Linear Regression*

Linear regression is a statistical data analysis technique that estimates the value of unknown data using known data values. It mathematically models the dependent variable in terms of the independent variables as a linear equation. The formulas for linear regression parameters are:

$$y = a_0 + a_1 x \qquad (2)$$

$$a_0 = \bar{y} - a_1\bar{x}\ ,\ \bar{x} = \frac{\sum x_i}{n}\ ,\quad \bar{y} = \frac{\sum y_i}{n} \qquad (3)$$

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum {x_i}^2 - (\sum x_i)^2} \qquad (4)$$

IV. Methodology

In this work, we propose a blockchain-based data-sharing system that uses smart contracts and homomorphic encryption to allow statistical computations on encrypted data. In this section, we describe the system, how the components of the system interact with each other, and our implementation of the system for linear regression.

A. *Proposed System*

Our proposed system allows third parties to do certain types of analysis on data that they are not authorized to view. To give some context to our work, we assume a scenario where distributed edge devices aggregate data from sensors. The edge devices keep the data in aggregated form. We assume a third party wants to do statistical analysis on the sensor data. Following the terminology in [9], we call the edge devices “data owners” and the third party a “researcher”. The other actor in our system is the “contract developer” who deploys smart contracts for researchers on the blockchain. The smart contracts gather the encrypted aggregated data from the data owners and get them ready for the researcher using homomorphic operations. We assume that the sensor data coming from each data owner consists of pairs (x_i, y_i), and the researcher is interested in the linear regression equation for the total sensor data. The formulas for linear regression parameters are provided in (3) and (4) in Section 3.

The sensor data is aggregated in distributed data owners connected to the blockchain. To illustrate how the proposed system works, we provide Figure 1, where four data owners and a researcher interact with the smart contract. In the figure, $x_{i,j}$ represents the x-coordinate of the jth data point from the ith data owner and $y_{i,j}$ represents the y -coordinate of the jth data point from the ith data owner. The sums are over finitely many data points, and data owners may have different numbers of data points; however, the limits are not included with the sums in the diagram for the sake of simplicity.

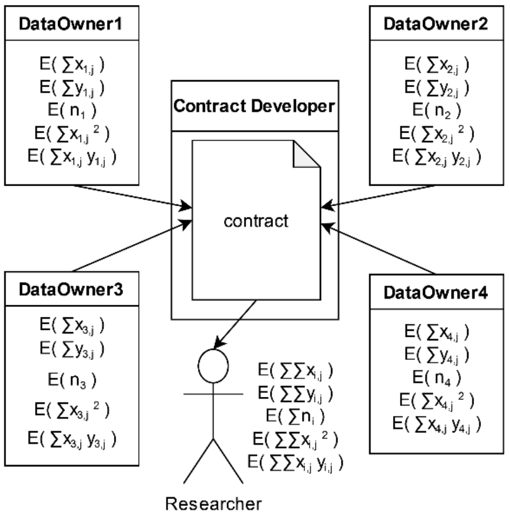

Figure-1: Data Flow in the Blockchain

The encrypted data sent by a data owner contains certain sums required in the calculation of the linear regression parameters. As seen in (3) and (4), five different sums are needed for the calculation of the parameters a_0 and a_1. These are sums of x's, y's, n's, x^2's and xy's. Each data owner encrypts the relevant sums of its data points and sends it to the smart contract. These encrypted sums are stored in the smart contract, and each type of encrypted sums are homomorphically added (i.e. multiplied) together to get the total encrypted sum for that type. For example, the encrypted sum of all x's for the four data owners is calculated as:

$$E\left(\sum_{j=1}^{n_1} x_{1,j}\right) \times E\left(\sum_{j=1}^{n_2} x_{2,j}\right) \times E\left(\sum_{j=1}^{n_3} x_{3,j}\right) \times E\left(\sum_{j=1}^{n_4} x_{4,j}\right)$$

The homomorphic property of Paillier ensures that the result of the calculation above is the same as the encrypted sum of all x's:

$$E\left(\sum_{j=1}^{n_1} x_{1,j} + \sum_{j=1}^{n_2} x_{2,j} + \sum_{j=1}^{n_3} x_{3,j} + \sum_{j=1}^{n_4} x_{4,j}\right)$$

More generally, for k data owners with n_i data points for data owner $i,$ we have the equation:

$$E\left(\sum_{1=1}^{k}\sum_{j=1}^{n_i} x_{i,j}\right) = \prod_{i=1}^{k}\left(E\left(\sum_{j=1}^{n_i} x_{i,j}\right)\right)$$

After the encrypted sums are calculated on the blockchain, they are available for the researcher. Upon receiving the sums, the researcher decrypts the sums using their private key and plugs them into equations (3) and (4) to get the parameters for linear regression.

B. *Implementation*

We implemented the proposed system on the Ethereum test network Goerli using the Paillier cryptosystem. The tools we used to develop and interact with the smart contracts are:

- *Web3.0:* Also called Web3. It is a library used for interacting with the blockchain system from the outside.
- *Remix:* To develop and deploy the smart contract, the online Remix platform is chosen.
- *Metamask:* This is the wallet we use to connect to Ethereum blockchain. Metamask account was introduced on Remix to deploy the smart contract.
- *Visual Studio Code:* It is used for developing user JavaScript applications.

We provide an overview of how the three actors in our system, namely the smart contract developer, data owner, and researcher, interact with the blockchain in Figure 2. The contract developer codes the smart contract in Solidity and deploys it to the blockchain using the Web3 library. The compilation of the smart contract produces the Application Binary Interface (ABI) for the contract, which allows other users to interact with the smart contract. In our implementation, we use JavaScript to develop the scripts to interact with the smart contract.

Because the Paillier Cryptosystem needs calculations with big integers, both smart contract and user applications need big integer libraries. In our implementation, we used BigNumbers.sol library [12] for the smart contract and BigInteger.js library [13] for the scripts used in the data owner and researcher applications.

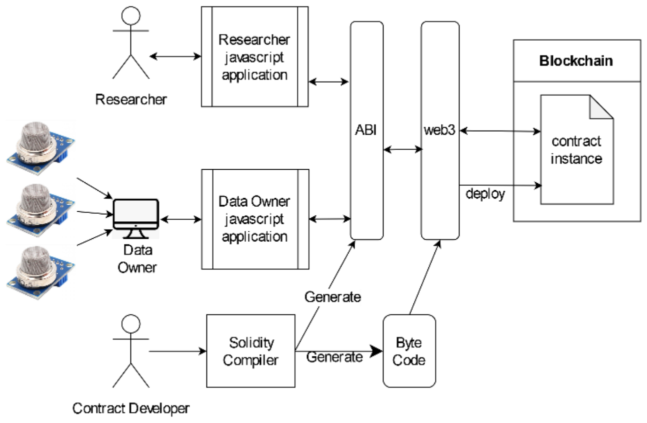

Fig. 2. Overview of interactions of actors with the smart contract

A sequence diagram depicting the flow in the proposed system is given in Figure 3. The diagram includes a smart contract that is already deployed on the blockchain to facilitate the sharing of the data between the researcher and the data owners. In the diagram , there is a researcher who wants to do linear regression analysis on the data (using equation (2) in section III.C), and a data owner who provides the data to be used in this analysis.

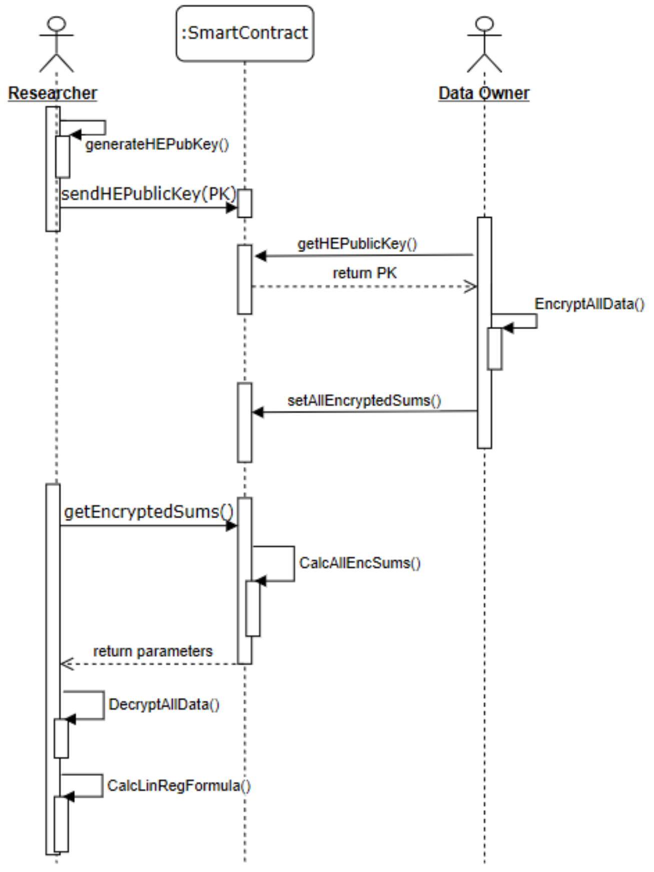

Fig. 3. Overview of the flow of data sharing

According to the flow of data in Figure 3, the researcher generates the homomorphic key pairs and records the homomorphic public key in the smart contract. The data owner gets this public key from the smart contract and encrypts the sum of the data values used in linear regression parameter calculations with the researcher's public key and saves the results to the smart contract. Each data owner follows the same steps and sends its encrypted sums to the smart contract. The smart contract "adds" the encrypted sums it receives from the data owners to get the encrypted totals of all data points. The "add" operation is the homomorphic addition and, in the Paillier system, is indeed a modular multiplication operation. The pseudocode for the homomorphic addition operation on the smart contract is provided in Figure 4. The encrypted totals are what the researcher needs to calculate the parameters for linear regression analysis. Upon receiving the encrypted totals, the researcher decrypts them using their private key and plugs them in equation (3) and (4) to find the regression parameters.

The pseudocode of the homomorphic addition function to add the encrypted values stored in an array in the smart contract is as follows:

```
bytes[] arrayOfEncData
BigNumber data, nextData, n2

FUNCTION calculateEncSumofArray(){
        data = arrayOfEncData[0] in the type of BigNumber
        FOR i = 1 to length of the arrayOfEncData DO
                nextData = arrayOfEncData[i] in the
                        type of BigNumber
                data = modular multiplication of data and
                   nextData with respect to modulus $n^2$
        ENDFOR
        RETURN data
}
```

Fig. 4. Pseudocode of homomorphic addition in smart contract

V. Experimental Results

We tested our system with respect to the key length as well as the number of data owners. We recorded two timings:

Timing 1. The average time it takes data owners to encrypt their data and send them to the smart contract.

Timing 2. The time it takes for the smart contract to homomorphically add the encrypted sums and send the result to the researcher.

The timings are tabulated in Tables I-III for key sizes 256-bit, 512-bit, and 1024 bit, respectively. Our results show that, for a given key size, for each data owner it takes 74 to 87 seconds to encrypt and send the data to the smart contract, and seconds to do homomorphic additions on the encrypted data on the smart contract and send it to the researcher. The timings indicate that using the Ethereum blockchain for the proposed system is feasible in real life applications.

Table I. 256-bit key length

Number of Data Owners	4	8	12	16
Timing1(sec. msec)	80.068	85.441	79.846	86.107
Timing2 (seconds)	2.417	2.455	2.343	2.386

Table II. 512-bit key length

Number of Data Owners	4	8	12	16
Timing1 (sec. msec)	74.287	86.764	81.894	86.087
Timing2 (seconds)	2.300	2.387	3.023	3.072

Table III. 1024-bit key length

Number of Data Owners	4	8	12	16
Timing1 (sec. msec)	87.019	86.863	85.811	83.091
Timing2(seconds)	2.379	2.381	2.436	2.687

VI. Conclusion

In this study, we proposed a system based on blockchain technology with smart contracts and homomorphic encryption that allows third parties to do analysis on data that they are not authorized to view. We implemented the proposed system on the Ethereum platform and used the Paillier cryptosystem to allow third parties to do linear regression analysis on the data.

With only a bit over than a minute for data sharing for a data owner, our test results indicate that the system is feasible for real-life applications. The fact that it takes slightly over a minute to encrypt and send the encrypted data to the smart contract is not surprising because each sending of data is a transaction and transactions on the blockchain take time. In the case of Ethereum Goerli testnet, depending on the intensity of use and gas prices, a transaction can take 15-30 seconds or more. In our implementation, because each sum is sent as a separate transaction and 5 sums need to be sent by each data owner, the sending of data by an owner takes more than a minute. This can be improved in future implementations by having the necessary data sent by the owner at once.

The homomorphic calculations on the smart contract to find the total encrypted sums take only seconds making the system appealing for practical applications.

The use of blockchain technology brings the much desired security properties of fault-tolerance, tamper-resistance, and accountability to the system. In addition, the use of a public blockchain allows data owners to have an open system where interested third parties have easy access to data that they need for analysis. The most significant contribution of the study is that all parties have, following the principle of least privilege, the minimal access they need to the data: The data owner stores only the aggregated sums of the plain data; the contract stores only the encrypted sums of the data, and the third party ultimately gets the total sums of all data. On the blockchain network, all that is transmitted about the data is the encrypted sums from the data owners and the encrypted total sums never exposing any individual data points or sums in plain form.

VII. Future Works

The system can be extended to include other analysis on data including AI and machine learning algorithms. Paillier cryptosystem is a partially homomorphic system that allows for addition of plaintexts hence is not suitable for systems where more complex analysis involving other operations is needed. For those use cases, fully homomorphic algorithms over blockchain technology can be studied.

In our scenario, there is only one smart contract designed for one researcher. As an expanded scenario, more than one researcher can be added to the system. This can be done in one of these two ways:

1. A new smart contract is deployed when a researcher is interested in the data: For this, a manager contract can be created that accepts requests from researchers and creates a new smart contract for each researcher. The smart contract stores the public key of the researcher that it is created for.

2. One smart contract can serve both as the manager and the facilitator for responses to the requests from researchers. In this method the public keys of all researchers are stored on the same smart contract.

Depending on the application, each may have benefits: The first method has the advantage of separating the tasks of management and facilitation. If used for researchers interested in different types of data and analysis, the first method will allow for easy customization of the smart contract created for them. In cases where the system is used for researchers interested in similar types of analysis with data, then keeping all the functionality in the same smart contract may make managing the contract and requests easier. A method to manage the keys and facilitate the receiving of the data encrypted with the key will need to be included in the contract.

References

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” *SSRN Electronic Journal,* 2008.

[2] M. Krichen, M. Ammi, A. Mihoub, and M. Almutiq, “Blockchain for modern applications: A survey,” *Sensors,* vol. 22, no. 14, p. 5274, 2022.

[3] G. Hilary, “Blockchain: Security and confidentiality.,” *SSRN Electronic Journal,* 2018.

[4] H. Ç. Bozduman and E. Afacan, “Simulation of a homomorphic encryption system,” *Applied Mathematics and Nonlinear Sciences,* vol. 5, no. 1, pp. 479–484, 2020.

[5] C. Regueiro, I. Seco, S. de Diego, O. Lage, and L. Etxebarria, “Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption,” *Information Processing & Management,* vol. 58, no. 6, p. 102745, 2021.

[6] A. Mitra, B. Bera, A. K. Das, S. S. Jamal, and I. You, “Impact on blockchain-based AI/ML-enabled big data analytics for Cognitive Internet of Things Environment,” *Computer Communications,* vol. 197, pp. 173–185, 2023.

[7] F. Loukil, C. Ghedira-Guegan, K. Boukadi, and A.-N. Benharkat, “Privacy-preserving IOT data aggregation based on blockchain and homomorphic encryption,” *Sensors,* vol. 21, no. 7, p. 2452, 2021.

[8] B. U. Umar, O. M. Olaniyi, D. O. Olajide, and E. M. Dogo, “Paillier cryptosystem based Chainnode for secure electronic voting,” *Frontiers in Blockchain,* vol. 5, 2022.

[9] M. Ghadamyari, "Privacy-Preserving Statistical Analysis of Health Data Using Paillier Homomorphic Encryption and Permissioned Blockchain," *Electronic Theses and Dissertations.* 8139.

[10] “Ethereum whitepaper,” *ethereum.org*. [Online]. Available: https://ethereum.org/en/whitepaper/. [Accessed: 29-Jan-2023].

[11] T. Sridokmai and S. Prakancharoen, “The homomorphic other property of Paillier cryptosystem,” *2015 International Conference on Science and Technology (TICST),* 2015.

[12] Firoorg, “Firoorg/solidity-bignumber: Full BigNumber library implementation for solidity.,” *GitHub*. [Online]. Available: https://github.com/firoorg/solidity-BigNumber. [Accessed: 29-Jan-2023].

[13] Peterolson, “Peterolson/BigInteger.js: An arbitrary length integer library for Javascript,” *GitHub*. [Online]. Available: https://github.com/peterolson/BigInteger.js. [Accessed: 29-Jan-2023].