A Privacy Preserving Internet of Things Smart Healthcare Financial System

Singh, Rajani; Dwivedi, Ashutosh Dhar; Srivastava, Gautam; Chatterjee, Pushpita; Lin, Jerry Chun Wei

Published in:
IEEE Internet of Things Journal

DOI (link to publication from Publisher):
10.1109/JIOT.2022.3233783

Publication date:
2023

Document Version
Accepted author manuscript, peer reviewed version

A Privacy Preserving IoT based Smart Healthcare Financial System

Rajani Singh, Ashutosh Dhar Dwivedi, Gautam Srivastava, Pushpita Chatterjee

***Abstract*—Several emerging areas like sensor networks, the Internet of Things (IoT), and distributed networks are gaining traction where resource-constrained devices communicate by sharing privacy-preserving information. Due to heavy cryptographic components, standard cryptographic algorithms do not fit these IoT devices. In this paper, we propose an efficient zero-knowledge blockchain-based privacy-preserving decentralized smart healthcare finance system that is suitable for lightweight computer devices. The proposed design mainly focuses on non-interactive zero-knowledge proof, which substantially reduces the cost of communication between two devices. We explain the system framework and its use-case for a healthcare financial system at a micro-level. However, it can easily be extended to more general financial systems as well. Our system framework is very fast and lightweight, using more efficient zero knowledge-based proofs; validation of the transactions is done in milliseconds. As an advancement to our work, the proposed healthcare financial system for lightweight computer devices is also auditable without leaking any extra information than required.**

***Index Terms*—Zero-knowledge, Distributed ledger, Blockchain, Privacy based Smart Healthcare**

I. Introduction

In today's healthcare environment, financial realities play a vital role in making decisions for both the patients and healthcare providers. Any medical procedure or procurement meant to improve a person's well-being is known as healthcare, and management of funds for these medical resources is called Healthcare financing. To be more specific, it includes all payments related to dental care, hospital care, physician care prescriptions, and other medical services. An intelligent healthcare financing system has the potential to enhance the financial well-being of health institutions. In health service organizations, healthcare finance is about financing the healthcare system and consists of both the financial and the accounting management functions. Accounting of a health organization concerns recording economic events that reflect the operations, assets, and financing. The main purpose of accounting is to provide useful information of the health's organization operations and financial status to the interested parties. These parties can be both internal such as managers, and external such as investors and other stakeholders. On the other hand, financial management provides the concepts, theory, and tools necessary to help healthcare managers make wise decisions. Healthcare financing is all about how society pays for the healthcare services it consumes. When payment collection responsibility is solely on the healthcare providers, the provider-patient relationship may become stressful. To maintain a positive relationship, providers must delicately balance the healthcare information and collect balances owed by the patient. On the one hand, providers can have healthy financial conversations with their patients with digital and flexible financing options. On the other hand, the patient can have a sustainable, budget-friendly payment plan as they know the terms, conditions, and payment deadlines in advance. Blockchain is the most famous distributed system technology making headline globally through Ethereum, Bitcoin, and several other cryptocurrencies. Due to its decentralized behavior, blockchain is treated as the most secure, trusted, and immutable system to store transactions. Blockchain technology is already in use by several industries including banking [7], voting[17], supply chain[16] etc. Blockchain has great potential to assist any financial system by using cryptocurrencies. In this work, we considered a healthcare financial system that supports cryptocurrencies. The main key feature of blockchain is its transparency, but on the other hand, it has a serious side-effect on users' privacy. Nowadays, millions of devices are connected with the blockchain network, where users share more and more information every day. These pieces of information cannot be removed from the network in the future, and therefore, anyone may be willing to collect this information for business purposes, for example, an insurance company. The other issue with financial systems is auditing. A financial system should be something that can be auditable without revealing any extra sensitive information, i.e. if a non-governmental organization, widely known as NGO pays some donation to any hospital, then the NGO can audit the financial system of the hospital without knowing the other sensitive payments of hospital ledger that are not related with NGOs money. Similarly, a government official can also audit the finance system of hospitals or NGOs without knowing sensitive information apart from what they want (see Figure 1).

Therefore, the major goal of this work is to develop a zero-knowledge based privacy-preserving financial system that is most suitable for auditing purpose without revealing sensitive information. To understand Zero-knowledge, let's take an example without going much into technical details. Consider a straightforward game of where is Waldo without sharing his location. Alice and Bob compete to find Waldo from the picture in a kid's book series. Suppose Alice spots Waldo and tell Alice that I know where Waldo is. However, she does not

Rajani Singh and Ashutosh Dhar Dwivedi is with Centre for Business Data Analytics, Department of Digitization, Copenhagen Business School, Denmark. E-mail: rs.digi@cbs.dk, add.digi@cbs.dk

Gautam Srivastava is with Department of Math and Computer Science, Brandon University, Canada. E-mail: srivastavag@brandonu.ca

Pushpita Chatterjee is with Department of Computer Science, Tennessee State University, Nashville, USA. E-mail: pchatter@tstate.edu

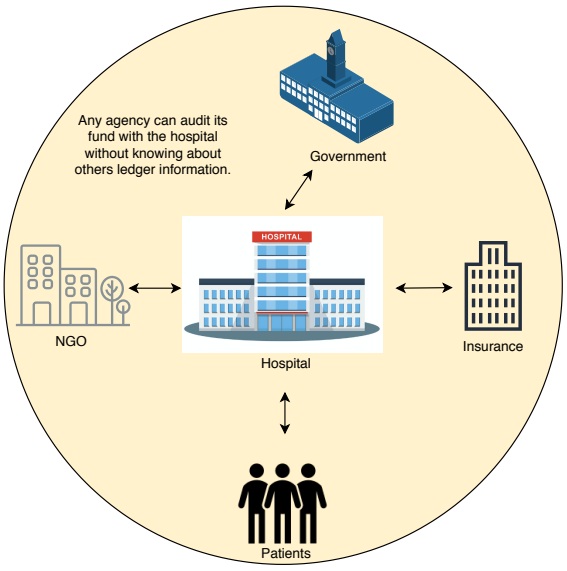

Fig. 1. Healthcare Financial System

tell him the exact position or location of Waldo; instead, she will convince him that she knows Waldo by giving some kind of proof. This is similar to prove to Bob that Alice spent 50 dollars for coffee on a certain day without revealing the whole month's payment details.

For example, Alice can cut out Waldo from the picture she has and only shows the snipped Waldo to Bob. To ensure that she did not take just a printout of a new picture of Waldo, Bob can watermark the back of Alice's scene page. Or, he can do a thorough deep search on Alice before she enters a secret room to cut the page. Alternatively, Alice can cut a hole in a huge, opaque sheet of cardboard. Then, she places the cardboard cutout on top of the original picture. This way, only Waldo is shown, but the coordinates or location relative to the rest of the scene is hidden. Later, she can reproduce the picture underneath to prove that she used the original puzzle picture.

Several researchers have already deployed privacy-preserving cryptographic techniques based on zero-knowledge proofs into the transactions between cryptocurrencies. To the best of the author's knowledge, all these transactions are monetary based and provide privacy to the money-related transactions. These proofs are only used for integer values and thus cannot be applied to hide the decimal values. We tried to use the technique of the zero-knowledge proof at an advanced level by providing the same level of privacy and security to the healthcare data and money transactions. Moreover, our zero-knowledge proofs can also be used for the case when the transaction amounts are in decimal. Our work is novel in the sense that it provides privacy to the healthcare data also. Moreover, the zero-knowledge proofs used here will also work for decimal values. Our healthcare financial system is auditable, any authenticated user can audit the system. Therefore, our financial system helps to maintain trust among its users such as patient, healthcare providers etc.

II. Related Literature

Zcash [1] is the first privacy-focused blockchain and cryptocurrency that applies the concept of zero-knowledge proofs called zk-SNARK, proposed by Eli et al. [5] on JP Morgan Chase's payment system that is based on blockchain. They use it to authenticate clients to servers securely. It is also used for transaction privacy by hiding the transaction amount with some commitment scheme, for example, the scheme proposed by Pedersen [12]. However, it does not support auditing queries, requires a trusted setup, and not secure against quantum computers. zk-STARK also proposed by Eli et al. [4] is an extension to zk-SNARK, which does not require a trusted setup, but the proof sizes are bigger than zk-SNARK.

Polestra *et al.* [14] proposed a privacy-preserving transaction called a confidential transaction. As the name suggests, the transaction amount is confidential, so no other parties can know the outgoing amount sent by the sender called input and the incoming amount received by the receiver called output. To hide the input and output of the transaction, they use the Pedersen commitment scheme defined on the elliptic curve over the finite field. This commitment scheme is popular because of its homomorphic property. However, anyone can verify if the transaction is valid or not. For this, the authors use the zero-knowledge proof that the sum of committed input is greater than or equal to the sum of committed output, and all the received amounts are positive more specifically, they lie in the range $[0, 2^n]$.

Mimblewimble [13] is a recent improvement to the confidential transaction. A drawback of their work is that it exposes the transaction graph, which can leak substantial information, for example, one can track the origin of the input. Moreover, it does not support private auditing.

Tomaz *et al.*[18] proposed a privacy-preserving mobile health system using non-Interactive zero-knowledge proof and blockchain. The authors address the issues of managing, storing and sharing data using decentralized blockchain. Authors used attribute-based encryption to protect health data. The paper's outcome provides fine-grained access control, entirely governed by the patient and end-to-end privacy. In another work, Androulaki *et al.* [3] presented a privacy preserving token management system that was suitable for permissioned blockchains. The system can be instantiated with easy setup.

Partala *et al.* [11] presented a survey paper for non-interactive zero knowledge proof and their applications in private smart contracts and confidential transactions on the blockchain. The authors briefly discussed the background of zero knowledge, asymptotic computational complexities and proof lengths, and cryptographic security models. They also discussed existing circuit generation tools that are needed to transform the computation into a circuit representation.

Lin *et al.* [8] proposed a framework called "Blockchain based Personally Identifiable Information (PII) Management System (BcPIIMS)". The proposed system can track the life cycle of Personally Identifiable Information throughout controllers and processors. The proposed model provide privacy and security.

Narula *et al.* [9] proposed a Blockchain and zero- knowledge proofs based secure and privacy preserving banking

Fig. 2. Example of Zero-Knowledge Game

system. Although, their financial system facilitate unbiased auditing , it has scalability issue.

III. Our Contributions

To the best of the author's knowledge, with regards to healthcare financial systems, the application of blockchain technology and zero-knowledge proofs to preserve privacy has not been investigated in any of the current existing literature. Therefore, this paper presents a zero-knowledge blockchain-based privacy-preserving financial system for use in healthcare financial systems. Although healthcare organizations are working for a public cause, the financial details of these hospitals and the beneficiary's information are quite sensitive. So, zero-knowledge proofs can be used to provide a high level of privacy to blockchain users. However, none of the hospitals started using the blockchain and zero-knowledge based system in their financial system because of a lack of knowledge in this area. Therefore, in the paper, we focused on improving the healthcare financial system, but our framework can be applied to any financial system where auditing is also required. Therefore, our financial system is realistic and can also be implemented as a prototype which is in our plan. Such a prototype will help the organizations drastically improve their financial system, the auditing process, and most importantly, build and maintain trust among the different stakeholders such as fund providers, etc. Moreover, we have improved the efficiency of our blockchain zero ledgers by using the more efficient range proof technique than the one used in zkLedger. So, our system is much faster than the original zkLedger based audit system. In summary, we try to provide an overall understanding of the framework. Showing practical proof will be very comprehensive and will lead to another research work. Therefore, here we restrict ourselves to provide theoretical explanations. In fact, to do the same in practice and developing a prototype is our next plan.

IV. A Zero-Knowledge based blockchain for Healthcare

On the one hand, there are several key features of blockchain that make it advantageous to use as a distributed ledger in the proposed system, while on the other hand, several issues are also available that make blockchain unsuitable for applying as a distributed ledger for healthcare financial systems. Therefore, while applying blockchain as a distributed ledger in the proposed model, it is necessary to address all issues properly. Our notations table used in the paper is given in Table **??**. Following are the key features and drawbacks of blockchain:

A. Key features of existing blockchain

- **Decentralization :** In a centralized network, transactions are validated by a third party, and therefore peers engaged in the transaction have to trust central authorities to maintain a record or perform authorization. Blockchain distributed ledger plays an important role in making trust among all peers due to its decentralized and distributed behavior. In a blockchain, users are connected in a decentralized manner through the peer to peer network. So, there is no need to have a middleman or third person to do any transaction.
- **Immutability:** All the peers in the blockchain network agree for new transaction entries by using a decentralized consensus. All transactions are stored in a chain of blocks that are connected by cryptographic hash function and, therefore very hard to tamper with.
- **Transparency:** Blockchain ledger of blockchain, containing the transaction record or data, is visible to all of its users or peers. Therefore the system is transparent for all its users.
- **Enhanced Security:** Due to the decentralized property of blockchain, no one from the network can change the characteristics or information stored in the blockchain network. Encryption provides another level of security for the whole system. Blockchain uses several special features of cryptography and therefore provide a very secure network.
- **Auditability:** The ledger used by blockchain is distributed and therefore all peers in the network hold the same copy of the blockchain. Due to the distributed nature of the ledger, anyone can access it from the network and can verify the transactions stored in the blocks.
- **Fault tolerance:** The identical replicas of the ledger is stored in all peers and therefore any data leakage or any

kind of fault in the ledger can be easily identified. If any node in the network lost the local blockchain data then it can be easily recovered with other copies.

B. Drawbacks of existing blockchain

- **Anonymity and data privacy:** On the one hand, transparency is the key feature of blockchain where any node in the network can check each transaction, but on the other hand, transparency has a knock-on effect on the privacy of blockchain. Although most of the cryptocurrencies, e.g. Bitcoin, hides assets and amount but leak the transaction graph. Some other cryptocurrencies, e.g. Zerocash, uses zk-SNARKs to hide the amount, transaction graph, and participants in the network but these policies do not support arbitrary queries and therefore cannot be useful for other practical systems.
- **Scalability and storage capacity:** The scalability of the blockchain and storage capacity is the major challenge of the current blockchain system. The transaction throughput of bitcoin is approximately 3-5 TPS. Similarly, Ethereum has a throughput of 15 to 20 TPS. The number of transactions per second is not sufficient in many cases. The Bank VISA card system can be an example that is made to handle approximately 2000 to 4000 TPS. On the other hand, the chain of this technology grows continuously, and the copies stored among peers consume a lot of storage space.

V. Proposed System Architecture

For blockchain's scalability issue, the proposed framework uses permissioned Hyperledger Fabric blockchain with a Raft consensus algorithm. Transactions are stored on blockchain but to store other data, cloud storage is used. In the proposed framework, the major challenge of the financial system is the privacy of ledger with arbitrary queries by stakeholders or other entities. To address the privacy issue, the system is using a zero knowledge-based blockchain ledger. The model is inspired by the work of Narula et al. [10] and Bunz et al. [6] with major modifications.

A. Blockchain architecture:

Scalability of blockchain is the major issue with several pubic blockchains like Ethereum and Bitcoin. However, in the proposed model of healthcare finance system, we are using consortium blockchain (Hyperledger) that has some property of the public and some private. These type of blockchain use *permissioned* network where only permissioned and a limited number of nodes are allowed to join the network. Multiple organizations are also allowed to join the same network with different channels and separate privacy settings. Hyperledger[2] is the first extensible blockchain system for distributed applications that support several components, such as membership services and consensus algorithms to plug and play. For small organizations, Hyperledger Fabric is the best choice that supports more than 400 prototypes for distributed ledger technology currently. Selection of the consensus algorithm is another important issue that has to be addressed. To provide high throughput, Practical Byzantine Fault Tolerance (pBFT) consensus algorithms are used here. pBFT provides high-performance Byzantine state machine replication. Due to the use of high-performance Byzantine machine, pBFT throughput has increased to a thousand transactions per second. pBFT consensus has fault tolerance of 33% that means, the system performs very well until one-third nodes become malicious. The network consists of client and replica nodes (see Figure 3). Nodes are sequentially ordered where one node works as a leader while other nodes known as backup nodes. Operations in pBFT can be divided into four phases:

- Client (C) sends the request to the leader node (0 in this case) to invoke service.
- Leader node broadcast the message to other backup nodes (1,2,3 in this case).
- After executing the request, all the nodes sends reply to Client.
- Client expects the same results from $f + 1$ nodes, f is the number of malicious nodes.

Leader node can be replaced if it does not broadcast the message in a certain time. If the leader is changed by the protocol, the process is called view change. The biggest drawback of pBFT is network scalability. pBFT is not suitable for a very large network but only support limited nodes and therefore for healthcare systems or any other B2B organizations, it is the most suitable consensus algorithm. The problem with large network is a huge communication cost, as pBFT performs voting based decision where each node must communicate with each other to keep the network secure.

B. Cloud based storage:

If the hospitals/other entities purchases any good or service from the goods or service provider, then it needs to have a receipt or bill as evidence of purchase. Moreover, the patients also needs a piece of evidence for the service. All these data containing the evidence is needed to be stored and accessible by all of its blockchain system participants. However, the blocks in blockchain have limited space for data storage with at most 1 MB. So, we cannot save these receipts into the blockchain as they could be in large numbers.

Therefore, we use the cloud(see Figure 4) to store these receipts in encrypted form and save the hash of these encrypted receipts into the blockchain ledger. By this way, we prevent the data tampering, meaning that if any malicious participant or user manipulates these cloud data, the new hash will generate immediately after such tempering and therefore, it will be immediately noticed by the other system users as the new hash will not match with the old hash.

C. Zero Knowledge Proofs (ZKP) for privacy

The most important key challenge in the financial system is to provide secure transactions and data sharing while keeping sensitive information confidential to preserve the privacy of the system users. ZKP offers a lot of possibilities to improve privacy while sharing financial data among stakeholders. For example, it allows the prover to prove that he has the right

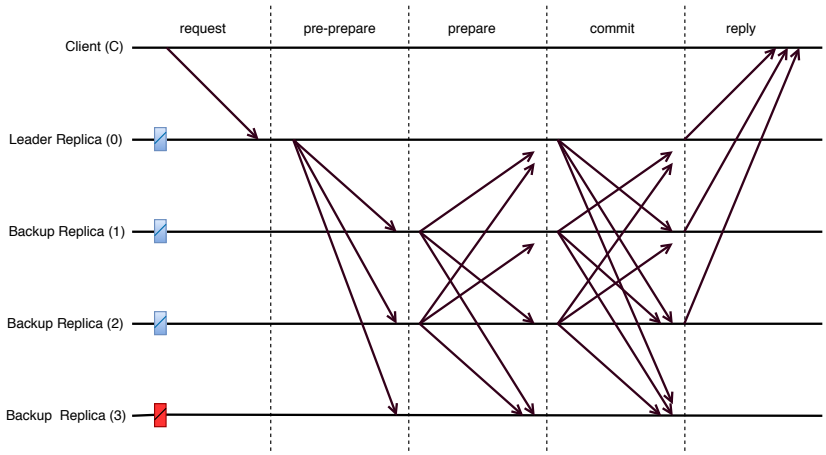

Fig. 3. Practical Byzantine Fault Tolerance (pBFT)

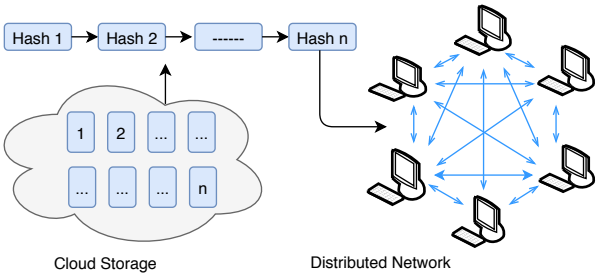

Fig. 4. Cloud Storage

to access some information without revealing his identity or enables the information or data holder to prove its ownership of information or data while keeping his identity secret. Zero-knowledge proofs or protocol for any problem or statement consists of three process namely: *commitment*, *challenge* and *verify*:

- **Commitment:** In the commitment process, the prover commits to the secret value that he wants to hide. Commitment value is nothing but the encrypted form of the secret value, which ensures the verifier that he will not be able to change his secret value later on. So, the prover generates the commitment using any commitment scheme and sends it to the verifier.
- **Challenge:** Once the verifier receives the commitment value, he sends a checkpoint or queries for the problem called a challenge to provide the answer to the challenge. By doing so, the verifier wants to be sure whether the prover is bluffing by sending some wrong value or not. Verifier can send multiple queries to the prover, however sending different queries and getting the response for them may be time-consuming moreover, it can also leak the part of the solution to the given problem. Therefore, a non-interactive way is becoming quite popular because it reduces communication cost and lowers the risk of information leakage.
- **Verify:** The third and last phase of the protocol is to verify where the verifier validates the solution to the sent challenge. If it is correct, the verifier will get convinced with overwhelming probability about the statement validity or the problem's solution.

Definition 1: Zero-knowledge proof or ZKP is a cryptographic mechanism allowing one party (prover) to prove to another party (verifier) that the given statement is true without revealing any information about the statement.

In other words, the prover tries to convince the verifier that he knows something without revealing what it is by providing zero-knowledge proof to him. By using ZKP, the prover can prove to a verifier that he possesses knowledge of certain secret information without revealing the secret information or any additional information. ZKP has the following three properties:

1) **Completeness:** If the prover is honest about the statement, he will always be able to create the correct proof for the given statement, and therefore verifier will always accept such proof with overwhelming probability.
2) **Soundness:** If the statement is wrong or false, the prover will never be able to convince the verifier that the given statement is true and therefore, the verifier will reject the proof. However, it is not true in general. So, some assumptions have to be made; for example, in our case, soundness holds under the assumption that the discrete log problem is hard.
3) **Zero-knowledge:** Proofs that the prover send to the verifier, will not leak or reveal any secret or additional information to the verifier. So, the proofs contain zero knowledge in the sense that the verifier will learn nothing about the secret.

1) Types of Zero-Knowledge Proof (ZKP) Protocol : ZKP protocols are of two types: interactive and non-interactive, as explained below:

- **Interactive:** In Interactive zero-knowledge proof, prover and verifier exchange messages with each other. The interactions are generally based on a commit-challenge-verify protocol. Firstly, Alice(prover) sends a *commitment* that she knows the solution to the problem without revealing the solution that is her secret. Then Bob(verifier) sends a *challenge* to check if she knows the answer. Using

this challenge, Alice computes the value and sends it to Bob. Finally, Bob *verify* this value and get convinced that Alice knows the solution.

- **Non-interactive:** In a non-interactive proof, a prover and verifier do not need to interact with each other. An interactive proof can be changed into non-interactive by using a hash function. Proof of the message already available in a single message sent from prover to verifier. However, not all interactive protocols can be made non-interactive in this manner. Only public coin ones. In the proposed framework, we are using non-interactive Schnorr protocol [15] with the Pedersen commitment scheme [12].

It is important to know the difference between the two frequently used cryptographic terms: *zero-knowledge* and *zero-knowledge proofs*. Zero-knowledge is used to save the data or information using the third party but keeping the content of the data or information secret from the third party. Thus, the main use-cases for Zero-knowledge are data storage and messages.

Zero-knowledge proofs are also used to keep the content hidden from a third party, in addition to that, it provides extra verification (for instance, the proof can ensure that the number lies in the range from 1 to 10, without revealing what that number is). Zero-knowledge proofs are used to prove and verify the statements about data, such as CPU intensive verifications.

2) Participants or users of system:: The blockchain-based hospital's financial system has a finite number of participants or users say N. These participants are hospitals themselves, all fund providers or donors, patients, government bodies, NGOs, all good or service providers having an agreement with hospitals to provide the goods or services at the lowest possible price or a reasonable price. We will write P_i to denote the i^{th} participants or users of the financial system. Our system is also auditable, but the role of the system participants is not distinct and explicitly defined. For example, any fund provider can be a participant as well as an auditor.

Moreover, a hospital can also audit its financial system called an internal audit. However, if the audit needs to be done by an external party other than the system participants, we can also add them to our system. In that case, the system will have N participants and 1, auditor. Our blockchain-based financial system supports certain type of queries that the auditor can ask and we will explain them later.

3) Transactions:: Transaction can be defined as the monetary exchange between any two parties. The party who sends the money is called the sender and the party who receives the money is called the receiver. These parties are referred to as N system participants in our system. In our system architecture, the sender sends money and makes a transaction by creating a digital signature. We hide the transaction amount by using the zero-knowledge proofs. To create zero-knowledge proofs, we use the Pedersen commitment scheme [12], which has an important property that it can be homomorphically combined. So, the transaction amounts are hidden from all of the system participants except the participant involved in that particular transaction. However, all the transactions are publicly verifiable by all of the system participants. User P_i with a negative amount means that he is the sender, a user with a positive amount means that he is the receiver, while the remaining $N-1$ participants with 0 amount means that they are not involved in this transaction. We add 0 into the transaction of all other participants so that an adversary cannot identify which user is involved in which transaction. Therefore, it anonymizes the transaction participants of the system.

4) Privacy preserving blockchain ledger:: We use a row-column structure to create a blockchain ledger. There are 6 columns in the ledger containing the metadata such as Transaction ID, Date, Time, Asset type, Purpose, and Receipt. Apart from this metadata, there are N columns containing the participant's information and numerous rows depending on the number of transactions. For example, if there are M transactions held, then the number of rows is also M. Each row in the ledger represents a transaction and information related to this transaction, while each column contains different information. Table I shows how the blockchain ledger will look like in plain text form. For example, we represent the receipt by R_i.

However, in the paper we use the blockchain ledger in hidden form by using the zero knowledge proofs based on the Pedersen commitment scheme. So, the ledger in hidden form will be look like as shown in Table II.

In Table I and II, *Transaction id* is the specific identifier to identify the transaction. *Date* and *time* are used for timestamping the transaction detail. Since the donor or service provider can be from all over the world with a different currency, therefore we use *asset* to show the transaction in different currencies. For example, transaction 1 and 2 are in Euro while m is in US dollar. To make the audit process more transparent, we also add the purpose of the transaction. For example, in both tables, transaction 1 and m are donations, while transaction 2 is for the hospital to buy some good or service from the provider. In Table I, we show the transaction amounts by a box to symbolically show that they are hidden, but in actual the commitment to the transaction amounts should need to be written here, which we will do in a later section. In *Receipt* column of both tables, Hash_{R_i} represents the hash of receipts or other evidence of the transaction that is being stored in the cloud. However, to access these receipts or other pieces of evidence, one needs to have the key to decrypt them as they are stored in the clouds in encrypted form.

5) Audit Token (π^T):: Each user i generates public PK_i and private or secret key sk_i using Schnorr protocol (see Definition **??**) and distributes the public key to all of the system participants.

6) Proof of an asset (π^A): : It is zero-knowledge proof that guarantees that the sender has the assets or money to transfer. For this, the prover (sender) sum all the commitment values in the column for the asset and proves that the sum is greater than or equal to 0. For this, prover provides a disjunctive proof (see Definition **??**) that either sum is greater than or equal to 0 or the creator of transaction or sender i knows the secret key for user P_i.

7) Proof of balance (π^B): : It is zero-knowledge proof that grantees no assets are created or destroyed, meaning that the transaction conserves assets. For this, the sum of committed

ID	Date	Time	Asset	Purpose	P_1	P_2	...	P_N	Receipt
1	01.01.2020	10 : 00	Euro	Donation	1000	-1000	...	0	Hash(R_1)
2	13.03.2020	14 : 30	Euro	Goods	-50	50	...	0	Hash(R_2)
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	19.06.2020	18 : 00	USD	Donation	5000	0	...	-5000	Hash(R_M)

TABLE I
BLOCKCHAIN LEDGER WITH TRANSACTION IN PLAIN-TEXT FORM

Transaction ID	Date	Time	Asset	Purpose	P_1	P_2	...	P_N	Receipt
1	01.01.2020	10 : 00	Euro	Donation	□	□	...	□	Hash(R_1)
2	13.03.2020	14 : 30	Euro	Goods	□	□	...	□	Hash(R_2)
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	19.06.2020	18 : 00	USD	Donation	□	□	...	□	Hash(R_M)

TABLE II
BLOCKCHAIN LEDGER IN OUR SYSTEM WITH HIDDEN TRANSACTION USING COMMITMENTS

values in each row should equal to 0, and the sum of all the random number used in commitment should equal to 0. Verifier simply multiplies all the commitments and checks if the multiplication is equal to 1 or not. If it equals to 1, then the verifier accepts the proof; otherwise, reject.

8) Range proof (π^R): : Commitment values belong to the cyclic group and are based on modular arithmetic. Therefore, it is necessary to check whether the committed value is in an acceptable range or not. Moreover, we also need to make sure that all the receiver receives a positive amount of money. Therefore, in the model, we need a range-proof that a committed number lies in between $(0, 2^{64})$. We require two range proofs: one for the commitment value, and another for the sum of assets in the column. We used the range proof based on improved Inner product proofs (see Definition 2) introduced by Bunz et al. [6] in their Bulletproofs paper to improve the efficiency and to reduce the computation cost.

Definition 2: **Improved Inner Product Proofs:**
In the inner product proof, the prover convinces the verifier that he knows two vectors $\vec{a}, \vec{b}$ such that their inner product is equal to v. For $\vec{g}, \vec{h} \in \mathbb{G}^n$, prover commits to the vectors $\vec{a}, \vec{b} \in \mathbb{Z}_p^n$ as

$$\text{Com}\left((\vec{a}, \vec{b}), 0\right) = \vec{g}^{\vec{a}} \cdot \vec{h}^{\vec{b}} \text{ and } v = \langle \vec{a}, \vec{b} \rangle \qquad (1)$$

for commitment $\text{Com}\left((\vec{a}, \vec{b}), 0\right) \in \mathbb{G}$ and inner product value $v \in \mathbb{Z}_p$.
Alternatively, prover can also commits to the vectors $\vec{a}, \vec{b} \in \mathbb{Z}_p^n$ as:

$$\text{Com}\left((\vec{a}, \vec{b}, v), 0\right) = \vec{g}^{\vec{a}} \cdot \vec{h}^{\vec{b}} \cdot u^v \text{ for } v = \langle \vec{a}, \vec{b} \rangle \qquad (2)$$

Prooving relation 2 is equally complex and hard to prooving the relation 1. So, we consider the proof for statement 1 throughout the work.

Note that the above vector commitments are blinding but not hiding. To prove the statement (2), the prover will follow the procedure described in Algorithm **??** and the verifier follow the procedure described in Algorithm 1.

*9) **Range proof based on Improved Inner product:**:*
Consider a simple statement to prove that a secret number v is in range $(0, 2^n)$. To prove it, the prover needs to provide a range of proof to the verifier that will guarantee the verifier that he is telling true that the value is, in fact in the range without revealing the value itself. Several range proof techniques exist in literature; however, in all of them, the bit decomposition technique used in Bulletproof paper named "proof of knowledge of vector" is the most efficient. Since any integer which is a scalar number v can be decomposed into bit form. To ensure that v lies in the interval $(0, 2^n)$, the decomposition need to be done using 2^{n-1}. For example to prove that $v = 3$ lies in interval $0, 2^4$, we can rewrite integer $v = 5$ as $5 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3$. We can also define the bit form of value $v = 5$ as one vector and call it $\vec{a}$ similarly, base 2 values as another vector and call it $\vec{b}$. So, vector $\vec{a} = (1, 0, 1, 0)$ and vector $\vec{b} = (2^0, 2^1, 2^2, 2^3)$ with $v = \vec{a} \cdot \vec{b}$. Notice that $\vec{a}. \cdot \vec{b}$ is nothing but the inner product value of two vectors. Therefore, in this scheme, forgiven v, prover convince the verifier that he knows two vectors $\vec{a}$ and $\vec{b}$ such that their inner product is equal to v that is $v = \langle \vec{a}, \vec{b} \rangle$. Prover uses the Pedersen commitment scheme but using the randomness value as zero. In range-proof, prover convince the verifier that he knows the secret v which is in range $(0, 2^n)$ or $0 \leq v < 2^n$ by publishing a commitment $\text{Com}(v, \alpha) = g^v h^\alpha$ to v.

Proving that v lies in range $(0, 2^n)$ is equivalent to proving the three statements given below:

1) For $\overrightarrow{2}^n = (1, 2^0, \ldots, 2^n)$, any number v can be written as the inner product of its bit representation v_{bits} (made of 0 and 1) and base 2 form,
$$v = \left\langle v_{bits}, \overrightarrow{2}^n \right\rangle$$
2) Let $\overrightarrow{Rv}$ be the n dimensional vector, defined as $\overrightarrow{Rv} = v_{bits}$, now define $\overrightarrow{Lv}$ as,
$$\overrightarrow{Lv} = \overrightarrow{Rv} - 1^n$$
3) v_{bits} are really composed of 0 and 1 only,
$$\langle \overrightarrow{Lv} \circ \overrightarrow{Rv} \rangle = 0^n.$$

If $n = 1$, then

1) Compute the commitment as $V := \text{Com}\left((\vec{a}, \vec{b}, c), 0\right) = \vec{g}^{\vec{a}} \cdot \vec{h}^{\vec{b}} \cdot u^v$.
2) Publish the vectors $\vec{a}, \vec{b}$ and commitment value V to the verifier.

If $n > 1$, then do the following:

1) Halves the length of all vectors as: $n' = \frac{n}{2}$ and define $\overrightarrow{Lv}$ and $\overrightarrow{Rv}$ as $\overrightarrow{Lv} = \langle \overrightarrow{La}, \overrightarrow{Rb} \rangle$ and $\overrightarrow{Rv} = \langle \overrightarrow{Ra}, \overrightarrow{Lb} \rangle$.
2) Define a Hash function $\mathcal{H}$ as

$$\mathcal{H}(\overrightarrow{La}, \overrightarrow{Ra}, \overrightarrow{Lb}, \overrightarrow{Rb}, c) = (\overrightarrow{Lg})^{\overrightarrow{La}} \cdot (\overrightarrow{Rg})^{\overrightarrow{Ra}} \cdot (\overrightarrow{Lh})^{\overrightarrow{Lb}} \cdot (\overrightarrow{Rh})^{\overrightarrow{Rb}} \cdot u^v$$

3) Using the function from Eq. (2), calculate values of $L\mathcal{H}, R\mathcal{H} \in \mathbb{G}$ as

$$L\mathcal{H} := \mathcal{H}\left(\vec{0}^{\frac{n}{2}}, \overrightarrow{La}, \overrightarrow{Rb}, \vec{0}^{\frac{n}{2}}, \langle \overrightarrow{La}, \overrightarrow{Rb} \rangle\right) = (\overrightarrow{Rg})^{\overrightarrow{La}} \cdot (\overrightarrow{Lh})^{\overrightarrow{Rb}} \cdot (u)^{\overrightarrow{Lv}}$$
$$R\mathcal{H} := \mathcal{H}\left(\overrightarrow{Ra}, \vec{0}^{\frac{n}{2}}, \vec{0}^{\frac{n}{2}}, \overrightarrow{Lb}, \langle \overrightarrow{Ra}, \overrightarrow{Lb} \rangle\right) = (\overrightarrow{Lg})^{\overrightarrow{Ra}} \cdot (\overrightarrow{Rh})^{\overrightarrow{Lb}} \cdot (u)^{\overrightarrow{Rv}}$$

4) For a random challenge $r \in \mathbb{Z}_p$, compute new vectors $\vec{a}', \vec{b}', \vec{g}', \vec{h}'$ as:

$$\vec{a}' = \overrightarrow{La} \cdot r + \overrightarrow{Ra} \cdot r^{-1}, \quad \vec{b}' = \overrightarrow{Lb} \cdot r^{-1} + \overrightarrow{Rb} \cdot r$$
$$\vec{g}' = (\overrightarrow{Lg})^{r^{-1}} \circ (\overrightarrow{Rg})^r, \quad \vec{h}' = (\overrightarrow{Lh})^r \circ (\overrightarrow{Rh})^{r^{-1}}$$

5) Compute the new commitment value as $V' = (L\mathcal{H})^{r^2} \cdot \text{Com} \cdot (R\mathcal{H})^{r^{-2}}$.
6) Publish $L\mathcal{H}, R\mathcal{H}, \vec{a}', \vec{b}', \vec{g}', \vec{h}'$ and V′, so the verifier will know all these values.
7) Repeat step 1 to 5 until $n = 1$.

Algorithm 1: Verifier Algorithm:

- If $n = 1$, then
  1) Check if the inner product value is correct by computing $v = \vec{a} \cdot \vec{b}$.
  2) Check if the commitment value is correct by recomputing $\overline{V} = \vec{g}^{\vec{a}} \cdot \vec{h}^{\vec{b}} \cdot u^v$.
  3) Accept the proof if and only if $\overline{V} = V$, otherwise reject.
- If $n > 1$, then
  1) Check if the new commitment value is correct by recomputing $\overline{V'} = (L\mathcal{H})^{r^2} \cdot V \cdot (R\mathcal{H})^{r^{-2}}$.
  2) Compute the new hash value by using new $\vec{g}', \vec{h}'$ as $\mathcal{H}' = \mathcal{H}\left(\frac{\vec{a}'}{r}, r\vec{a}', r\vec{b}', \frac{\vec{b}'}{r}, \langle \vec{a}', \vec{b}' \rangle\right)$
  3) Accept the proof if and only if $\overline{V'} = V' = \mathcal{H}'$, otherwise reject.
  4) Repeat step 1 to 3 until $n = 1$.

Algorithm 2: Range-proof: Prover Algorithm

1) For $\alpha, \beta, \gamma, \delta, \rho$ being a random number choosen from $\mathbb{Z}_p$, calculate the commitments as follows:
   Commitment to the secrets value v as $\text{Com}(v, \alpha) = g^v h^\alpha$.
   Commitment to the bit value $\overrightarrow{Rv} \in \{0, 1\}^n$ (of v) and its complement $\overrightarrow{Lv} \in \mathbb{Z}_p^n$ as $\text{Com}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta) = \vec{g}^{\overrightarrow{Lv}} \cdot \vec{h}^{\overrightarrow{Rv}} \cdot h^\beta$.
   Commitment to the blinding values $\overrightarrow{Ls}, \overrightarrow{Rs} \in \mathbb{Z}_p^n$ of $\overrightarrow{Lv}$ and $\overrightarrow{Rv}$ as $\text{Com}(\overrightarrow{Ls}, \overrightarrow{Rs}, \gamma) = \vec{g}^{\overrightarrow{Ls}} \cdot \vec{h}^{\overrightarrow{Rs}} \cdot h^\gamma$.
   Commitment to scaler value t_1 as $\text{Com}(t_1, \delta) = g^{t_1} h^\delta$.
   Commitment to scaler value t_2 as $\text{Com}(t_2, \rho) = g^{t_2} h^\rho$.
2) Calculate the following challenges $x, y, z \in \mathbb{Z}_p^*$, for non-interactive version by using the hash function:
   $y = \text{Hash}\left(\text{Com}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta)\text{Com}(\overrightarrow{Ls}, \overrightarrow{Rs}, \gamma)\right)$
   $z = \text{Hash}\left(\text{Com}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta), \text{Com}(\overrightarrow{Ls}, \overrightarrow{Rs}, \gamma), y\right)$
   $x = \text{Hash}\left(\text{Com}(t_1, \delta), \text{Com}(t_2, \rho)\right)$
3) Calculate the two vector polynomials $\vec{l}(X), \vec{r}(X) \in \mathbb{Z}_p^n$ as: $\vec{l}(X) = \overrightarrow{Lv} - z \cdot \vec{1}^n + \overrightarrow{Ls} \cdot X$
   $\vec{r}(X) = \vec{y}^n \circ \left(\overrightarrow{Rv} + z \cdot \vec{1}^n + \overrightarrow{Rs} \cdot X\right) + z^2 2^n$
4) Calculate the inner product $\hat{t}(X)$ of $\vec{l}(X)$ and $\vec{r}(X)$ as $\hat{t}(X) := \langle \vec{l}(X), \vec{r}(X) \rangle = \sum_{i=0}^{n} \sum_{j=0}^{i} \langle \vec{l_i}, \vec{r_j} \rangle \cdot X^{i+j}$

5) Calculate the random number μ and τ as
$\mu = \beta + \gamma \cdot x \qquad \tau = \rho \cdot x^2 + \delta \cdot x + z^2 \cdot \alpha.$
6) Commit to the vector polynomials $\vec{l}(X)$ and $\vec{r}(X)$ by using improved inner product argument as
$\text{Com}\left(\vec{l}(X), \vec{r}(X)\right) = \vec{g}^{\vec{l}(X)} \cdot \vec{h}^{\vec{r}(X)}$.
7) By using the improved inner product method, generate the proof of knowledge of secret vector polynomials $\vec{l}(X)$ and $\vec{r}(X)$ such that their inner product is equal to $\hat{t}(X)$.
8) Publish the parameters $\tau, \mu, \hat{t}, \text{Com}(v, \alpha), \text{Com}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta), \text{Com}(\overrightarrow{Ls}, \overrightarrow{Rs}, \gamma)$, $\text{Com}(t_1, \delta), \text{Com}(t_2, \rho), \text{Com}\left(\vec{l}(X), \vec{r}(X)\right)$ along with the proof of knowledge of two vector polynomials $\vec{l}(X)$ and $\vec{r}(X)$ by using improved inner product proofs.

Algorithm 3: Range-proof: Verifier Algorithm

1) Check if the challenges are calculated correctly by recalculating them as:
$x? = \text{Hash}\left(\text{Com}(t_1, \delta), \text{Com}(t_2, \rho)\right)$
$y? = \text{Hash}\left(\text{Com}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta)\text{Com}(\overrightarrow{Ls}, \overrightarrow{Rs}, \gamma)\right)$
$z? = \text{Hash}\left(\text{Com}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta)\text{Com}(\overrightarrow{Ls}, \overrightarrow{Rs}, \gamma), y\right)$
2) Compute the new generator vector $\vec{h}'$ by computing its component values as:
$h'_i = h_i^{y^{1-i}}$, for $i = 1, 2, \ldots, n$.
3) For $f(y, z) = (z - z^2) \cdot \langle \vec{1}^n, \vec{y}^n \rangle - z^3 \cdot \langle \vec{1}^n, \vec{2}^n \rangle$ and $t_0 = f(y, z) + z^2 \cdot v$,
check if $\hat{t}(x)$ is equal to $t_0 + t_1 x + t_2 x^2$ by checking the following:
$g^{\hat{t}(x)} \cdot h^{\tau}? = \text{COM}^{z^2}(v, \alpha) \cdot g^{f(y,z)} \cdot \text{COM}^{x}(t_1, \delta) \cdot \text{COM}^{x^2}(t_2, \rho)$
4) Check the commitments to vector polynomials $\vec{l}(X)$ and $\vec{r}(X)$ by recomputing them as
$\text{Com}_{\text{new}} = \text{COM}(v, \alpha) \cdot \text{COM}^{x}(\overrightarrow{Lv}, \overrightarrow{Rv}, \beta) \cdot \vec{g}^{-z} \cdot \vec{h}'^{z \cdot \vec{y}^n + z^2 \cdot \vec{2}^n}$.
5) Check if Com_{new} is equal to $h^{\mu} \cdot \vec{g}^{\vec{l}(x)} \cdot \vec{h}'^{\vec{r}(x)}$.
6) Check if the inner product is calculated correctly by checking $\hat{t}(X) = \langle \vec{l}(X), \vec{r}(X) \rangle$.
7) If all these are correct, then accept the proof, otherwise, reject.

D. Discussion

zkLedger is the first zero knowledge-based distributed ledger system. In most of the blockchain's ledger, transactions are stored in their plain-texts form while in zkLedger, they are stored in the form of commitments to the transaction, which is a secret amount. zkLedger uses a table construction: a transaction is a row that includes an entry for every participant, and columns represent participant's transfers.

1) No trusted setup required (unlike zk-SNARKs), so the participants of the zk-ledger can either be honest or malicious.
2) It uses the Schnorr-type non-interactive zero-knowledge proofs (NIZK) [15] which drastically reduces the communication cost between participants.
3) To hide the commitment values, zk-ledger uses Pedersen commitments that can be homomorphically combined.
4) It provides strong transaction privacy. Amount of transaction, sender and receiver information, transaction graph, or linkages between transactions are hidden. Only the time of transactions and the type of asset being transferred are public.
5) It provides completeness. Because of its columnar ledger construction, (hospitals) cannot hide any transactions from the verifier.
6) There is a set of proofs that everyone can publicly verify, and transactions with incorrect proofs will be ignored. These proofs guarantee that every participant has an audit token and is thus allowed to perform the audit.
7) All proofs and commitments are consistent.

VI. Conclusion

Any financial system contains high-level sensitive information, for example, salary amount, beneficiary account details, and personal details. In the era of the Internet of Things, everything is at the tipping point of IoT device users; therefore, privacy is very important. We have proposed a decentralized financial system suitable for the healthcare financial system which guarantees accountability and transparency, thereby mitigating fraud risks. To mitigate the privacy issue to hide the sensitive information, we use the non-interactive version of zero-knowledge proofs by using the Fiat-Shamir heuristic, reducing the communication cost between the IoT devices. Our system uses several zero-knowledge proofs such as range-proof, proof of asset, proof of balance, range-proof, proof of consistency, etc. to ensure that the transactions are correct and there are no double-spending. Moreover, our system framework is capable of answering the basic audit queries such as sum, averages, ratio, mean, variance, skewness that is mainly needed for the audit process. Since hospitals are the backbone to achieve the UN sustainable goals, therefore, for better understanding, we explain our system framework for the healthcare financial system at a micro-level. However, our

system can easily be extended to a broader level which is our plan.

FUNDING

This research is part of the Youth-Community for Cyber-Skills project supported by Industriens Fond (The Danish Industry Foundation).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Zcash. URL https://z.cash/

[2] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., Caro, A.D., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolic, M., Cocco, S.W., Yellick, J.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: R. Oliveira, P. Felber, Y.C. Hu (eds.) Proceedings of the Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal, April 23-26, 2018, pp. 30:1–30:15. ACM (2018). DOI 10.1145/3190508.3190538. URL https://doi.org/10.1145/3190508.3190538

[3] Androulaki, E., Camenisch, J., Caro, A.D., Dubovitskaya, M., Elkhiyaoui, K., Tackmann, B.: Privacy-preserving auditable token payments in a permissioned blockchain system. In: AFT '20: 2nd ACM Conference on Advances in Financial Technologies, New York, NY, USA, October 21-23, 2020, pp. 255–267. ACM (2020). DOI 10.1145/3419614.3423259. URL https://doi.org/10.1145/3419614.3423259

[4] Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable, transparent, and post-quantum secure computational integrity. IACR Cryptol. ePrint Arch. **2018**, 46 (2018)

[5] Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Succinct non-interactive zero knowledge for a von neumann architecture. In: 23rd {USENIX} Security Symposium ({USENIX} Security 14), pp. 781–796 (2014)

[6] Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 315–334. IEEE (2018)

[7] Dwivedi, A.D., Malina, L., Dzurenda, P., Srivastava, G.: Optimized blockchain model for internet of things based healthcare applications. In: N. Herencsar (ed.) 42nd International Conference on Telecommunications and Signal Processing, TSP 2019, Budapest, Hungary, July 1-3, 2019, pp. 135–139. IEEE (2019). DOI 10.1109/TSP.2019.8769060. URL https://doi.org/10.1109/TSP.2019.8769060

[8] Lin, C., He, D., Huang, X., Khan, M.K., Choo, K.K.R.: Dcap: A secure and efficient decentralized conditional anonymous payment system based on blockchain. IEEE Transactions on Information Forensics and Security **15**, 2440–2452 (2020). DOI 10.1109/TIFS.2020.2969565

[9] Narula, N., Vasquez, W., Virza, M.: zkledger: Privacy-preserving auditing for distributed ledgers. In: 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), pp. 65–80. USENIX Association, Renton, WA (2018). URL https://www.usenix.org/conference/nsdi18/presentation/narula

[10] Narula, N., Vasquez, W., Virza, M.: zkledger: Privacy-preserving auditing for distributed ledgers. In: 15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18), pp. 65–80 (2018)

[11] Partala, J., Nguyen, T.H., Pirttikangas, S.: Non-interactive zero-knowledge for blockchain: A survey. IEEE Access **8**, 227945–227961 (2020). DOI 10.1109/ACCESS.2020.3046025

[12] Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Annual international cryptology conference, pp. 129–140. Springer (1991)

[13] Poelstra, A.: Mimblewimble. White paper (2016)

[14] Poelstra, A., Back, A., Friedenbach, M., Maxwell, G., Wuille, P.: Confidential assets. In: International Conference on Financial Cryptography and Data Security, pp. 43–63. Springer (2018)

[15] Schnorr, C.P.: Efficient signature generation by smart cards. Journal of cryptology **4**(3), 161–174 (1991)

[16] Singh, R., Dwivedi, A.D., Srivastava, G.: Internet of things based blockchain for temperature monitoring and counterfeit pharmaceutical prevention. Sensors **20**(14), 3951 (2020). DOI 10.3390/s20143951. URL https://doi.org/10.3390/s20143951

[17] Srivastava, G., Dwivedi, A.D., Singh, R.: PHANTOM protocol as the new crypto-democracy. In: K. Saeed, W. Homenda (eds.) Computer Information Systems and Industrial Management - 17th International Conference, CISIM 2018, Olomouc, Czech Republic, September 27-29, 2018, Proceedings, *Lecture Notes in Computer Science*, vol. 11127, pp. 499–509. Springer (2018). DOI 10.1007/978-3-319-99954-8_41. URL https://doi.org/10.1007/978-3-319-99954-8_41

[18] Tomaz, A.E.B., Nascimento, J.C.D., Hafid, A.S., De Souza, J.N.: Preserving privacy in mobile health systems using non-interactive zero-knowledge proof and blockchain. IEEE Access **8**, 204441–204458 (2020). DOI 10.1109/ACCESS.2020.3036811