UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Blockchain-based data governance for privacy-preserving in multi-stakeholder settings

Rodrigo Dutra Garcia

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Rodrigo Dutra Garcia

Blockchain-based data governance for privacy-preserving in multi-stakeholder settings

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Jó Ueyama

USP – São Carlos
August 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

```
        Garcia, Rodrigo Dutra
G216b      Blockchain-based data governance for privacy-
        preserving in multi-stakeholder settings / Rodrigo
        Dutra   Garcia; orientador Jó Ueyama. -- São Carlos,
        2023.
           64 p.


           Dissertação (Mestrado - Programa de Pós-Graduação
        em Ciências de Computação e Matemática
        Computacional) -- Instituto de Ciências Matemáticas
        e de Computação, Universidade de São Paulo, 2023.


           1. Privacidade. 2. Blockchain. 3. Consentimento.
        4. Descentralização. 5. Governança. I. Ueyama, Jó,
        orient. II. Título.
```

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Rodrigo Dutra Garcia

Governança de dados baseada em blockchain com preservação da privacidade em configurações com múltiplas partes interessadas

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Jó Ueyama

USP – São Carlos
Agosto de 2023

ACKNOWLEDGEMENTS

Firstly, I express my deepest gratitude to my family for their support throughout the entire process of completing this research. Their encouragement and belief in me were integral to the realization of this work.

I would also like to thank my supervisor, Professor Dr. Jó Ueyama, for his patience and guidance throughout the various stages of this research. His expertise and direction were valuable to me. Additionally, I am grateful to Dr. Gowri Ramachandran for his suggestions and contributions to improving this research.

I am also thankful to National Council for Scientific and Technological Development (CNPq) for providing financial support during this research (**Grant: 133470/2020-2**).

RESUMO

GARCIA, R. D. **Governança de dados baseada em blockchain com preservação da privacidade em configurações com múltiplas partes interessadas**. 2023. 64 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Em sistemas envolvendo múltiplas partes interessadas, como o setor da saúde, internet das coisas e o gerenciamento da cadeia de suprimentos, há uma geração, troca e compartilhamento frequente de dados. Como resultado, os proprietários dos dados geralmente precisam controlar e preservar a privacidade de suas informações, enquanto os consumidores dos dados exigem métodos para determinar as origens e os criadores dos registros. Esses conflitos exigem soluções de governança que garantam a proveniência dos dados, proteção da privacidade, gestão de consentimento e compartilhamento seletivo. Para responder a esses desafios, esta pesquisa apresentou um sistema descentralizado de governança de dados que utiliza a tecnologia blockchain, re-criptografia por proxy (PRE) e assinaturas de Boneh, Boyen e Shacham (BBS). A abordagem proposta permite que os proprietários dos dados controlem, compartilhem seletivamente e rastreiem seus dados, mantendo a privacidade dos registros. Além disso, o sistema proposto permite que os consumidores dos dados compreendam a linhagem dos registros por meio de um mecanismo de proveniência baseado em blockchain. Como estudo de caso, a pesquisa examinou e avaliou prescrições médicas eletrônicas envolvendo dados sensíveis e múltiplas partes interessadas, incluindo pacientes como proprietários dos dados, médicos e farmácias como consumidores dos dados. A pesquisa foi estruturada como uma coletânea de artigos organizados na seguinte ordem: formulação do problema e desenvolvimento de contratos inteligentes, implementação do gerenciamento de privacidade e consentimento por meio de re-criptografia por proxy e aplicação de assinaturas de Boneh, Boyen e Shacham para compartilhamento seletivo de dados. As avaliações de prova de conceito e implementação, utilizando CosmWasm, Hyperledger Besu, Ethereum, pyUmbral PRE e BBS, mostram que o sistema descentralizado proposto é independente de plataforma, escalável e capaz de fornecer uma maior transparência, privacidade e confiança com uma sobrecarga mínima.

Palavras-chave: Governança de Dados, Descentralização, Prescrição Eletrônica, Privacidade, Blockchain, Contratos Inteligentes, Re-criptografia por Proxy, Compartilhamento Seletivo.

ABSTRACT

GARCIA, R. D. **Blockchain-based data governance for privacy-preserving in multi-stakeholder settings**. 2023. 64 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

In multi-stakeholder systems, such as healthcare, the internet of things, and supply chain management, there is frequent data generation, exchange, and sharing. As a result, data owners often desire control over their data and maintain privacy, while data consumers require methods to ascertain the origins and creators of the data. These conflicts of interest require developing data governance systems that guarantee data provenance, privacy protection, consent management, and selective disclosure. This research proposed a decentralized data governance system utilizing blockchain technology, proxy re-encryption (PRE), and Boneh, Boyen, and Shacham (BBS) signatures to address these challenges. The proposed system enables data owners to control, selectively share, and track their data through privacy-enhancing, consent management, and selective disclosure mechanisms while also allowing data consumers to understand the lineage of the data through a blockchain-based provenance mechanism. As a case study, the research examined and evaluated electronic prescriptions involving sensitive data and multiple stakeholders, including patients as data owners and doctors and pharmacists as data consumers. The research was structured as a collection of articles organized in the following sequence: problem formulation and developing smart contracts, implementing privacy and consent management through PRE, and applying BBS signatures for selective data sharing. The proof-of-concept implementation and evaluations, conducted using CosmWasm, Hyperledger Besu, Ethereum, pyUmbral PRE, and BBS signatures, demonstrate that the proposed decentralized system is platform-agnostic, scalable, and capable of providing a higher level of transparency, privacy, and trust with minimal overhead.

Keywords: Data Governance, Decentralized, E-prescription, Privacy, Blockchain, Smart Contracts, Proxy Re-encryption, Selective Sharing.

CONTENTS

CHAPTER 1

INTRODUCTION

The expansion of digital technologies has led to exponential data production and sharing among stakeholders. Applications such as healthcare and the internet of things are examples of this trend, as they rely heavily on collecting and disseminating data (MUKTA *et al.*, 2022; UDDIN *et al.*, 2021). However, as the scale of data sharing expands, it is crucial to ensure that the privacy rights of individuals are safeguarded. In multi-stakeholder applications, data consumers need to clearly understand the lineage of the data they utilize, including knowledge of the services and companies involved in collecting, storing, and disseminating the data. Furthermore, data owners need to consent to share specific information with data consumers and have the control to grant or revoke access to personal information and sensitive data, allowing greater control and transparency in handling personal information (KAKARLAPUDI; MAHMOUD, 2021).

In the healthcare industry, sensitive data such as electronic medical records (EMRs) are routinely produced and shared among various stakeholders, including patients, doctors, hospitals, and pharmacies. For instance, EMRs contain personally identifiable information (PII), diagnosis, and medication. This information is required for providing quality care. However, handling sensitive data requires robust data protection measures to ensure the privacy and security of individuals. The centralization of data storage in current healthcare systems, including electronic prescription systems, imposes a significant challenge to protecting EMRs and the privacy of individuals whose information is being collected and shared. The centralized structure of these systems creates a single point of failure, making them vulnerable to breaches and unauthorized access (WAZID *et al.*, 2022; KSIBI; JAIDI; BOUHOULA, 2022). Additionally, the lack of transparency inherent in such architecture can compromise the ability of patients to exert control and oversight over their personal information, raising concerns about potential violations of privacy rights (QAHTAN *et al.*, 2023).

The trust mechanism in a centralized architecture is typically based on a central authority to enable controlling and managing data access. In contrast, a decentralized architecture, such as blockchain, uses distributed ledger technology to record and share information in a tamper-proof manner without needing a central entity (NAKAMOTO, 2009). Blockchain technology uses a peer-to-peer (P2P) network to establish trust through a consensus mechanism among participating nodes rather than relying on a central authority, providing security and transparency. The transparency inherent in blockchain technology enables all participants in the network to have a clear view of the data and its history, making it easier to trace the lineage of the data and understand how it has been shared and processed. Furthermore, the decentralized structure of the network makes it more resilient to attacks and failures. In addition to its decentralized structure, blockchain technology also enables smart contracts. This feature was initially proposed by Szabo (SZABO, 1997) as protocols, and in blockchain platforms such as Ethereum, acts as immutable self-executing programs written in code and stored on the blockchain (BUTERIN *et al.*, 2014). It enables tasks automation and agreements without needing a third party as an intermediary, which increases efficiency, security, and trust in the execution of the contract while also reducing costs (HEWA; YLIANTTILA; LIYANAGE, 2021).

Despite capabilities, one of the main limitations of using blockchain technology in sensitive applications such as healthcare and the internet of things is the issue of data privacy (PENG *et al.*, 2021). The transparency of blockchain technology means that all data stored on the blockchain is available to all network nodes. It implies a significant challenge to maintaining the confidentiality of sensitive healthcare information, such as medical records and personal information, which must be protected in compliance with regulations such as the General Data Protection Regulation (GDPR) (QI *et al.*, 2022). Researchers have proposed some approaches to address the issue of sensitive data privacy in healthcare applications, such as off-chain data storage, encryption, and zero-knowledge proofs (ZKP) (WANG; ZHAO; WANG, 2020; QI *et al.*, 2022). However, there is a lack of study on enabling data owners to manage and selectively share attributes with stakeholders while preserving sensitive data privacy, particularly in the healthcare sector, such as electronic prescription. The objective of the research aims to propose a blockchain-based system that answers the following research questions (RQ):

- **RQ1:** how can blockchain and smart contracts secure and manage sensitive data in a tamper-proof ledger? In particular, how can smart contracts be implemented for electronic prescription use cases using byzantine fault tolerance (BFT) platforms such as Tendermint?

- **RQ2:** how can data owners, such as patients, maintain their privacy while still tracking and governing the usage by other parties?

- **RQ3:** while maintaining data owners' privacy, how can a regulatory entity access data for accountability and compliance verification in a decentralized data governance system?

- **RQ4:** how can data owners selectively share specific attributes with certain stakeholders in a reliable and scalable manner?

Particularly, this study employed proxy re-encryption (PRE) to ensure the privacy of sensitive data and enable data sharing with owner consent. Additionally, Boneh, Boyen, and Shachum (BBS) signatures built with zero-knowledge proof were utilized to allow the selective sharing of data in a blockchain-based system. The research focused on the electronic prescription (e-prescription) use case, a multi-stakeholder application with sensitive data sharing (VEJDANI *et al.*, 2022; ALDUGHAYFIQ; SAMPALLI, 2021). Patients act as data owners and can selectively share their data with relevant stakeholders, such as pharmacies and doctors while keeping the data securely encrypted and stored on the blockchain.

1.1 Thesis Organization and Contributions

This dissertation was structured as a collection of articles arranged according to contribution. Figure 1 presents the dissertation organization with the research questions and the techniques utilized. Chapter 2 and 3 are part of an incremental study focusing on answering RQ1 by presenting the implementation and evaluation of smart contracts. Chapter 4 aims to address RQ2 by presenting a blockchain-based system with privacy protection using PRE. In addition, Chapter 5 focuses on answering RQ3 and RQ4 by allowing a regulatory entity to access data for accountability and enabling data owners to disclose attributes using BBS signatures selectively.

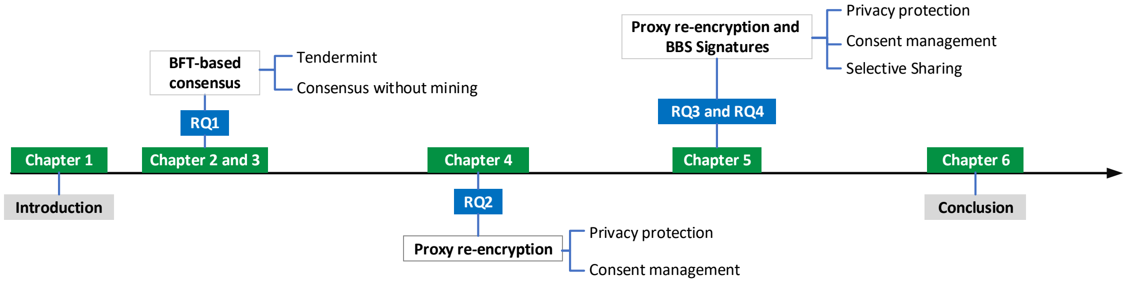

Figure 1 – Representation of the dissertation structure, highlighting the research questions and the techniques and platforms employed within the chapters

The first article, titled *“Towards a Decentralized e-Prescription System Using Smart Contracts”* and presented in Chapter 2, introduces an electronic prescription system that leverages smart contracts on BFT platforms. The main contribution of this research is the design and implementation of a blockchain-based e-prescription system using smart contracts on a BFT-based consensus mechanism. In particular, the work used Tendermint consensus, which is not widely adopted for smart contract applications. The study compares the performance of the proposed system with another BFT platform, Hyperledger Fabric, evaluating contract file size, transaction overhead, scalability, and smart contract deployment complexity. *However, the study does not evaluate and compare the cost with another existing consensus mechanism, such as Proof of Work (PoW). Furthermore, the work does not address the issue of data privacy in transactions, and the mechanisms used to protect it.*

Chapter 3 presents the second article, titled *"Exploiting smart contracts in PBFT-based blockchains: A case study in medical prescription system"* which builds upon the previous work by providing a more comprehensive evaluation and discussion. The contributions are to evaluate the implementation of smart contracts on BFT blockchain platforms such as Tendermint and Hyperledger Besu and compare their operational cost and performance to Ethereum, a PoW blockchain. *However, the study only briefly discusses the use of public key encryption to preserve patient privacy in the blockchain-based e-prescription system and does not evaluate its effectiveness.*

The third work, entitled *"A Blockchain-based Data Governance with Privacy and Provenance: a case study for e-Prescription"* presented in Chapter 4, implements the use of proxy re-encryption to ensure patient consent and privacy. The main contribution of the research is the development of a system that enables data owners to control and monitor their data through privacy-enhancing and consent management mechanisms while also allowing data consumers to trace the origins of the data through a blockchain-based provenance system. *However, the study does not implement a selective disclosure mechanism to allow data owners to share specific information with selected stakeholders.*

As an extension of prior work, the fourth article, entitled *"Blockchain-aided and Privacy-preserving Data Governance in Multi-stakeholder Applications"* which is presented in Chapter 5, includes a regulator authority and incorporates the use of Boneh, Boyen, and Shacham (BBS) signatures to enable selective sharing by data owners. The research presents the following contributions: a decentralized architecture for multi-stakeholder applications which combines blockchain, smart contracts, and proxy re-encryption mechanism that allows for data owner consent while keeping data encrypted on the blockchain, enabling regulator entity to track the records with data owner permission; the use of BBS signatures to enable data owners to share specific attributes with data consumers; and a proof-of-concept performance evaluation using BFT blockchain platforms (CosmWasm and Hyperledger Besu) compared to Ethereum PoW. Additionally, the study employs NuCypher's pyUmbral proxy re-encryption (PRE) library and MATTR JSON-LD library using BLS12-381 key pairs for BBS signature evaluation.

The study's conclusion, limitations, and future research are discussed in Chapter 6. All evaluation software and smart contracts developed as part of this research can be accessed on GitHub[1].

[1] <https://github.com/rodrigodg1/e-prescription>

CHAPTER

2

TOWARDS A DECENTRALIZED E-PRESCRIPTION SYSTEM USING SMART CONTRACTS

This chapter presents the article published in the International Symposium on Computer-Based Medical Systems (CBMS) under the following IEEE permission:

Contribution Statement: Software, Data curation, Writing – review & editing.

[1] <https://doi.org/10.1109/CBMS52027.2021.00037>

Towards a decentralized e-prescription system using smart contracts

Rodrigo Dutra Garcia,
Gabriel Augusto Zutião
Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, Brazil
{rgarcia,gabriel.zutiao}@usp.br

Gowri Ramachandran
USC Viterbi School of Engineering
University of Southern California
CA, USA
gsramach@usc.edu

Jo Ueyama
Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, Brazil
joueyama@icmc.usp.br

***Abstract*—Electronic prescription (e-Prescription) is a digital way to manage medical prescriptions and reduce inconsistencies in the communication between doctors, patients, and pharmacies. Smart contracts allow the automation of tasks and business rules in a decentralized architecture (i.e., without the need for an intermediary or central authority). Platforms such as Ethereum and Hyperledger Fabric support smart contracts development through a consensus mechanism such as Proof-of-Work or another criterion among the network's participating nodes. This paper explores Tendermint, a Byzantine Fault Tolerant (BFT) based consensus mechanism that has not yet been widely adopted for smart contracts platforms. We apply our devised model to the healthcare application domain, more precisely in the field of e-prescription, and our results demonstrate that smart contracts can be implemented on a BFT-based platform. To the best of our knowledge, this is the first study investigating the implementation of smart contracts in a BFT-based platform such as Tendermint. We exploit this domain as there can exist some conflicting interests of profit-taking. For example, pharmacists can increase the medication dosage above the one prescribed by doctors for profit-taking. Such a scenario can occur particularly in countries where healthcare is free of charge and offered as a public service (e.g., Brazil). We show that smart contracts in BFT-based blockchain can help in solving problems in these application scenarios. Finally, our two key contributions in this paper are two-fold: (*i*) exploit smart-contracts in BFT-based platforms where they (smart contracts) are not very established yet in the blockchain domain; (*ii*) provide a smart-contract-based e-prescription solution to reduce the costs (no need for a central authority) and scams particularly in countries where the medical service is free and public.**

***Index Terms*—Smart Contracts, e-Prescription, Tendermint, BFT**

I. Introduction

Electronic prescriptions are an efficient way to communicate and manage prescriptions between doctors, patients, and pharmacies. Medical records and recommended medications are stored digitally, allowing more effective communication between stakeholders. However, existing systems use a centralized architecture, a single point of failure, and control, allowing a central authority to manage and change sensitive records such as medical prescriptions enabling patients to receive medications without a doctor's prescription. Note that a few medicines can only be purchased from pharmacies with doctor's prescription. However, some pharmacies illegally sell medication to patients without a valid doctor's prescription, which leads to drug abuse. In some cases, it may cause serious health issues [1], [2]. To overcome such a problem, we propose a decentralized e-prescription system using blockchain technology and smart contracts.

Blockchain is a technology that has a set of features such as hard-to-change records and encryption using a fault-tolerant and decentralized architecture. Records are stored through blocks and added in time-sequential order based on a consensus among network participants. This technology has been adopted by several applications, including finance sectors, for performing tasks efficiently with cost reduction [3].

Smart contracts are programs that contain business logic and are executed within the blockchain. These programs make it possible to automate tasks, without the need for an intermediary. In e-prescription, smart contracts are a safe and efficient way to manage medical records and medicines sales. With this technology, it is possible to create rules to validate prescriptions in a decentralized and fault-tolerant architecture. Smart contracts have been used in social applications such as the Internet of Things [4], Finance [5], and several other applications [6], [7].

In Ethereum, smart contracts can be developed using a set of languages like Solidity [6]. These programs are compiled as byte code and run on a virtual machine. Although Ethereum is a widely used smart contract platform, the transaction cost is one of its limitations, making it less attractive for cost-sensitive applications. Hyperledger Fabric follows a different structure than the common one present on other Distributed Ledger Technologies. In Fabric, the order that each transaction follows is execute-order-validate. Besides, it follows a private and consortium deployment model [8]. Fabric supports smart contracts developed in Go, JavaScript, TypeScript, and Java.

Tendermint, a BFT-based blockchain platform, introduces a solution for the blockchain consensus without the high cost of mining and allows flexibility for developing applications in different existing high-level languages. This solution was built from the adaptation of a solution to the Byzantine Generals Problem [9], [10]. One of the goals of Tendermint is to separate the application from the consensus mechanism. The application is executed in a separate process from Tendermint, and communication is carried out through an interface called ABCI (Application Blockchain Interface). These features allow flexibility for developing applications with the replicated state machine in different existing high-level languages. Tendermint can thwart byzantine failures, but its ability to execute smart contracts is not explored in literature, which is one of the aims of this work.

In this paper, we explore the use of smart contracts in Tendermint, which is a Practical Byzantine Fault Tolerance (PBFT) [11] based blockchain platform, for a decentralized e-prescription application. For the experiments, we use CosmWasm, a secure multi-chain smart contracts platform based on Tendermint and compares its performance with a Hyperledger Fabric implementation, a modular decentralized ledger technology (DLT) platform. By such a comparison, we want to show the feasibility of executing smart contracts over PBFT-based platforms such as Tendermint. The software, including smart contracts, used for the evaluation is available online [12]. To the best of our knowledge, this is the first study investigating the implementation of smart contracts in a BFT-based platform such as Tendermint.

II. Motivation and Related Work

In many countries, including Brazil, healthcare service is public and free of charge for the entire population. This has a downside of bringing problems, such as scams to acquire particularly costly medications. One of the existing problems is the adulteration of medical prescriptions that lead to medication trades in the black market [1]. It is noteworthy that the centralized Electronic Health Record (EHR) systems are typically managed and controlled by a single stakeholder. Such systems are susceptible to central points of failure [13], [14] and interest conflicts (e.g. the pharmacist may take profit while the country may suffer losses). A number of blockchain-based solutions for managing medical and medication data have been proposed so far. Thatcher and Acharya explored the blockchain's immutability feature in digital prescription systems and proposed an application called RxBlock [15]. Azaria et al. [16] introduced MedRec, a solution using Ethereum's private network for managing medical records, where the smart contracts are established between patient and a provider. Similarly, Ribeiro and Vasconcelos [17], proposed MedBlock, a solution using Hyperledger Fabric to save patient information and consultations using smart contracts. Xia et al. [18] proposed a solution for sharing medical data called MeDShare. Ying et al. [19] suggested an architecture for the supply of medicines to prevent illegal actions by an agent in confidential transactions.

Tanwar et al. [20] explored several solutions using blockchain for the current model of healthcare systems. The authors proposed a model for registering EHRs on a unified ledger to use administrators, patients, clinicians, and laboratories. This solution does not involve the pharmacy operations and is more focused on forming a service between a laboratory and a clinician, which is complementary to the present work. Wu and Tsai [21] presented an architecture for blockchain systems where the hospitals' nodes request medical record data between them to gather the information necessary on the case of an appointment to help the doctor make a diagnosis and complete prescriptions.

The key problem with the above mentioned efforts is that none of them explores a lightweight and cost-efficient smart contract solutions based on PBFT and to the best of our knowledge none of the implemented solutions claim or prove to be blockchain-agnostic. In this paper, we explore Tendermint BFT as a smart contract platform targeting resource-constrained and cost-sensitive e-Prescription systems and compare it with an implementation in Hyperledger Fabric. We compare our solution with Fabric as most of the existing works have proof of concept based on Fabric.

III. Background

A. Smart Contracts

The concept of smart contract technology was initially suggested and presented by Nick Szabo [22] in the 1990s, and currently has potential applications for a variety of use cases and application domains [6], [7]. Smart contracts are programs that allow the business logic to be encapsulated as a program in the blockchain and executed when some pre-established condition between the parties is reached. First, the contract is implemented by the developer with the business logic of the application using a programming language accepted by the platform. Then, the contract is compiled and published on the blockchain [7]. In most cases, the address is necessary to send transactions for the contract method. These transactions will be processed, and the results will be compared using the criteria of the consensus protocol.

B. Tendermint Consensus

Tendermint is an open-source project that focuses on the consensus mechanism. It allows the development of replicated and Byzantine fault-tolerant systems - systems that do not have a central point of failure or a central point of control [23]. Communication between the application and Tendermint is carried out through the ABCI interface that integrates the blockchain with the application logic. In this regard, the application can be built in any high-level language and then combined with Tendermint.

Tendermint uses the PBFT protocol for the Byzantine fault-tolerant replicated state machine [9]–[11]. From the expression $N \geq 3m + 1$, it is possible to determine the minimum number of active nodes N to keep the system fault-tolerant. Here, m is the maximum number of simultaneous faults.

Each node validator will execute the smart contract to process and validate each transaction, and the results of the validation will be stored in that node's mempool. If the transaction is valid, it will be transmitted to another peer validator to be validated, and the process is repeated for the other peers. Upon reaching the block time, the proposer node will build the block with all the transactions stored in the interval with the respective hashes values of those transactions [23].

From the block created by the proposer, each validator node will process the block and send the result to the adjacent peers. First, if 2/3 of the network approves the block, a *pre-vote* message will be sent between the nodes, and again if 2/3 of the nodes confirm the block, the *pre-commit* message will be sent between the nodes and the new block will be stored on the blockchain. Figure 1 shows the steps in the consensus mechanism in Tendermint.

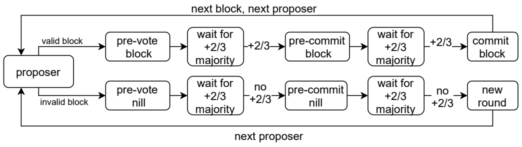

Figure 1. Tendermint consensus mechanism

C. Consensus in Hyperledger Fabric

The consensus in Hyperledger Fabric is pluggable, that is, it isn't hard-coded on the implementation, so it can be implemented differently for each case. The endorsement part of the consensus consists of the broadcast of a proposal of transaction to other nodes, which will be the endorsers. Each one of them executes locally the proposed transaction and sends both the data that were used as input and the output data to the client, which will send the collection of the endorsements to the ordering service [8].

IV. A Decentralized e-Prescription System

A. Model Architecture

We propose a model for the registration and management of medical prescriptions and medicines sales using smart contracts. In this scenario, we have the doctor who performs the diagnosis and prescribes the medications to the patient through the application of the doctor/clinic or hospital. The pharmacist analyzes the prescription and sells the medicines to the patient through the application of the pharmacy. The representation of the scenario can be seen in Figure 2.

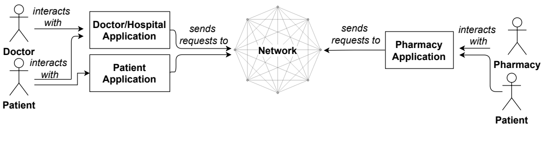

Figure 2. Decentralized architecture with stakeholders

The pharmacy application will only allow the pharmacist to finalize the medication's sale in the patient's medical prescription by checking the smart contract. For each sale made to the patient, it will be recorded in the contract's transaction history with the pharmacy and will be consulted later by the medical application.

Prescriptions and medicines sales are transactions sent to the network and are validated by the contract. Each patient has a contract with the doctor and the pharmacy. The prescriptions and sales of medications are transactions in these contracts. Thus, both the doctor and the pharmacist will be able to consult the patient's prescription and medicines' sales history through the contract address. Therefore, prescriptions and sales will only be valid on the network if it meets all the conditions established in the contract. Hence, counterfeiting, alteration of records, and sales without prescriptions will not be valid.

In the model, the doctor, through the application, can create an instance of the contract, i.e., assign the two parts of the contract (doctor and patient) through the addresses. Then, create a prescription specifying the diagnosis and medications and consult the patient's prescription history through the instantiated contract address. The patient will inform his address so that contracts, prescriptions, and sales can be carried out.

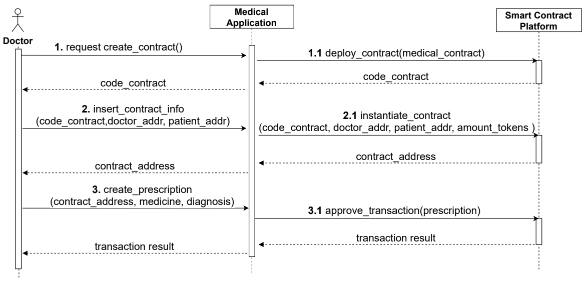
Doctor
Medical Application
Smart Contract Platform
1. request create_contract()
1.1 deploy_contract(medical_contract)
code_contract
code_contract
2. insert_contract_info (code_contract,doctor_addr, patient_addr)
2.1 instantiate_contract (code_contract, doctor_addr, patient_addr, amount_tokens)
contract_address
contract_address
3. create_prescription (contract_address, medicine, diagnosis)
3.1 approve_transaction(prescription)
transaction result
transaction result

Figure 3. Sequence diagram for medical contract

The pharmacist creates an instance of the sales contract with the patient and concludes the drug's sale. Accessing the functionality of selling medication includes consulting the contract's status and checking the medication prescribed by the doctor and the latest sales made to the patient, avoiding multiple transactions. These features are represented in the use case diagram in Figure 4 and the sequence diagram of the events in Figure 3.

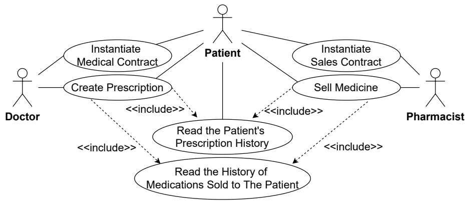
Patient
Instantiate Medical Contract
Instantiate Sales Contract
Doctor
Create Prescription
Sell Medicine
Pharmacist
<<include>>
<<include>>
Read the Patient's Prescription History
<<include>>
<<include>>
Read the History of Medications Sold to The Patient

Figure 4. Use case diagram with stakeholders

A note on Privacy: Privacy of the patients can be respected in our solution by leveraging a permissioned blockchain in combination with encryption schemes, wherein the patient and the doctor can authorize the pharmacy to decrypt the encrypted data stored on the blockchain. As Feng et al. [24] and Huynh et al. [25] explored privacy in blockchain technology, our work can comply with privacy requirements having identity privacy (with the use of the contract addresses instead of the patient identity) and transaction privacy (also possible with the use of the contract address and cryptography, which makes it difficult to analyse a single transaction, similar to the work done by Oliveira et al. [26], which used a public key infrastructure to ensure authenticity), however data confidentiality is a open research question and future expansions on it should be considered. We plan to extend our decentralized e-prescription system with privacy-preserving techniques in our future work.

V. Implementation Overview

We implement the e-Prescription system presented in Section IV using Tendermint and Hyperledger Fabric. To the best of our knowledge, this is the first study that implements smart contracts over PBFT-based platforms such as Tendermint and compares its performance with Hyperledger Fabric.

A. Smart Contracts over PBFT

We use CosmWasm as the PBFT-based platform, which uses Tendermint as the consensus layer while providing smart contracts support. In the CosmWasm platform, the contract was implemented in the Rust programming language to address who can send the transaction to the network (doctor or pharmacist) and who can receive it (patient). Each contract has an initial amount of tokens sent to the patient to represent a completed transaction. This contract model required 165 lines of code, excluding external and test modules.

To store these transactions, Tendermint uses LevelDB and the contract status, as the Cosmos SDK modules are stored in a variant and persistent key-value structure of the AVL trees. For transaction costs, CosmWasm at the current stage uses a symbolic and configurable token for transaction fees. Tendermint as the consensus platform, allows configurations of these parameters.

The transaction tests were carried out on a local network with a single validator node, and in a second step, we used a testnet with eight and twelve active validators from around the world. For all transaction tests, shell scripts were implemented to determine the transaction's validation time on the network.

B. Hyperledger Fabric and Smart Contracts

The Hyperledger Fabric local test network implementation consists of a network with four nodes, an orderer node considering only one organization. The implementation would involve multiple organizations for each hospital and Pharmacy with a separate ordering service in a real-world scenario. The transactions are sent to the network using a system implementing the Hyperledger Fabric's Node SDK, accessible, for example, through an API.

Two smart contracts were used, one for the Doctor/Patient registers and the other for the Pharmacy/Patient registers. Both of them were written using the Go programming language and implemented two functions: one used to send a new register to the ledger and one to query by patient address. The Doctor and Pharmacy smart contract files developed are 164 and 157 lines long respectively.

For benchmarking the transaction time we implemented a simple script that uses Hyperledger Fabric's Node SDK to measure the time of a given quantity of transactions and outputs each transaction time and the overall time in the terminal and then compared the results with the ones obtained using Hyperledger Caliper, a well stablished benchmarking tool.

Table I
Contract information for each platform

Platform	Prog. Language	File Size (for upload)	Nº Lines of Code	
			Doctor	Pharmacy
Fabric	Golang	780B	164	157
CosmWasm	Rust	175kB	165	165

VI. Evaluation: Comparing Hyperledger Fabric Smart Contracts with Smart Contracts in Tendermint BFT

This section presents the evaluation of our prototype with the intent to show that smart contract in Tendermint is viable. It starts off by showing a comparison between the contract implementation carried out in Tendermint and Fabric. Note that we did not consider the Ethereum platform for our test implementations due to the costs associated with deploying contracts and processing transactions, but we uphold further extension in implementing our solution using the Ethereum platform.

A. Size of Smart Contracts

Table I compares the contract file size and the number of lines of code for the CosmWasm and Hyperledger Fabric platform. As shown on the table, each contract's size on the Hyperledger Fabric implementation is approximately 780 bytes (including composite key and the register) and 175 kilobytes on the CosmWasm. Still, sizes can vary based on the business rule encapsulated in the contract, which can change mostly related to the prescription/sale data. Both implementations achieve a small size for each contract, which means that storage wouldn't be a limiting factor in a large scale case. This is due to the simplicity of the data that the contracts deal, since the user don't need to provide much information to have a contract associated to him, just his key. It is important to note that this test used CouchDB with Fabric, different results could be obtained using a LevelDB database, but we chose to use CouchDB because it's a fully-fledged external database.

B. Transaction Overhead

Each implementation's average transaction time (validation and inclusion in a block) is shown in Table III, and each transaction's time between the platforms in Figure 6. To observe the data's dispersion, we calculate the amplitude and standard deviation of the transaction times. The results are summarized in Table II for a local network with a single validator and Table IV for the CosmWasm platform with one, eight, and twelve validators. In contrast, the Hyperledger Fabric implementation evaluation could only be evaluated locally due to the absence of a testnet. Still, as shown by

Table II
AMPLITUDE AND STD. DEVIATION OF TRANSACTIONS ON A LOCAL NETWORK WITH A SINGLE VALIDATOR (IN SECONDS)

Platform	Max. Time (s)	Min. Time (s)	Amplitude (s)	Std. Deviation (s)
Fabric	3.056	2.072	0.984	0.099
CosmWasm	4.919	1.995	2.924	0.291

Table III
AVERAGE TRANSACTION TIME BETWEEN EACH PLATFORM (IN SECONDS)

Platform	Total Transactions	Nº Validators		
		Single (Local) (s)	8 (Testnet) (s)	12 (Testnet) (s)
Fabric	100	2.094	-	-
CosmWasm	100	2.099	8.574	8.940

Shalab et al. [27], the implementations with smaller block sizes can be more performant.

For the Fabric platform on a local network with a single validator measured using our benchmarking tool the amplitude was 0.984s and the standard deviation was 0.099s. For CosmWasm, the amplitude was 2.924s and the standard deviation was 0.291s. The average time for each transaction on a local network with one validator is 2.094s on the Fabric implementation and 2.099s on CosmWasm implementation. As the difference is on the milliseconds' field, we can safely assume that both implementations have a relatively equal transaction time on the local network scenario. In the case of a testnet with eight validators, the CosmWasm transaction requires 8.574s, and with twelve validators, 8.940s.

The means measured with Hyperledger Caliper for a local network with 5 peers and a single orderer are the following: send rate of 174.52 transactions per second (TPS), maximum and minimum latency of 2.767s and 0.116s, average latency of 1.005 seconds and throughput (TPS) of 169.16. On a local network with 5 peers and 8 orderers the means were the following: send rate (TPS) of 115.16, maximum and minimun latency of 4.39s and 0.119s, average latency of 1.569s and throughput (TPS) of 112.53. The smaller time of the results obtained with Caliper compared to the tests with the custom benchmark tool is due to the change of the aforesaid tool, and this is proven by repeating the tests with a single orderer using Caliper as shown above, where the average latency for all the tests was 1.005 seconds. These results are represented on Figure 5, and it is proven that the presence of more orderers affect the latency of each transaction, but the impact is not linear. The number of endorsers didn't significantly impact each transaction's latency, so it is safe to assume that the transaction time, in this case, would be similar to the CosmWasm implementation. The small standard deviation and amplitude seen in Table II shows a consistency across the obtained times, which is important for the end user experience in a real case scenario.

For CosmWasm, the results in Table IV shows that the average transaction time on a local network with a single validator is similar to Fabric. With the use of test networks (testnets) in Figure 7, the complexity in processing transactions and consensus in Tendermint is increased with the number of active validators caused by the block size and also the increase in consensus and network message traffic [23].

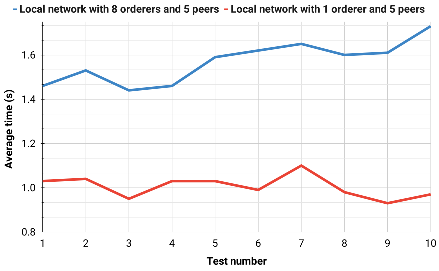

Figure 5. Comparison of the average latency of the Fabric test networks

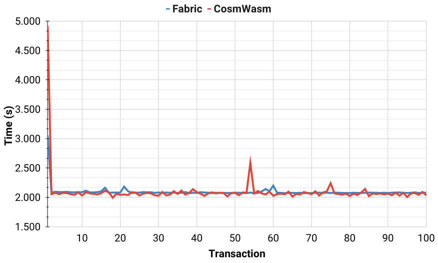

Figure 6. Time for each transaction on a local network with one validator

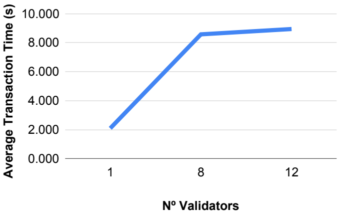

Figure 7. Average transaction time using the CosmWasm platform - Local network with a single validator and a test network with 8 and 12 validators

C. Deployment Complexity

The deployment time measurement had more different results, as shown in Table V. The CosmWasm implementation took 2.489s on average to deploy the contract on a local

Table IV
EVALUATION OF THE COSMWASM PLATFORM WITH 1, 8 AND 12 VALIDATORS

Nº Validators	Average Transaction Time (s)	Standard Deviation (s)	Amplitude (s)
1 (Local)	2.099	0.291	2.924
8 (Testnet)	8.574	1.418	6.572
12 (Testnet)	8.940	3.199	18.575

Table V
AVERAGE TIME TO UPLOAD CONTRACTS BETWEEN PLATFORMS (IN SECONDS)

Platform	Single Validator (Local) (s)	8 Validators (Testnet) (s)	12 Validators (Testnet) (s)
Fabric	51.850	-	-
CosmWasm	2.489	9.898	11.810

network, 9.898s on a testnet with eight validators and 11.810s on a testnet with twelve validators. In comparison, the Hyperledger Fabric implementation took 51.850s on average to be deployed on the local network. The difference can be explained by the Hyperledger Fabric's format of deploying a smart contract, which follows the steps of installing the contract on the network and requiring the majority of the channel members to approve it and for it to be committed. The time of the deployment on a testnet also could not be measured. Still, it may be assumed that it would increase significantly based on the number of peers and if the default approval policy was implemented (which requires the approval of the majority of the nodes).

VII. CONCLUSIONS

We present an electronic prescription system using smart contracts through a decentralized and fault-tolerant architecture. This technology can help manage electronic prescriptions to prevent patient misuse of medications and make it hard-to-change confidential data to obtain illegal benefits, as is possible in a centralized architecture. We implemented it using Tendermint, a PBFT blockchain platform. It is acknowledged that smart contracts in PBFT platforms are not explored in the literature yet. As a result, we have investigated how smart contracts can be implemented over PBFT-based platforms such as Tendermint. We have included an evaluation comparing the performance of Tendermint and Fabric DLTs. For Tendermint, consensus and network message traffic, as well as transaction time, depend on the number of validator nodes on the network, which means the higher number of validator nodes would provide stronger BFT guarantees while incurring high latency.

ACKNOWLEDGMENT

Rodrigo Dutra Garcia would like to acknowledge the financial support granted by CNPq (National Council for Scientific and Technological Development , process 133470/2020-2) in his research. Jó Ueyama would like to thank FAPESP for funding the bulk of his research work (grant ID 2018/17335-9).

REFERENCES

[1] Clair White, Justin Ready, and Charles M. Katz. Examining how prescription drugs are illegally obtained: Social and ecological predictors. *Journal of Drug Issues*, 46(1):4–23, 2016.

[2] John Martin Corkery Stefania Chiappini, Amira Guirguis and Fabrizio Schifano. Misuse of prescription over-the-counter drugs to obtain illicit highs: how pharmacists can prevent abuse. *The Pharmaceutical Journal*, page 30, 11 2020.

[3] Yan Chen and Cristiano Bellavitis. Blockchain disruption and decentralized finance: The rise of decentralized business models. *Journal of Business Venturing Insights*, 13:e00151, 2020.

[4] H. Dai, Z. Zheng, and Y. Zhang. Blockchain for internet of things: A survey. *IEEE Internet of Things Journal*, 6(5):8076–8094, 2019.

[5] Hesam Hamledari and M. Fischer. Construction payment automation using blockchain-enabled smart contracts and reality capture technologies. *ArXiv*, abs/2010.15232, 2020.

[6] Z. Zheng et al. An overview on smart contracts: Challenges, advances and platforms. *ArXiv*, abs/1912.10370, 2020.

[7] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son. Recent advances in smart contracts: A technical overview and state of the art. *IEEE Access*, 8:117782–117801, 2020.

[8] Elli Androulaki et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, New York, NY, USA, 2018. Association for Computing Machinery.

[9] Jae Kwon. Tendermint : Consensus without mining. 2014.

[10] Ethan Buchman. *Tendermint: Byzantine fault tolerance in the age of blockchains*. PhD thesis, 2016.

[11] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, page 173–186, USA, 1999. USENIX Association.

[12] R.D Garcia. e-prescription model using smart-contracts. https://github.com/rodrigodg1/e-prescription, 2021.

[13] E Lau. Decoding the hype: Blockchain in healthcare-a software architecture for the provision of a patient summary to overcome interoperability issues. Master's thesis, 2018.

[14] E. Chukwu and L. Garg. A systematic review of blockchain in healthcare: Frameworks, prototypes, and implementations. *IEEE Access*, 8:21196–21214, 2020.

[15] Camden Thatcher and Subrata Acharya. Rxblock: Towards the design of a distributed immutable electronic prescription system. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 9, 08 2020.

[16] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman. Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 25–30, 2016.

[17] Maria Inês da Fonseca Ribeiro. and André Vasconcelos. Medblock: Using blockchain in health healthcare application based on blockchain and smart contracts. In *Proceedings of the 22nd International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 156–164. INSTICC, SciTePress, 2020.

[18] Qi Xia et al. Medshare: Trust-less medical data sharing among cloud service providers via blockchain. *IEEE Access*, PP:1–1, 07 2017.

[19] B. Ying, W. Sun, N. R. Mohsen, and A. Nayak. A secure blockchain-based prescription drug supply in health-care systems. In *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pages 1–6, 2019.

[20] Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications*, 50:102407, 2020.

[21] H. Wu and C. Tsai. Toward blockchains for health-care systems: Applying the bilinear pairing technology to ensure privacy protection and accuracy in data sharing. *IEEE Consumer Electronics Magazine*, 7(4):65–71, 2018.

[22] Smart contracts: Building blocks for digital markets. https://cutt.ly/XvAXXVC. Accessed: 04-21-2021.

[23] Michael Merz. *Blockchain for B2B Integration*, volume 1. MM Publishing, 1 edition, 2020.

[24] Qi Feng, Debiao He, Sherali Zeadally, Muhammad Khurram Khan, and Neeraj Kumar. A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications*, 126:45–58, 2019.

[25] T. T. Huynh, T. D. Nguyen, and H. Tan. A survey on security and privacy issues of blockchain technology. In *2019 International Conference on System Science and Engineering (ICSSE)*, pages 362–367, 2019.

[26] M. T. de Oliveira, L. H. A. Reis, R. C. Carrano, F. L. Seixas, D. C. M. Saade, C. V. Albuquerque, N. C. Fernandes, S. D. Olabarriaga, D. S. V. Medeiros, and D. M. F. Mattos. Towards a blockchain-based secure electronic medical record for healthcare applications. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[27] Salma Shalaby, Alaa Abdellatif, Abdulla Al-Ali, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. Performance evaluation of hyperledger fabric. 04 2020.

CHAPTER

3

EXPLOITING SMART CONTRACTS IN PBFT-BASED BLOCKCHAINS: A CASE STUDY IN MEDICAL PRESCRIPTION SYSTEM

This chapter presents the article published in the Elsevier Computer Networks under the following Elsevier permission:

Contribution Statement: Software, Data curation, Writing – review & editing.

[1] <https://doi.org/10.1016/j.comnet.2022.109003>

Exploiting smart contracts in PBFT-based blockchains: a case study in medical prescription system

Rodrigo D. Garcia[a,*], Gowri Ramachandran[b,*] and Jó Ueyama[a,*]

[a] Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, SP, Brazil
[b] School of Computer Science, Queensland University of Technology, Australia

ARTICLE INFO

Keywords:
Blockchain
Smart Contracts
Tendermint
Byzantine Fault Tolerance (BFT)
Electronic Prescriptions

ABSTRACT

Smart contracts allow application developers to automate business processes through a decentralized computation architecture. Contemporary blockchain platforms such as Ethereum and Hyperledger Fabric offer support for smart contracts through consensus mechanisms such as Proof-of-Work (PoW) or other types of transaction validation and ordering services. This article exploits smart contracts in the Byzantine Fault Tolerant (BFT) blockchain platforms. In particular, we explore Tendermint and Hyperledger Besu, BFT blockchain platforms, and apply them to a decentralized e-prescription case study to evaluate their effectiveness. We adopt Hyperledger Besu and Tendermint in this research, given that both are BFT-based blockchains. Also, it is noteworthy that smart contracts in BFT blockchain platforms such as Tendermint are not well established and not widely adopted yet. Our article empirically evaluates the performance of smart contracts in Tendermint and Hyperledger Besu using a decentralized medical prescription case study and compares their results with Ethereum, a PoW blockchain. Our results demonstrate that BFT blockchain platforms are efficient for multi-stakeholder applications such as e-prescription and supply chains. To the best of our knowledge, this is the first study investigating the implementation of smart contracts in BFT blockchain platforms, such as Tendermint and Hyperledger Besu.

1. Introduction

Digital healthcare systems help healthcare agencies efficiently manage patients' information, including their prescription history. The use of technology to share information about the patient (especially in periods of social distancing) enables more efficient communication between healthcare professionals and organizations through computational and multiplatform digital applications. The adoption of digital prescriptions allows efficient communication while avoiding inconsistencies compared to paper-based prescriptions, providing a better quality of health service to the patient [1].

However, most solutions use centralized digital systems to manage medical records [1]. Centralized architectures are susceptible to a single point of failure problem, allowing healthcare agencies to tamper or misuse patient records. Therefore, trust between healthcare organizations and the availability of records depend on a single central server, as in Figure 1. On the other hand, blockchain technology is decentralized and guarantees the integrity of records. By design, blockchain operates without dependence on a central authority or intermediary, and records are added to the chain through consensus among network nodes.

Smart contracts are executable programs stored and run on the blockchain and allow for automating tasks such as validating transactions without third-party intervention [2]. One of the most popular decentralized application development platforms using smart contracts is Ethereum. However, it uses Proof of Work (PoW), a high operational cost consensus mechanism in which the mining node must solve a cryptographic challenge [3, 4]. Practical Byzantine Fault Tolerance[1] (PBFT) platforms such as Tendermint and Istanbul Byzantine Fault Tolerance 2 (IBFT2) are an alternative to the high computational cost of PoW blockchains, as these platforms achieve consensus without mining. The consensus is divided into steps, and the participants are known as validators and act by proposing and validating blocks.

This work proposes a decentralized electronic prescription system using blockchain technology and smart contracts while employing BFT-based blockchain platforms. Our architecture prevents the central point of failure and provides transparency and privacy guarantees by leveraging immutable ledger and encryption techniques, respectively. As an alternative to the high operational cost of PoW, we evaluated the the effectivesness of our architecture using two BFT blockchain platforms: Tendermint and IBFT2. For this, we used the CosmWasm platform for Tendermint and Hyperledger Besu for IBFT2 and compared their operating costs with Ethereum's PoW approach. All software developed in this work is available on GitHub [5]. To the best of our knowledge, this is the first study comparing BFT smart contract platforms with PoW for healthcare applications such as electronic prescriptions.

The rest of the article is structured as follows: Section 2 lists the requirements for a trusted electronic prescription system and motivates the need for a blockchain-based solution. The related work and the gaps are discussed in Section 3. Section 4 introduces the blockchain technology and byzantine fault tolerance blockchain platforms. The proposed decentralized e-prescription system is presented in

*Corresponding author
rgarcia@usp.br (R.D. Garcia); g.ramachandran@qut.edu.au (G. Ramachandran); joueyama@icmc.usp.br (J. Ueyama)

[1] PBFT and BFT are used interchangeably in the rest of the manuscript.

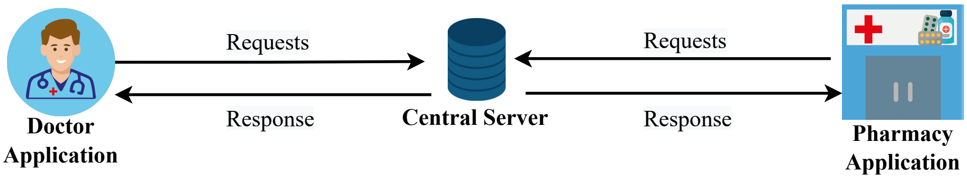

Figure 1: Electronic prescribing using centralized architecture

Section 5. Section 6 provides an overview of our implementation. The evaluation results are presented in Section 7. Our point of view on the adoption of Ethereum and Tendermint is presented in Section 8 while the limitations of the proposed scheme is discussed in Section 9. Section 10 concludes the article.

2. Requirements and Blockchain-based solution

To ensure a robust system for electronic medical prescriptions, we define the requirements based on the problems caused by the misuse of prescriptions [6, 7, 8]:

- **R1: Decentralized system:** Run the digital infrastructure following a distributed architecture.
- **R2: Fault-tolerant:** The system must remain available even when some nodes in the network fail.
- **R3: Immutability of medical records and medicine sales:** Prevent manipulation of medical records by a central authority or a third party.
- **R4: Record traceability and provenance:** Keep track of the origin and destination of medical records and medicine sales.
- **R5: Privacy:** Sensitive patient data must not be visible to unauthorized stakeholders.

Blockchain is a technology that has intrinsic characteristics that meet most of these requirements for preventing fraud and forging prescriptions following a decentralized architecture. In this sense, it prevents the patient from having health problems with the misuse of medications. In particular, it provides the following benefits to e-prescription applications:

- Operates in a decentralized fashion without a central authority (no single point of failure)
- Maintains patient records in redundant storage
- Prevents tampering of records for self-benefit
- Sale of medicines only with valid prescriptions
- Efficient and transparent communication between stakeholders
- Reduction of errors and inconsistencies in medications

Thereby, it is possible to create a decentralized and fault-tolerant prescription model. The addition of new medical records or the sale of drugs is carried out through a distributed consensus among the participants in the network. As a result, communication between stakeholders becomes more secure than centralized systems and paper-based prescriptions.

3. Motivation and Related Work

Exploring the features of blockchain technology for applications in the healthcare field has proved to be relevant, especially when the requirements are availability, integrity, and transparency of records [2, 9]. Dubovitskaya et al. proposed ACTION-EHR [10], a permissioned blockchain system to allow patients to manage their records across multiple hospitals. In particular, the authors analyzed data sharing for radiation treatment for cancer using Hyperledger Fabric platform. Dagher et al. introduced Ancile [11], a framework implemented using the Ethereum platform to manage electronic medical records. The main objective is access control among healthcare industry stakeholders while preserving patient privacy.

Other authors explore blockchain technology for the electronic prescription use case to manage and track records among healthcare professionals such as prescribers (doctors), hospitals, pharmacies, and patients. Li et al. introduced DMMS [12], a decentralized medication management system to create and query medical prescription records using Hyperledger Fabric platform. Similarly, He et al. proposed BlockMeds [13], a solution developed through Hyperledger Fabric platform for medical records such as electronic prescriptions with data anonymization to sell medicines.

Zhang et al. introduced OpTrak [14], a decentralized solution using the Ethereum platform to mitigate the opioid epidemic within a consortium of networks between healthcare organizations. Thatcher and Acharya proposed RxBlock [15], an electronic prescription system exploiting blockchain technology. The proposed model was evaluated using the Ethereum platform. Similarly, Alnafrani and Acharya proposed SecureRx [16], a framework for electronic prescriptions using the Ethereum platform. The main objective is to monitor the use of medicines by patients.

However, none of these works above analyzed the operational cost of the solutions, comparing them with other platforms. In this work, we proposed a decentralized electronic prescribing system through blockchain technology. We analyzed the operational costs of BFT platforms. In particular, we explored Tendermint consensus mechanism using the CosmWasm smart contract platform and the IBFT2 consensus using Hyperledger Besu platform. These platforms are not widely adopted in the literature, especially for cost-critical applications such as healthcare. As a comparison, we used Ethereum, a platform popularly adopted in several application domains such as academic research and industrial solutions.

4. Background

4.1. Blockchain and Smart Contracts

Blockchain technology combines encryption mechanisms, Peer-to-peer (P2P) networking, concepts such as transactions, and blocks to develop decentralized solutions with no central authority or intermediary managing records [17]. Blocks are created in chronological order and linked through cryptographic hashes to prevent transaction tampering. The users send the data through transactions, and it will be added to the blockchain through a consensus criterion between the network nodes.

Smart contracts are sophisticated mechanisms for creating business rules to automate tasks like validating transactions submitted by users. Within the blockchain, smart contracts are executable and immutable programs developed using a programming language compatible with the blockchain platform. The main feature of smart contracts is to eliminate an intermediary or third party to perform some tasks like transferring assets between accounts. Interaction with contract methods requires an address for the application to send transactions, and valid transactions update the current state of the contract records [18]. In addition to the financial sector [19], blockchain and smart contracts are explored in sectors such as healthcare [2, 9], supply chain [20], and Internet of Things (IoT) [21].

4.2. Tendermint Consensus

Tendermint consensus is a variation of the Practical Byzantine Fault Tolerance PBFT [22]. It is used by the Cosmos blockchain ecosystem, enabling the development of general-purpose applications [23]. Tendermint was built as a lower-cost alternative than the Bitcoin and Ethereum Proof of Work consensus algorithm with flexibility in application development independent of the programming language. In this way, the application communicates with Tendermint through the Application Blockchain Interface (ABCI), as shown in Figure 2.

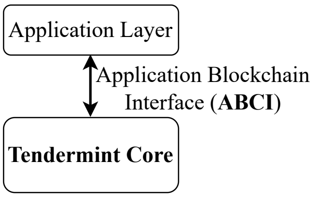

Figure 2: Communication of the application with Tendermint Core

Figure 3 presents the steps of Tendermint consensus, which is performed through rounds and steps [24]. The nodes participating in the consensus are validators and hold a pair of keys (public and private) to sign and verify blocks and transactions. In the first step, a validator node acts as a proposer creating a block with a set of valid transactions stored in the mempool. The proposer node signs the block with the private key and transmits it to the other validator nodes. Each validator node will check and validate the block. For the block to be added to the blockchain, 2/3 of the network must send the *prevote* message and then the *precommit* message. After the commit (including the block in the blockchain), a new round is started with another proposer node.

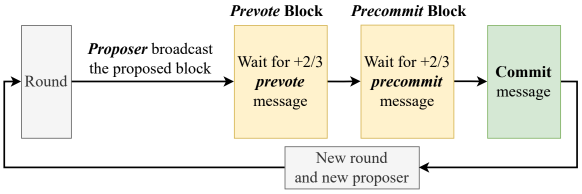
Prevote Block
Precommit Block
Round
Proposer broadcast the proposed block
Wait for +2/3 *prevote* message
Wait for +2/3 *precommit* message
Commit message
New round and new proposer

Figure 3: Tendermint consensus mechanism

According to the expression $N \geq 3f + 1$ in practical byzantine fault tolerant consensus mechanism [25], including Tendermint, N active nodes are required for f simultaneous failures. For example, in case of a failure ($f = 1$), it must have at least four active nodes ($N \geq 4$). Figure 4 illustrates network configurations for up to four simultaneous failures.

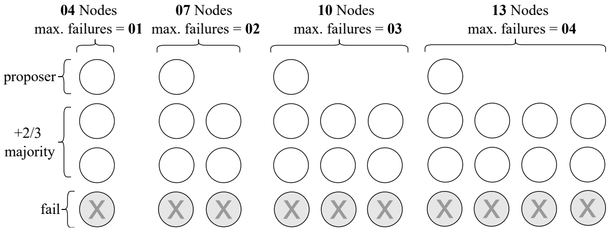

Figure 4: Network representation for up to 4 simultaneous faults

Figure 5 represents, as an example, the traffic of consensus and network messages carried out by Tendermint between four validator nodes. The topology uses the complete mesh configuration, and the choice of the proposer node is made alternately.

4.3. Hyperledger Besu and IBFT2

Hyperledger Besu is an open-source Ethereum client that enables the development of public and private blockchain solutions using different consensus algorithms, including PoW and IBFT2. Hyperledger Besu can be used for enterprise

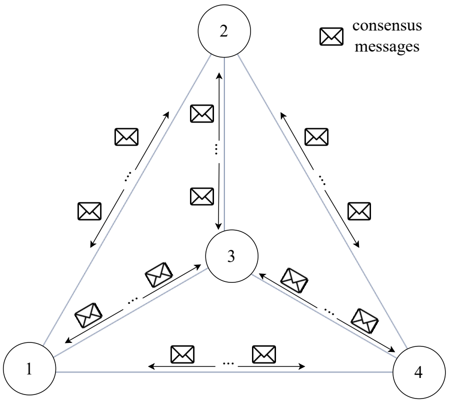

Figure 5: Representation of message traffic between 4 validator nodes

applications using private networks where high performance is required for processing transactions [26].

Besu implements the IBFT2 consensus algorithm, a Proof of Authority (PoA) protocol in which the nodes participating in the consensus are known as in permissioned settings. IBFT2 is a variation of PBFT where consensus is based on a leader (like Tendermint proposer node) and stages (i.e., pre-prepare, prepare and commit). Fault tolerance follows the same expression used by Tendermint (i.e., $N \geq 3f + 1$) where f is the number of faults, and N is the number of active nodes in the system [27]. Similar to the Tendermint steps in Figure 3, the block will only be added to the blockchain if the majority of validators ($\approx 66\%$) sign the block during all stages [27].

5. A Decentralized e-Prescription System

In this section, we present a model for medical records. In particular, we explore electronic prescriptions to enable the sale of medicines only with valid records (i.e., created and signed by the doctor) through blockchain technology and smart contracts ensuring integrity, availability, and information transparency. Our proposed system avoids: (i) creating invalid digital prescriptions (without the doctor's digital signature); (ii) medical records tampering, such as digital prescriptions. In this way, it avoids the misuse of medications and adverse consequences such as overdoses [8].

The proposed model can be used as a framework for future research in healthcare applications. The software developed to evaluate the decentralized prescribing system is available on GitHub [5].

5.1. Model Architecture

This work presents a decentralized model for electronic prescribing systems through blockchain technology and smart contracts. Figure 6 presents the steps using the decentralized architecture. From an appointment with the patient (step 1), the doctor creates a digital prescription with information about medication, and diagnosis. In centralized architecture, this record is stored on a server shared between the doctor and pharmacy application (as in Figure 1). In this way, all stakeholders trust the record's authenticity through a central server. Newaz et al. [28] analyzed a series of attacks on healthcare systems with centralized architecture.

We propose a decentralized architecture for the digital prescription system (i.e., without any central server or intermediary managing the medical records). We implement smart contracts to validate doctor transactions and medication sales. Each patient has a prescription instance contract with the doctor of a clinic or hospital and another instance for the medication sales contract with the pharmacy.

From the application connected with the wallet containing the public and private key pair, the doctor creates a transaction containing the prescription data, signs it with the private key, and sends it to the prescription contract on the network through the contract address (step 2 in Figure 6). With the doctor's public key, the other nodes verify the authenticity of the prescription, preventing the creation of false records. The transaction will be added to the blockchain through a consensus algorithm between the network nodes. In this work, we evaluated three approaches: Tendermint, IBFT2, and PoW.

The pharmacy verifies the drug prescribed by the doctor by consulting the current state of the patient's prescription contract (step 3 in Figure 6). Similar to the doctor's application, the pharmacy has a wallet with a pair of public and private keys to sign medication sales transactions for the patient (step 4 in Figure 6). The sales transaction is sent to the sales contract, and the patient receives the medication (step 5 in Figure 6).

5.2. Data Privacy

In our model, sensitive data such as Personally Identifiable Information (PII) and diagnostics must be protected when sharing electronic prescriptions. We propose a way to access records only by authorized parties. For this, the patient has a pair of keys, and the doctor will use the public key to encrypt the prescription data (i.e., medication and diagnosis). The patient will use the private key to decrypt the information when visiting the doctor or pharmacy.

In an appointment with the patient (steps 1 and 2 in Figure 6), the doctor creates the prescription without Personal Identifiable Information (PII), encrypts the information about the medication and diagnosis, and sends it to the prescribing contract. Transactions are submitted in encrypted form before being stored on the ledger. Only the authorized parties (doctor and pharmacy) can decrypt it. Therefore, the patient's sensitive data is not publicly available on the ledger for everyone to see.

The transactions are submitted using the anonymous public key. We are not storing doctor and patient's identifiable information. The information will only be revealed when the patient visits the doctor (step 1) or pharmacy (step 3) and the patient must use his private key to decrypt the prescription data. In summary, we provide the following privacy guarantees:

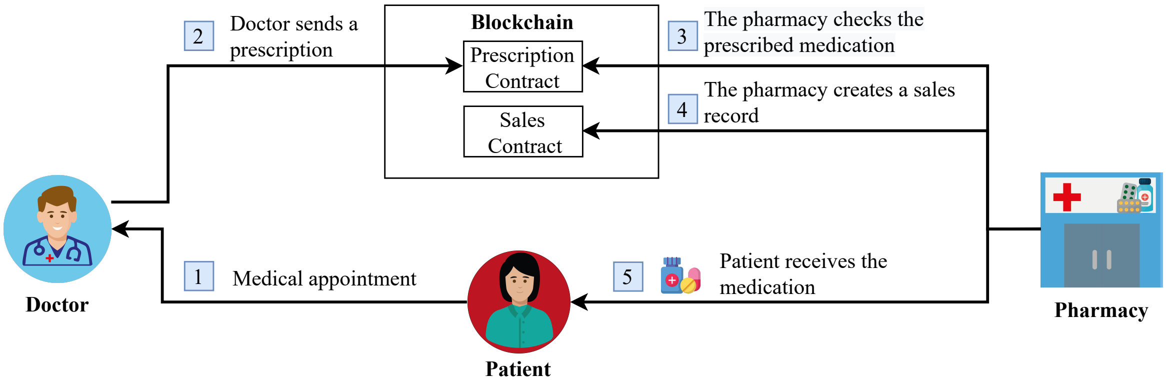
Blockchain
Prescription Contract
Sales Contract
2 Doctor sends a prescription
3 The pharmacy checks the prescribed medication
4 The pharmacy creates a sales record
1 Medical appointment
5 Patient receives the medication
Doctor
Patient
Pharmacy

Figure 6: Electronic prescribing model using decentralized architecture

- Patients and doctors are anonymous in the blockchain-based digital prescription system because they only use their public key for their transactions. When a patient visits a pharmacy, a pharmacist may associate the patient with their public key, but such situations cannot be avoided in e-prescription application.
- Data is stored in encrypted form. Therefore, the information is not visible to the participants in the blockchain network.

We plan to extend the privacy architecture by considering Shamir's secret sharing approach [29] in our future work, which will allow the pharmacy to decrypt only when a sufficient number of keys are revealed. In this way, a prescription can only be decrypted when the doctor and patient share their secret with the pharmacy.

5.3. System Guarantees

Section 2 lists the requirements for a trusted, decentralized, transparent, and privacy-preserving e-prescription systems. We detail how our system fulfils the requirements below: Section 2 lists the requirements for trusted, decentralized, transparent, and privacy-preserving e-prescription system. We detail how our system fulfills the requirements below:

Meeting R1 - Decentralized system: Our system employs a byzantine fault-tolerant blockchain platform involving multiple permissioned stakeholders to manage patients' prescription records.

Meeting R2 - Fault-tolerant: Our system relies on a byzantine fault-tolerant consensus mechanism, which can tolerate one fault in a five-node network. As long as more than two-thirds of active and honest nodes are in the network, the byzantine fault-tolerant consensus algorithm guarantees safety and liveness.

Meeting R3 - Immutability of medical records and medicine sales: The blockchain platforms include immutable storage, which is a write-once ledger. Any transaction, including the prescription data, stays in the blockchain ledger once written. Any effort to modify the data in the ledger will make the ledger invalid.

Meeting R4 - Record traceability and provenance: The ledger holds the records of prescription data belonging to a patient in encrypted form. When a healthcare agency needs to audit the patient's healthcare data, it is possible to do so with the patient's consent and decryption key.

Meeting R5 - Privacy Follows from guarantees in Section 5.2.

5.4. System Utility and Importance

Our solution allows doctors, pharmacies, patients, and healthcare agencies to provide transparency and provenance to e-prescription systems. However, the real-world adoption of such a system largely depends on the support of the concerned stakeholders, as they may hesitate due to their lack of knowledge and unwillingness to switch from legacy systems. We believe that the architecture is viable if backed by more stakeholders.

Besides, our architecture applies to any multi-stakeholder applications, including the supply chain. Data provenance, transparency, and privacy are essential in supply chain applications.

6. Implementation Overview

Our model proposes the prescription contract for submitting new medical records and the sales contract for sales transactions. The prescription contract only accepts transactions from the doctor and validates transactions containing data such as medication and diagnosis sent by the doctor. Therefore, the contract state is updated by the *set_prescription* method with the information related to prescribed medication.

The prescription contract has a *get_prescription_info* method to query the history of records, including the current state of the contract. This method is required by the doctor's application to consult records related to the patient, including previous diagnoses.

In the pharmacy application, the required method is *get_medication_info* to query the prescribed medication by the doctor. Transactions with medication sales information (i.e., name and price) are sent to the *create_sale* method of the sales contract. The pharmacy will request the patient's sales history through the *get_sales_history* method.

In this work, the contract was implemented using the Rust programming language for the CosmWasm platform and Solidity for Hyperledger Besu and Ethereum platforms.

7. Evaluations

In this work, we evaluated the operational cost of the nodes participating in the consensus (validators) by analyzing the memory and CPU usage during the consensus mechanism. We use Tendermint (CosmWasm), Hyperledger Besu, and Ethereum platforms to develop and evaluate the prescription contract. Using these platforms, we evaluate the time required for a transaction submitted by the doctor containing prescription data (i.e., medication and diagnosis) to be validated and included in the blockchain.

7.1. Evaluation Setup

To evaluate the model, we used a Linux virtual machine with an Intel Core i7-10510U processor with four CPUs (4x2.30GHz) and 8GB of RAM. In Hyperledger Besu configuration, the IBFT2 consensus mechanism was used with the block generation time (block time) every five seconds. We use four validator nodes on a local network to evaluate memory and CPU usage during consensus and the average time required for a transaction submitted by a client to be included in the blockchain.

Similarly, in CosmWasm configuration, we used a testnet provided by the platform with four validator nodes and a block generation time of five seconds [30]. To evaluate the memory and CPU usage of the CosmWasm validator, we used only one validator node on a local network. We explore Ethereum, a widely adopted platform for developing decentralized applications with smart contracts to compare the time required to mine the block with the transaction sent by the client (i.e., doctor) using the Ropsten testnet.

To analyze the memory and CPU usage by the validator nodes during the consensus steps, we used the *docker stats* command and Grafana, an open-source graphical visualization tool. We implement clients using *web3.js*, the Ethereum JavaScript API, to evaluate sending transactions and shell script to automate sending, querying, and storing blockchain data. Data such as the block's timestamp containing the transaction, memory usage, and CPU were stored in text files. Table 1 shows the configurations used to evaluate the electronic prescriptions model. All software developed for analysis and smart contracts is available on GitHub [5].

7.2. Operational Cost

7.2.1. *Tendermint (CosmWasm)*

Sending transactions to the *set_prescription* method using the client code, the validator node used an average of 74.44 Megabyte (MB) to perform operations to validate transactions and create blocks. Average CPU usage was 1.68% with a peak of 8.77% across the four evaluation virtual machine cores. Table 2 shows the minimum (min.), maximum (max.), average (avg.) usage, and the standard deviation (std.) of the executions performed by the script every second. Figure 7 shows the CPU usage during 200 seconds of validator node operation.

Setup	Tendermint (CosmWasm)	Hyperledger Besu	Ethereum
Version	0.23.0 (wasmd)	21.10.9	-
Consensus	Tendermint	IBFT2	PoW
Block Time	5s	5s	-
Testnet	4 validator nodes	4 validator nodes (local)	Ropsten
Testnet Type	Permissioned	Permissioned	Public

Table 1
Setup evaluation between CosmWasm, Hyperledger Besu and Ethereum

Tendermint (CosmWasm) Validator	Min.	Max.	Avg.	Std.
Memory Allocated (MB)	72.39	76.23	74.44	0.88
CPU Used (%)	0.03	8.77	1.68	1.67

Table 2
Memory allocated and CPU used by Tendermint (CosmWasm) node during transaction validation and block creation

	Memory Allocated (MB) by Validator Nodes			
	Node 01	**Node 02**	**Node 03**	**Node 04**
Min.	592	600	583	511
Max.	611	608	617	525
Avg.	602	605	607	520
Std.	5.86	2.80	11.84	4.19

Table 3
Memory allocated by Hyperledger Besu validators during IBFT2 consensus

7.2.2. *Hyperledger Besu*

In Hyperledger Besu configuration, the average memory usage by the validator nodes was around 600 MB during transaction validation and block creation. Figure 8 shows the average memory usage by each validator, and Table 3 shows the minimum, maximum, average usage, and standard deviation of executions performed by the shell script every second. Figure 9 shows CPU utilization during 200 seconds of validation and block creation, and Table 4 shows minimum, maximum, average usage, and standard deviation.

On average, the CPU usage by the validators was around 3.28% for validator 1, with a maximum peak of 18%. Validator 2 used an average of 2.83% of the CPU and a maximum of 20%. Similarly, validator 3 used an average of 2.67% and a maximum of 17%, and validator 4 used an average of 3.79% and a maximum of 20%.

7.3. Smart Contract Evaluation

We evaluate the time required for a transaction containing 1 Kilobyte (kB) of data (i.e., medication and diagnosis)

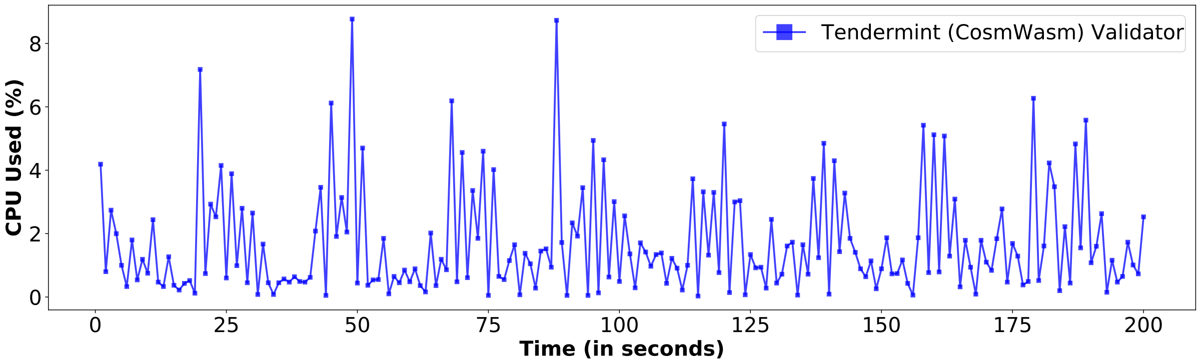

Figure 7: CPU used by Tendermint (CosmWasm) validator during 200 seconds of operation

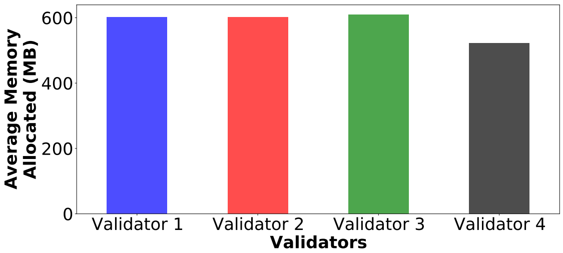

Figure 8: Average memory allocation by Hyperledger Besu validators during IBFT2 consensus

	CPU Used (%) by Validator Nodes			
	Node 01	**Node 02**	**Node 03**	**Node 04**
Min.	0	0	0	0
Max.	18	20	17	20
Avg.	3.28	2.83	2.67	3.79
Std.	3.36	3.06	2.88	3.73

Table 4
CPU used by Hyperledger Besu validators using docker container configuration during IBFT2 consensus

sent by the doctor application to be validated by the prescription contract and included in the blockchain. We analyzed the following BFT platforms: Tendermint (CosmWasm) and Hyperledger Besu. In the evaluation, 1000 consecutive transactions were sent to the *set_precription* method of the prescription contract. As a comparison, we analyzed the Ethereum platform with the PoW consensus algorithm. Table 5 shows the minimum, maximum, and average block generation time between the evaluated platforms, and Figure 10 shows the block time between the platforms for our decentralized electronic prescription model.

7.3.1. Tendermint (CosmWasm)

Using a non-local testnet with four validator nodes, the average block generation time and inclusion in the blockchain with the transaction submitted by the client (i.e., doctor application) was 5.40 seconds with a maximum of 5.61 seconds and a minimum of 5.28 seconds. The standard deviation (i.e., block generation time dispersion) was 0.06 seconds during all transactions.

7.3.2. Hyperledger Besu (IBFT2)

The average block time and inclusion in the blockchain using Hyperledger Besu implementation configured on a local network with four validator nodes was around 5 seconds, with a maximum time of 5.75 seconds and a minimum of 4.43 seconds. There was small dispersion in the block generation time with a standard deviation of 0.49 seconds in all transactions.

7.3.3. Ethereum (PoW)

The average block mining time in Ethereum platform evaluation with the Rospten testnet was around 23.79 seconds, with a maximum time of 107.84 seconds and a minimum of 2.95 seconds. Compared to Tendermint (CosmWasm) and Hyperledger Besu platforms, mining time proved to be more dispersed with a standard deviation of 17.78 seconds.

7.4. Scalability

Despite increasing the fault tolerance, the addition in the number of validator nodes in BFT platforms increases the message traffic between the nodes, harming the network performance [23]. According to an evaluation performed by Hyperledger Besu team, IBFT2 handles up to 30 validators without performance loss [31]. Cason et al. [23] analyzed Tendermint performance under different conditions, and the results showed a decrease in Throughput (TPS) and an increase in latency with the addition of validator nodes.

7.5. Discussion of results

Unlike the PoW used by the Ethereum platform, the operational cost of validator nodes on BFT platforms that use consensus without mining, such as Tendermint (CosmWasm) and Hyperledger Besu, proved to require low computational resources, especially for processing transactions and creating blocks. Therefore, BFT solutions for healthcare applications such as electronic prescriptions contribute to using a consensus mechanism with low computational consumption and less time to include the transaction to the blockchain. On the other hand, the PoW algorithm is used in public networks

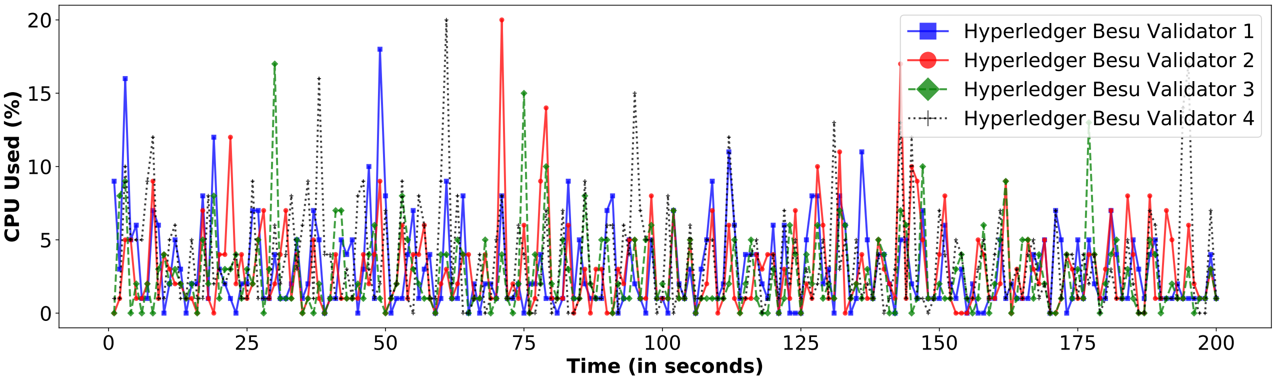

Figure 9: CPU used by Hyperledger Besu validators during IBFT2 consensus

	Block creation time and inclusion in the blockchain (in seconds)		
	Tendermint (CosmWasm)	Hyperledger Besu (IBFT2)	Ethereum (PoW)
Min.	5.28	4.43	2.95
Max.	5.61	5.75	107.84
Avg.	5.40	5.00	23.79
Std.	0.06	0.49	17.78

Table 5
Minimum, maximum and average time (in seconds) for a transaction to be added in a blockchain between CosmWasm, Hyperledger Besu and Ethereum platforms for a total of 1000 consecutive transactions

with many nodes such as Bitcoin and Ethereum. The miner must prove participation through a computational effort to solve a cryptographic challenge [4, 32]. Solutions using PoW have a higher energy consumption when mining a new block [3].

8. Ethereum and Tendermint Adoption

The Ethereum platform enabled the development of decentralized applications using smart contracts [33]. It became popular for simplifying the development of blockchain applications in different use cases [34, 35]. For this reason, Ethereum is still widely adopted in academic research and business in the industry sector.

As an alternative to Ethereum's PoW consensus mechanism, Tendermint was developed to allow application developers to adopt a lightweight and flexible solution using the BFT algorithm [24]. Tendermint is used as the consensus engine of the Cosmos ecosystem and has become popular for developing blockchain projects. For example, we used CosmWasm, a smart contract module on top of the Cosmos framework. There are several other applications developed through the Cosmos ecosystem for different domains [36]. Compared to Hyperledger Fabric with Solo ordering scheme, Tendermint relies on a decentralized consensus scheme without a centralized intermediary, improving trust while overcoming single point of failure.

We have shown that Tendermint is suitable for multi-stakeholder permissioned applications while allowing the application developers to build smart contracts in popular programming languages such as JavaScript and Go. Hyperledger Besu is also a viable BFT blockchain platform, but it demands the user to implement smart contracts in Solidity.

9. Limitations

In this work, we propose a decentralized model using blockchain technology and smart contracts to share prescription data while maintaining record integrity through BFT blockchain platforms such as Tendermint and Hyperledger Besu, which is still little adopted in the literature for healthcare applications.

Besides, our existing architecture ensures privacy through anonymity as the decentralized ledger only stores prescription data in an encrypted form without including any personally identifiable information. In comparison, some works have analyzed methods for data privacy integrated with blockchain technology. However, privacy was exploited in an off-chain mode. Dagher et al. [11] proposed a solution that stores medical records using encryption to protect patient privacy. In particular, the authors explored symmetric encryption for large files, public encryption for protected health information (PHI), and proxy re-encryption for sharing records with third parties. The blockchain keeps the hashes of records stored in a database outside the blockchain (off-chain).

Similarly, Zou et al. [37] and Chen et al. [38] analyze the proxy re-encryption mechanism as access control and to ensure privacy in the sharing of medical data. However, these approaches do not exploit encryption of sensitive data within the blockchain. Huang et al. [39] proposed a model with differential privacy to protect patient data across healthcare organizations. In this way, doctors will obtain information from the data while preserving patient privacy. Our approach also ensures privacy by storing data encrypted on the blockchain while letting the patient decrypt the prescription data when buying medications from the pharmacy. This approach allows the patient to control their data, but a more robust scheme involving the doctor would further

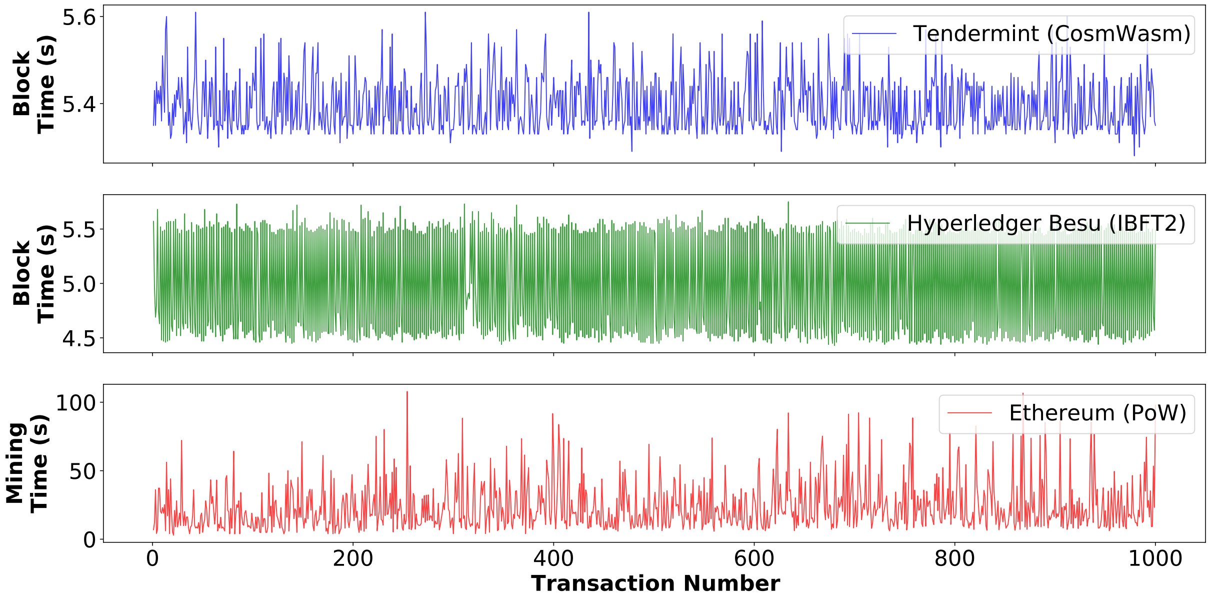

Figure 10: Comparison of smart contract execution and block creation time between Ethereum, Hyperledger Besu and Tendermint (CosmWasm) for a total of 1000 transactions

reduce medication abuse. In our future work, we plan to integrate Shamir secret sharing to let both the doctor and the patient share their secret for decrypting the data via a smart contract to ensure transparency and integrity.

10. Conclusion and future research

We have presented a decentralized e-prescription system using smart contracts on Byzantine fault-tolerant (BFT) blockchain platforms. We have also shown that our system can guarantee decentralization, provenance, privacy, fault tolerance, and immutable storage for the healthcare sector to efficiently manage prescription data. It is acknowledged that smart contracts in BFT platforms are not widely explored in the literature yet. As a result, we have investigated how smart contracts can be implemented over BFT platforms such as Tendermint and Hyperledger Besu. We evaluated the operational cost of BFT platforms compared to Ethereum for healthcare applications such as e-prescriptions. BFT solutions have a lower operating cost when compared to PoW platforms. The addition of new validator nodes in BFT platforms increases fault tolerance. However, performance is degraded as the network complexity increases. We believe that multi-stakeholder applications with up to twenty stakeholders can use BFT blockchain platforms while avoiding PoW mining and transaction fees.

Acknowledgement

Rodrigo D. Garcia would like to acknowledge the financial support granted by CNPq (National Council for Scientific and Technological Development, process 133470/2020-2) in his research.

CRediT authorship contribution statement

Rodrigo D. Garcia: Software, Data curation,Writing - review editing. **Gowri Ramachandran:** Supervision, Conceptualization of this study, Methodology. **Jó Ueyama:** Supervision, Conceptualization of this study, Methodology.

References

[1] Bader Alduhayfiq and Srinivas Sampalli. Digital Health in Physicians' and Pharmacists' Office: A Comparative Study of e-Prescription Systems' Architecture and Digital Security in Eight Countries. *OMICS : a Journal of Integrative Biology*, 25(2):102–122, 2021. ISSN 1536-2310. doi: 10.1089/omi.2020.0085.

[2] Mehdi Sookhak, Mohammad Reza Jabbarpour, Nader Sohrabi Safa, and F. Richard Yu. Blockchain and smart contract for access control in healthcare: A survey, issues and challenges, and open issues. *Journal of Network and Computer Applications*, 178:102950, 2020. ISSN 1084-8045. doi: 10.1016/j.jnca.2020.102950.

[3] Rong Zhang and Wai Kin (Victor) Chan. Evaluation of Energy Consumption in Block-Chains with Proof of Work and Proof of Stake. *Journal of Physics: Conference Series*, 1584(1):012023, 2020. ISSN 1742-6588. doi: 10.1088/1742-6596/1584/1/012023.

[4] Xiang Fu, Huaimin Wang, and Peichang Shi. A survey of Blockchain consensus algorithms: mechanism, design and applications. *Science China Information Sciences*, 64(2):121101, 2021. ISSN 1674-733X. doi: 10.1007/s11432-019-2790-1.

[5] R.D Garcia. Decentralized e-prescription model. https://github.com/rodrigodg1/e-prescription-bft, 2022. Accessed: 2022-02-26.

[6] Mir M. Ali, William N. Dowd, Timothy Classen, Ryan Mutter, and Scott P. Novak. Prescription drug monitoring programs, nonmedical use of prescription drugs, and heroin use: Evidence from the national survey of drug use and health. *Addictive Behaviors*, 69:65 – 77, 2017. ISSN 0306-4603. doi: https://doi.org/10.1016/j.addbeh.2017.01.011. URL http://www.sciencedirect.com/science/article/pii/S030646031730014X.

[7] Clair White, Justin Ready, and Charles M. Katz. Examining how prescription drugs are illegally obtained: Social and ecological predictors. *Journal of Drug Issues*, 46(1):4–23, 2016.

doi: 10.1177/0022042615608502. URL https://doi.org/10.1177/0022042615608502.

[8] John Martin Corkery Stefania Chiappini, Amira Guirguis and Fabrizio Schifano. Misuse of prescription over-the-counter drugs to obtain illicit highs: how pharmacists can prevent abuse. *The Pharmaceutical Journal*, page 30, 11 2020. doi: 10.1211/PJ.2020.20208538.

[9] Swati Megha, Hamza Salem, Enes Ayan, Manuel Mazzara, Hamna Aslam, Mirko Farina, Mohammad Reza Bahrami, and Muhammad Ahmad. Survey on Blockchain Applications for Healthcare: Reflections and Challenges. *Lecture Notes in Networks and Systems*, pages 310–322, 2021. ISSN 2367-3370. doi: 10.1007/978-3-030-75078-7_32.

[10] Alevtina Dubovitskaya, Furqan Baig, Zhigang Xu, Rohit Shukla, Pratik Sushil Zambani, Arun Swaminathan, Md Majid Jahangir, Khadija Chowdhry, Rahul Lachhani, Nitesh Idnani, Michael Schumacher, Karl Aberer, Scott D Stoller, Samuel Ryu, and Fusheng Wang. ACTION-EHR: Patient-Centric Blockchain-Based Electronic Health Record Data Management for Cancer Care. *Journal of Medical Internet Research*, 22(8):e13598, 2020. ISSN 1439-4456. doi: 10.2196/13598.

[11] Gaby G. Dagher, Jordan Mohler, Matea Milojkovic, and Praneeth Babu Marella. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustainable Cities and Society*, 39:283–297, 2018. ISSN 2210-6707. doi: 10.1016/j.scs.2018.02.014.

[12] Patrick Li, Scott D Nelson, Bradley A Malin, and You Chen. DMMS: A Decentralized Blockchain Ledger for the Management of Medication Histories. *Blockchain in healthcare today*, 2:38, 2019. doi: 10.30953/bhty.v2.38.

[13] Minhua He, Xu Han, Frank Jiang, Rongbai Zhang, Xingzi Liu, and Xiao Liu. BlockMeds: A Blockchain-Based Online Prescription System with Privacy Protection. *Lecture Notes in Computer Science*, pages 299–303, 2020. ISSN 0302-9743. doi: 10.1007/978-3-030-45989-5_27.

[14] Peng Zhang, Breck Stodghill, Cory Pitt, Cavin Briody, Douglas C Schmidt, Jules White, Alan Pitt, and Kelly Aldrich. OpTrak: Tracking Opioid Prescriptions via Distributed Ledger Technology. *International Journal of Information Systems and Social Change*, 10(2):45–61, 2019. ISSN 1941-868X. doi: 10.4018/ijissc.2019040104.

[15] Camden Thatcher and Subrata Acharya. RxBlock: Towards the design of a distributed immutable electronic prescription system. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 9(1): 58, 2020. ISSN 2192-6662. doi: 10.1007/s13721-020-00264-5.

[16] May Alnafrani and Subrata Acharya. SecureRx: A blockchain-based framework for an electronic prescription system with opioids tracking. *Health Policy and Technology*, 10(2):100510, 2021. ISSN 2211-8837. doi: 10.1016/j.hlpt.2021.100510.

[17] Arun Sekar Rajasekaran, Maria Azees, and Fadi Al-Turjman. A comprehensive survey on blockchain technology. *Sustainable Energy Technologies and Assessments*, 52:102039, 2022. ISSN 2213-1388. doi: 10.1016/j.seta.2022.102039.

[18] Tharaka Hewa, Mika Ylianttila, and Madhusanka Liyanage. Survey on blockchain based smart contracts: Applications, opportunities and challenges. *Journal of Network and Computer Applications*, 177: 102857, 2021. ISSN 1084-8045. doi: 10.1016/j.jnca.2020.102857.

[19] Sonal Trivedi, Kiran Mehta, and Renuka Sharma. Systematic Literature Review on Application of Blockchain Technology in E-Finance and Financial Services. *Journal of technology management & innovation*, 16(3):89–102, 2021. doi: 10.4067/s0718-27242021000300089.

[20] Peter Gonczol, Panagiota Katsikouli, Lasse Herskind, and Nicola Dragoni. Blockchain Implementations and Use Cases for Supply Chains-A Survey. *IEEE Access*, 8:11856–11871, 2020. ISSN 2169-3536. doi: 10.1109/access.2020.2964880.

[21] Md.Ashraf Uddin, Andrew Stranieri, Iqbal Gondal, and Venki Balasubramanian. A Survey on the Adoption of Blockchain in IoT: Challenges and Solutions. *Blockchain: Research and Applications*, 2(2):100006, 2021. ISSN 2096-7209. doi: 10.1016/j.bcra.2021.100006.

[22] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, 2002. ISSN 0734-2071. doi: 10.1145/571637.571640.

[23] Daniel Cason, Enrique Fynn, Nenad Milosevic, Zarko Milosevic, Ethan Buchman, and Fernando Pedone. The design, architecture and performance of the Tendermint Blockchain Network. *2021 40th International Symposium on Reliable Distributed Systems (SRDS)*, 00:23–33, 2021. doi: 10.1109/srds53918.2021.00012.

[24] Ethan Buchman, Jae Kwon, and Zarko Milosevic. The latest gossip on BFT consensus. *arXiv*, 2018.

[25] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OsDI*, volume 99, pages 173–186, 1999.

[26] Hyperledger Besu. Besu enterprise ethereum client. https://besu.hyperledger.org/en/stable/, 2022. Accessed: 2022-02-27.

[27] Henrique Moniz. The Istanbul BFT Consensus Algorithm. *arXiv*, 2020.

[28] Akm Iqtidar Newaz, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. A Survey on Security and Privacy Issues in Modern Healthcare Systems: Attacks and Defenses. *ACM Transactions on Computing for Healthcare*, 2(3):1–44, 2021. ISSN 2691-1957. doi: 10.1145/3453176.

[29] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, nov 1979. ISSN 0001-0782. doi: 10.1145/359168.359176. URL https://doi.org/10.1145/359168.359176.

[30] CosmWasm. Cosmwasm testnets. https://github.com/CosmWasm/testnets, 2022. Accessed: 2022-02-26.

[31] Joshua Fernandes. Maximum validator count for an ibft2 network. https://wiki.hyperledger.org/display/BESU/Maximum+Validator+count+for+an+IBFT2+Network, 2022. Accessed: 2022-02-26.

[32] Bahareh Lashkari and Petr Musilek. A Comprehensive Review of Blockchain Consensus Mechanisms. *IEEE Access*, 9:43620–43652, 2021. ISSN 2169-3536. doi: 10.1109/access.2021.3065880.

[33] Vitalik Buterin. Ethereum white paper. https://github.com/ethereum/wiki/wiki/White-Paper. Accessed: 2022-02-26.

[34] Mohammad Dabbagh, Kim-Kwang Raymond Choo, Amin Beheshti, Mohammad Tahir, and Nader Sohrabi Safa. A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities. *Computers & Security*, 100:102078, 2021. ISSN 0167-4048. doi: 10.1016/j.cose.2020.102078.

[35] Muhammad Nasir Mumtaz Bhutta, Amir A. Khwaja, Adnan Nadeem, Hafiz Farooq Ahmad, Muhammad Khurram Khan, Moataz A. Hanif, Houbing Song, Majed Alshamari, and Yue Cao. A Survey on Blockchain Technology: Evolution, Architecture and Security. *IEEE Access*, 9:61048–61073, 2021. ISSN 2169-3536. doi: 10.1109/access.2021.3072849.

[36] Cosmos Ecosystem. Apps and services. https://cosmos.network/ecosystem/apps. Accessed: 2022-02-26.

[37] Renpeng Zou, Xixiang Lv, and Jingsong Zhao. SPChain: Blockchain-based medical data sharing and privacy-preserving eHealth system. *Information Processing & Management*, 58(4):102604, 2021. ISSN 0306-4573. doi: 10.1016/j.ipm.2021.102604.

[38] Weizhe Chen, Shunzhi Zhu, Jianmin Li, Jiaxin Wu, Chin-Ling Chen, and Yong-Yuan Deng. Authorized Shared Electronic Medical Record System with Proxy Re-Encryption and Blockchain Technology. *Sensors (Basel, Switzerland)*, 21(22):7765, 2021. doi: 10.3390/s21227765.

[39] Avery W Huang, Adharsh Kandula, and Xiaodi Wang. A Differential-Privacy-Based Blockchain Architecture to Secure and Store Electronic Health Records. *2021 The 3rd International Conference on Blockchain Technology*, pages 189–194, 2021. doi: 10.1145/3460537.3460555.

CHAPTER

4

A BLOCKCHAIN-BASED DATA GOVERNANCE WITH PRIVACY AND PROVENANCE: A CASE STUDY FOR E-PRESCRIPTION

This chapter presents the article published in the International Conference on Blockchain and Cryptocurrency (ICBC) under the following IEEE permission:

Contribution Statement: Software, Data curation, Writing – review & editing.

[1] <https://doi.org/10.1109/ICBC54727.2022.9805545>

A Blockchain-based Data Governance with Privacy and Provenance: a case study for e-Prescription

Rodrigo Dutra Garcia*, Gowri Sankar Ramachandran†, Raja Jurdak†, and Jo Ueyama*
**Institute of Mathematics and Computer Science, University of São Paulo, Brazil. {rgarcia,joueyama@icmc.}@usp.br*
†*School of Computer Science, Queensland University of Technology, Australia. {g.ramachandran,r.jurdak}@qut.edu.au*

***Abstract*—Real-world applications in healthcare and supply chain domains produce, exchange, and share data in a multi-stakeholder environment. Data owners want to control their data and privacy in such settings. On the other hand, data consumers demand methods to understand when, how, and who produced the data. These requirements necessitate data governance frameworks that guarantee data provenance, privacy protection, and consent management. We introduce a decentralized data governance framework based on blockchain technology and proxy re-encryption to let data owners control and track their data through privacy-enhancing and consent management mechanisms. Besides, our framework allows the data consumers to understand data lineage through a blockchain-based provenance mechanism. We have used Digital e-prescription as the use case since it has multiple stakeholders and sensitive data while enabling the medical fraternity to manage patients' prescription data, involving patients as data owners, doctors, and pharmacists as data consumers. Our proof-of-concept implementation and evaluation results based on CosmWasm and pyUmbral PRE show that the proposed decentralized system guarantees transparency, privacy, and trust with minimal overhead.**

***Index Terms*—Data Governance, Decentralized, E-prescription, Privacy, Blockchain, Smart Contracts, Proxy Re-encryption**

I. Introduction

Prescription systems allow healthcare professionals, such as physicians, to create digital records about a patient's health status by adding diagnosis and medications data. It allows for more efficient communication and reduced inconsistencies compared to paper-based prescriptions [1, 2]. Thus, digital prescription systems have increased globally, enabling multiple stakeholders, including doctors and pharmacies, to effectively access and manage patients' data.

Patients want to control their data and privacy in healthcare settings since prescription and diagnosis data contain sensitive and personally identifiable information. Note that unauthorized parties may gain access and misuse patients' data [3]. Therefore, it is essential to *protect patients' privacy while letting them manage and permit access to their data transparently*, which is one of the problems this paper aims to investigate.

Pharmacies must sell certain drugs such as antibiotics with a valid doctor's prescription. A prescription containing an antibiotic medicine is valid for only a single purchase, meaning the pharmacy and the patient must obey the recommended dosages. However, pharmacies tend to sell medications illegally to patients to gain financial revenue, even with the old and used prescription. Such illegal sales would lead to unwanted side effects, including drug abuse and overdoses [4, 5], burdening the healthcare system. Therefore, it is essential to regulate the medicine supply chain to prevent the unauthorized sales of medications, which is one of the focuses of this work.

978-1-6654-9538-7/22/$31.00 ©2022 IEEE

Existing digital prescription systems primarily employ a centralized architecture, offering limited to no visibility into the operations providing maximum power to the administrating organization [1, 2]. Such centralized architectures are susceptible to single points of failure, enabling opportunities for data tampering. In addition, centralized systems may also misuse patients' health data without their consent, resulting in privacy violations. In summary, centralized architectures offer no transparency undermining the integrity of medical information while affecting patients' privacy [6]. We, therefore, argue that a decentralized architecture with support for consent management, privacy preservation, and data provenance is essential for a trusted digital prescription system.

Existing works in digital e-prescription do not securely manage consent while providing support for privacy protection and accountability [7, 8, 9]. We propose a decentralized data governance framework for the electronic prescription that:

- Helps patients *store, manage, and share* prescription data with other stakeholders through a tamper-proof ledger.
- Protects patients' *privacy* by storing encrypted prescription data on the blockchain ledger to withhold personally identifiable and sensitive information from third parties, including drug regulators.
- Provides support for *consent management* using proxy re-encryption scheme and smart contracts.
- Supports *data provenance* to let data owners and data consumers efficiently monitor the historical records of the data and its origin, including who accessed the data and for what purposes.
- Enables the drug regulators to control and monitor the flow of medications to the pharmacies through the *accountable* blockchain ledger, thereby limiting illegal sales.

We have developed a proof-of-concept implementation using the CosmWasm, which uses Tendermint (a Byzantine Fault Tolerance (BFT) consensus mechanism) and NuCypher pyUmbral [10] proxy re-encryption (PRE) library to estimate the overhead and feasibility. Our evaluation results show that the proposed data governance framework introduces minimal overhead while letting data owners control and manage their data with transparency and trust guarantees. Although we discuss the data governance framework through an e-prescription use case, the proposed framework is suitable for any multi-stakeholder application, including supply chain management, dealing with digital and sensitive data.

II. Related Work

Electronic prescription systems operate in a multi-stakeholder environment. It requires the integrity and transparency of information to avoid illegal drug sales while preventing patients' health problems due to drug overdose. Besides, the application of privacy-preserving techniques for medical records is another requirement to avoid the misuse of sensitive information present in prescriptions. Alnafrani and Acharya proposed SecureRx [7], a blockchain solution using the Ethereum platform to maintain patient records and prescriptions. Garcia et al. [8] proposed a decentralized e-prescription system using smart-contracts on a BFT platform. However, these solutions do not *manage consent* and focus on writing records to an immutable ledger without providing mechanisms to *track who accessed the data and for what purposes* while *protecting patient's sensitive information.*

Other research works investigate approaches to ensure the integrity and privacy of medical records by preventing tampering and data leakage. Zou et al. proposes SPchain [11], a blockchain and PRE-based solution for sharing electronic health records (EHR). Li et al. introduced DMMS [9], a solution that exploits blockchain technology for medication history management and electronic prescriptions. Bhaskaran et al. [12] proposed a solution to store consumers' data in encrypted form on blockchain. Entities that wish to get access can raise consent requests on the chain, and the data owners can provide such consent cryptographically. However, the works above do not manage patient consent for sharing sensitive data between multi-stakeholder applications using the PRE mechanism in the electronic prescriptions use case.

III. Background on Proxy Re-Encryption (PRE)

Proxy re-encryption is an asymmetric encryption technique initially proposed by Blaze et al. [13] in which an entity A (delegator) can delegate the decryption rights to another entity B (delegatee) through a proxy server.

Initially, a message m is encrypted using the delegator's public key, $C_A = Enc(pk_A, m)$, and stored in a database. If delegatee B needs to decrypt the message, he must initially request decryption rights for the delegator, informing his public key pk_B. If the delegator agrees, it will produce a delegation key $rk_{A \to B}$ and send it to the proxy.

For delegatee B to be able to decrypt the information, the proxy server must use the delegation key $rk_{A \to B}$ to re-encrypt C_A, that is, $C_B = ReEnc(rk_{A \to B}, C_A)$. After re-encryption, the delegatee can use his private (i.e., secret) key sk_B and decrypt the message. At all stages, only the public key is shared between the participants. From the proxy's point of view, it does not learn or try to decrypt confidential information. It receives encrypted information C_A and sends other encrypted information C_B.

IV. Decentralized Architecture with Consent Management and Privacy Protection

A. System Model and Threats

We assume a system comprising of patients, doctors, and pharmacies. When a patient visits a doctor, the doctor creates a new medical record that includes diagnostic data, personal details such as name and age, and prescriptions.

We focus on the following threats:

- Privacy threat: The patient's medical record includes sensitive data, which should not be revealed to unauthorized third parties without the patient's consent.
- Illegal drug sales: The lack of visibility into the medication supply chain leads to illegal medication sales, resulting in drug overdoses.

Given these threats, this work aims to develop a solution with the following objectives:

Objective 1: We aim to develop a transparent medical prescription system based on the blockchain without revealing sensitive data to unauthorized third parties. Note that the data stored on the blockchain is visible to the public on a blockchain platform. *Can we allow the patients to store and manage medical data in a tamper-proof ledger without violating patients' privacy?*

Objective 2: When the data get stored on a digital system, doctors and health care agencies can access the data for diagnostic and survey purposes. Under this circumstance, it is important to let patients or data owners have visibility into data usage. *Can we allow the data owners to track and govern data usage by other parties?*

B. Proposed Solution

We propose a decentralized data governance framework using blockchain technology and smart contracts to help patients manage their data more efficiently. When a patient consults a doctor, the doctor creates prescription data by recording the diagnosis, recommended medications, and dosage. Then, the prescription data is encrypted using the patient's public key and stored in the blockchain via the smart contract. We assume that the patient shares her public key with the doctor when she makes an appointment to see the doctor. The patient needs to allow the pharmacists access to the prescription data to receive medication from the pharmacy. We propose a data access tracking mechanism within the contract state to monitor data accesses. In this way, any query or update in the status of the records will be registered. Note that the existing blockchain-based systems support writing data to an immutable ledger. Still, they do not monitor or provide support for governing data usage, which is necessary for data provenance and privacy. Our framework not only records the data on an immutable ledger in a privacy-preserving manner but also logs access requests to govern data usage.

To prevent illegal medications sales by the pharmacy, the regulatory agency will count, through a control contract, the number of drugs supplied to the pharmacy with the number of drugs sold (in the sales contract). We assume that the blockchain can hold encrypted prescription data for brevity. Patients can report the pharmacy that sells unlawful drugs and receive rewards (tokens) through a smart contract. We can extend the framework by storing the encrypted prescription data on off-chain storage while maintaining the hash on-chain, which we plan to tackle in our future work.

C. Proxy Re-Encryption Mechanism

We use the proxy re-encryption technique to ensure data privacy in this work. In this way, stakeholders can decrypt prescription data only with the patient's consent via the delegation mechanism. The diagram in Figure 1 shows the architecture with the PRE operations:

1) From appointment with the patient, the doctor creates a prescription containing the items: personal information (*PI*), medication (*MED*), and diagnosis (*DIA*) for future analysis. Before sending the prescription to the $create\_prescription$ smart contract method and being stored in the contract state, the prescription items are encrypted by the doctor application separately using the patient's public key (pk_P). *Therefore, the patient has flexibility and can consent to data sharing.*
2) For the doctor, pharmacy, or regulator to analyze any item in the prescription, it will be necessary to request decryption rights from the patient, informing their respective public key (pk).
3) If the patient agrees with the request, the patient's application will generate a delegation key. The delegation key will be encrypted and sent to the *set_consent* contract method.
4) In the stakeholder application, the proxy will perform the re-encryption (RE) operation using the respective delegation key and the allowed prescription item. The doctor has access to all the prescription data. For this, the proxy will perform the re-encryption for personal information C_{PI}, medication C_{MED} and diagnosis C_{DIA}. The pharmacy and the regulator can only re-encrypt the prescribed medication.
5) After the re-encryption step, stakeholders can decrypt the item with their respective private key (s_k) and analyze the information allowed by the patient.

Note that sensitive prescription items are encrypted before being stored in the blockchain. In this way, records are private and immutable. Other organizations will only be able to decrypt the information with the patient's permission. The proxy re-encryption mechanism is a privacy software module implemented in the stakeholder application to act on confidential data sharing operations.

Note about proxy: a proxy is a software that only re-encrypts information. The proxies do not store any private keys and do not see any message from the ciphertexts. From their perspective, they only see an incoming ciphertext and the result after re-encryption, which is also a ciphertext.

Consent mechanism and delegation key: Figure 2 represents steps 2 and 3 of Figure 1 where each stakeholder is a full node (i.e., containing the PRE operations and blockchain). In step 2, the stakeholder sends a request to the patient through a consent contract. In step 3, the patient will create and encrypt the delegation key using the stakeholder's public key. In this way, requests are transparent to all network participants, and only the stakeholder can decrypt the delegation key.

D. How does the proposed solution meet the objectives?

Objective 1 focuses on providing transparency to the prescription system while protecting patients' privacy — our solution stores the patients' data on the blockchain but in an encrypted form. The prescription data is made available to other parties after the patient's consent.

Objective 2 focuses on governing data usage - our solution tracks the data access requests of consumers and permissions of data owners through a smart contract and immutable ledger. Therefore, data owners can have visibility into their data and its usage. We understand that a malicious data consumer may access the patient's data with their permission and then post it on a black market or other digital platforms. We plan to investigate digital watermarking and steganography in our future work to overcome this problem [14].

V. Proof-of-Concept Implementation and Evaluation

A. Privacy: Proxy Re-Encryption

Evaluation Goals: To understand the overhead and feasibility of PRE operations, we evaluate execution time tests for steps in Figure 1. We evaluated encrypting (step 1), creating a delegation key (step 3), re-encryption (step 4), and decrypting (step 5).

Evaluation Setup and Methodology: The PRE evaluation programs and scripts were implemented in Python programming language using NuCypher pyUmbral PRE technology, an open-source implementation that uses the *secp256k1* elliptic curve [15]. We use the *time* module to calculate the difference between the start and end of each operation. All software created for evaluation is available on GitHub [16].

To identify realistic file sizes for medications and dosage prescriptions, we used the English Prescribing Dataset [17]. Prescription items used for evaluation are represented in separate text files with different sizes ranging from 0.43 Kilobyte (kB) to 0.82 kB for personal information, 0.24 kB to 0.53 kB for medication, and 2.18 kB to 8975.74 kB ≈ 8.76 Megabyte (MB) for diagnosis. While the file sizes are inferred from [17], file contents are randomly generated by the evaluation software. In total, 1000 iterations were performed for different file sizes. We used a Linux virtual machine with an Intel Core i7-10510U 1.80GHz (Dual-Core) processor and 6GB of RAM for the evaluation.

Execution Time Evaluation: Figure 3 shows the average execution time for application-level PRE operations using the data files.

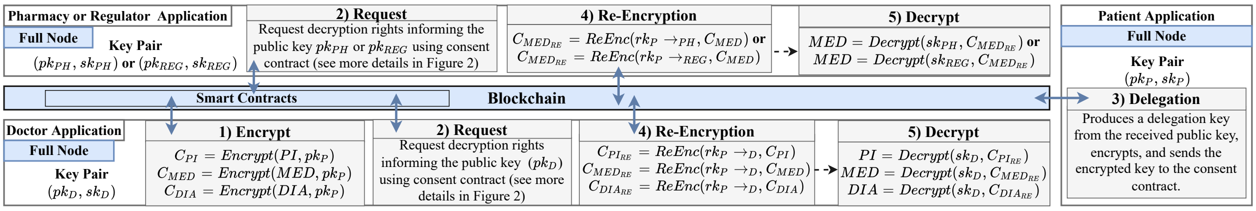
Pharmacy or Regulator Application
Full Node
Key Pair
(pk_{PH}, sk_{PH}) or (pk_{REG}, sk_{REG})
2) Request
Request decryption rights informing the public key pk_{PH} or pk_{REG} using consent contract (see more details in Figure 2)
4) Re-Encryption
$C_{MED_{RE}} = ReEnc(rk_P \rightarrow_{PH}, C_{MED})$ or
$C_{MED_{RE}} = ReEnc(rk_P \rightarrow_{REG}, C_{MED})$
5) Decrypt
$MED = Decrypt(sk_{PH}, C_{MED_{RE}})$ or
$MED = Decrypt(sk_{REG}, C_{MED_{RE}})$
Patient Application
Full Node
Key Pair
(pk_P, sk_P)
Smart Contracts
Blockchain
3) Delegation
Produces a delegation key from the received public key, encrypts, and sends the encrypted key to the consent contract.
Doctor Application
Full Node
Key Pair
(pk_D, sk_D)
1) Encrypt
$C_{PI} = Encrypt(PI, pk_P)$
$C_{MED} = Encrypt(MED, pk_P)$
$C_{DIA} = Encrypt(DIA, pk_P)$
2) Request
Request decryption rights informing the public key (pk_D) using consent contract (see more details in Figure 2)
4) Re-Encryption
$C_{PI_{RE}} = ReEnc(rk_P \rightarrow_D, C_{PI})$
$C_{MED_{RE}} = ReEnc(rk_P \rightarrow_D, C_{MED})$
$C_{DIA_{RE}} = ReEnc(rk_P \rightarrow_D, C_{DIA})$
5) Decrypt
$PI = Decrypt(sk_D, C_{PI_{RE}})$
$MED = Decrypt(sk_D, C_{MED_{RE}})$
$DIA = Decrypt(sk_D, C_{DIA_{RE}})$

Fig. 1. Our decentralized data governance framework with support for PRE mechanism, consent management, data provenance, and privacy.

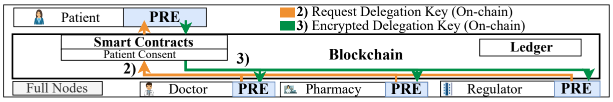

Fig. 2. On-chain request mechanism (step 2) and sending the encrypted delegation key (step 3) using smart contracts

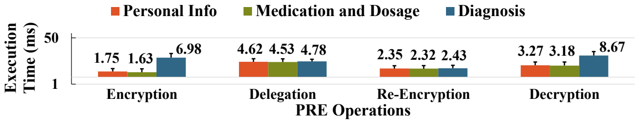

Fig. 3. Average execution time in PRE operations

To encrypt the diagnostic data (step 1), it took an average of 6.98 milliseconds (ms), while medication and personal information data took an average of 1.63 ms and 1.75 ms, respectively. There were slight variations in the average processing times in the delegation and re-encryption stage for the prescription items. The delegation stage (step 3) took an average of around 4 ms, while in the re-encryption operation (step 4), the average execution time was around 2 ms for all prescription data. In the decrypt stage (step 5), the item that obtained the highest execution average was the diagnosis with 8.67 ms. In comparison, medication and the patient's personal information took an average of around 3 ms.

These results show PRE operations' execution time cost is relative to the data size. In our evaluation, even with text files with sizes in Megabyte, the operations did not exceed 50 ms to be executed. In this sense, our proposed framework protects the privacy and manages consent with a low operational overhead. We also believe PRE operations can run on platforms like Raspberry Pi or mobile phones.

B. Smart Contract: CosmWasm Implementation

A test network called *Uni Junø* network [18] with 30 validator nodes was used to evaluate the transaction time of the encrypted prescription items (after encryption step in Figure 1). The steps to automate the sending of transactions to the network were implemented in a shell script. All software and contracts developed for model evaluation are available on GitHub [16].

Transaction time for Smart Contracts in CosmWasm: Table I shows the transaction validation time for each prescription ranging from 0.92 kB to 130.50 kB containing all items (i.e., patient's personal information, medication, and diagnosis). Time refers to the Tendermint consensus process with inclusion in a block. On average, the time for a transaction to be validated by contract method took 2.69 seconds, with the maximum and minimum time being 6.26 seconds and 1.50 seconds, respectively. The variation in transaction time is due to the consensus delay, including peer-to-peer messaging between validator nodes.

TABLE I
TRANSACTION VALIDATION TIME FOR THE CREATE_PRESCRIPTION CONTRACT METHOD USING COSMWASM IMPLEMENTATION WITH *Uni Junø* TESTNET

	Transaction time (in seconds)			
Number of Transactions	**Max.**	**Min.**	**Avg.**	**Std.**
300	6.26	1.50	2.69	0.71

VI. CONCLUSIONS

Real-world multi-stakeholder applications such as e-prescription and supply chain deal with digital and sensitive data, demanding privacy protection, consent management, data provenance, and transparency. We have presented a decentralized data governance framework for e-prescription that uses proxy re-encryption and smart contracts to let data owners control and manage their data through a trusted and transparent blockchain platform. We have shown how the data owners can record all the access requests and consents in an immutable ledger to monitor data lineage. Our proof-of-concept implementation uses CosmWasm and pyUmbral proxy re-encryption library to assess the feasibility and performance. Our evaluation results show that the proposed architecture can protect data owners' privacy and govern sensitive data access with minimal overhead. We believe that our data governance framework is beneficial to all multi-stakeholder applications that deal with sensitive and private digital data.

ACKNOWLEDGEMENT

Jo Ueyama would like to thank the Brazilian Research Council (CNPq) and São Paulo Research Foundation (FAPESP) for financing the bulk of his research, FAPESP grant ID 2013/07375-0. Rodrigo D. Garcia would like to acknowledge the financial support by the Brazilian Research Council (CNPq, process 133470/2020-2) for his research.

REFERENCES

[1] Bader Aldughayfiq and Srinivas Sampalli. Digital health in physicians' and pharmacists' office: A comparative study of e-prescription systems' architecture and digital security in eight countries. *OMICS: A Journal of Integrative Biology*, 25(2):102–122, 2021.

[2] Mahnaz Samadbeik, Maryam Ahmadi, Farahnaz Sadoughi, and Ali Garavand. A copmarative review of electronic prescription systems: Lessons learned from developed countries. *Journal of research in pharmacy practice*, 6(1):3, 2017.

[3] Shunrong Jiang, Haiqin Wu, and Liangmin Wang. Patients-controlled secure and privacy-preserving ehrs sharing scheme based on consortium blockchain. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. doi: 10.1109/GLOBECOM38437.2019.9013220.

[4] Phil Skolnick. The opioid epidemic: Crisis and solutions. *Annual Review of Pharmacology and Toxicology*, 58(1): 143–159, 2018. doi: 10.1146/annurev-pharmtox-010617-052534. URL https://doi.org/10.1146/annurev-pharmtox-010617-052534. PMID: 28968188.

[5] Rachel W. Faller, Jennifer Toller Erausquin, and Thomas P. McCoy. Misuse of prescription and illicit drugs in middle adulthood in the context of the opioid epidemic. *Substance Use & Misuse*, 56(2):333–337, 2021. doi: 10.1080/10826084.2020.1858107. URL https://doi.org/10.1080/10826084.2020.1858107. PMID: 33325317.

[6] Ehab Zaghloul, Tongtong Li, and Jian Ren. Security and privacy of electronic health records: Decentralized and hierarchical data sharing using smart contracts. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 375–379, 2019. doi: 10.1109/ICCNC.2019.8685552.

[7] May Alnafrani and Subrata Acharya. Securerx: A blockchain-based framework for an electronic prescription system with opioids tracking. *Health Policy and Technology*, 10(2):100510, 2021. ISSN 2211-8837. doi: https://doi.org/10.1016/j.hlpt.2021.100510. URL https://www.sciencedirect.com/science/article/pii/S2211883721000332.

[8] Rodrigo Dutra Garcia, Gabriel Augusto Zutião, Gowri Ramachandran, and Jo Ueyama. Towards a decentralized e-prescription system using smart contracts. In *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 556–561, 2021. doi: 10.1109/CBMS52027.2021.00037.

[9] Patrick Li, Scott D. Nelson, Bradley A. Malin, and You Chen. Dmms: A decentralized blockchain ledger for the management of medication histories. *Blockchain in Healthcare Today*, 2, Jan. 2019. doi: 10.30953/bhty.v2.38. URL https://blockchainhealthcaretoday.com/index.php/journal/article/view/38.

[10] DAVID Nunez. Umbral: a threshold proxy re-encryption scheme. *NuCypher Inc and NICS Lab, University of Malaga, Spain*, 2018.

[11] Renpeng Zou, Xixiang Lv, and Jingsong Zhao. Spchain: Blockchain-based medical data sharing and privacy-preserving ehealth system. *Information Processing & Management*, 58(4):102604, 2021.

[12] Kumar Bhaskaran, Peter Ilfrich, Dain Liffman, Christian Vecchiola, Praveen Jayachandran, Apurva Kumar, Fabian Lim, Karthik Nandakumar, Zhengquan Qin, Venkatraman Ramakrishna, Ernie GS Teo, and Chun Hui Suen. Double-Blind Consent-Driven Data Sharing on Blockchain. *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pages 385–391, 2018. doi: 10.1109/ic2e.2018.00073.

[13] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 127–144, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-69795-4.

[14] Sijia Zhao and Donal O'Mahony. Bmcprotector: A blockchain and smart contract based application for music copyright protection. In *Proceedings of the 2018 International Conference on Blockchain Technology and Application*, pages 1–5, 2018.

[15] NuCypher. pyumbral pre. https://github.com/nucypher/pyUmbral, 2021.

[16] R.D Garcia. E-prescription model. https://github.com/rodrigodg1/e-prescription, 2021.

[17] English Prescribing Dataset. English prescribing dataset (epd). https://opendata.nhsbsa.net/dataset/english-prescribing-data-epd, 2021.

[18] CosmWasm. Cosmwasm testnets. https://github.com/CosmWasm/testnets, 2021.

CHAPTER

5

BLOCKCHAIN-AIDED AND PRIVACY-PRESERVING DATA GOVERNANCE IN MULTI-STAKEHOLDER APPLICATIONS

This chapter presents the article published in the Transactions on Network and Service Management under the following IEEE permission:

© 2022 IEEE. Reprinted, with permission, from R. D. Garcia, G. S. Ramachandran, R. Jurdak and J. Ueyama, "Blockchain-Aided and Privacy-Preserving Data Governance in Multi-Stakeholder Applications" in IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 3781-3793, Dec. 2022, doi: 10.1109/TNSM.2022.3225254[1]

Contribution Statement: Software, Data curation, Writing – review & editing.

[1] <https://doi.org/10.1109/TNSM.2022.3225254>

Blockchain-aided and Privacy-preserving Data Governance in Multi-stakeholder Applications

Rodrigo Dutra Garcia*, Gowri Sankar Ramachandran†, Raja Jurdak†, and Jo Ueyama*
**Institute of Mathematics and Computer Science, University of São Paulo, Brazil. {rgarcia,joueyama@icmc.}@usp.br*
†*School of Computer Science, Queensland University of Technology, Australia. {g.ramachandran,r.jurdak}@qut.edu.au*

***Abstract*—Real-world applications in healthcare and supply chain domains produce, exchange, and share data in a multi-stakeholder environment. Data owners want to control their data and privacy in such settings. On the other hand, data consumers demand methods to understand when, how, and who produced the data. These requirements necessitate data governance frameworks that guarantee data provenance, privacy protection, consent management, and selective disclosure. We introduce a decentralized data governance framework based on blockchain technology, proxy re-encryption, and Boneh, Boyen, and Shacham (BBS) signatures to let data owners control, selectively share and track their data through privacy-enhancing, consent management, and selective disclosure mechanisms. Besides, our framework allows the data consumers to understand data lineage through a blockchain-based provenance mechanism. We use Digital medical e-prescription as the use case since it handles sensitive data in a multi-stakeholder environment while showing how the medical community can manage patients' sensitive prescription data, involving patients as data owners, and doctors, and pharmacists as data consumers. Our proof-of-concept implementation and evaluation results based on CosmWasm, Hyperledger Besu, Ethereum, pyUmbral PRE, and BBS signatures show that the proposed decentralized system is platform-agnostic, scalable and guarantees a higher degree of transparency, privacy, and trust with minimal overhead.**

***Index Terms*—Data Governance, Decentralized, E-prescription, Privacy, Blockchain, Smart Contracts, Proxy Re-encryption, Selective Sharing**

I. Introduction

Real-world applications in e-prescription, supply chain, distributed energy trading, and data marketplaces operate in a multi-stakeholder environment. Data producers may share data with one or more data consumers through a middleware. On the one hand, data producers, who are also data owners in some cases, want to learn who accesses their data and for what purposes while demanding privacy to protect their sensitive information. Nevertheless, data consumers require information about the data and its journey while requesting other stakeholders to share their data to maximize the operational efficiency of their businesses. Middleware platforms often connect the data producers with data consumers in multi-stakeholder applications.

Electronic prescribing systems is an example of a multi-stakeholder application that handles and stores sensitive medical records about the patient and share such information with other healthcare organizations such as pharmacies and hospitals [1, 2]. These systems must ensure data privacy to prevent misuse of information, security to prevent data tampering, and patient consent to share records with other stakeholders [3]. Healthcare professionals like doctors create patient records containing personally identifiable information (PII), medications, and diagnosis. These records must be securely stored for future analysis, such as patients revisiting the doctor or sharing with other stakeholders. Therefore, in the rest of this article, we use e-prescription as a representative multi-stakeholder application to illustrate the data governance requirements.

Relying on a centralized middleware for data sharing introduces the central point of failure as the organization that runs the middleware may misuse sensitive data belonging to data producers. For example, prescription systems allow healthcare professionals, such as physicians, to create digital records about a patient's health status by adding diagnosis and medications data. Allowing a centralized middleware to manage patients' health data is risky since the prescription and diagnosis data contain sensitive and personally identifiable information. Note that centralized systems may misuse patients' health data without their consent, resulting in privacy violations [4, 5]. This can be crucial in the face of the privacy laws, such as the EU GDPR (General Data Protection Regulation) policy.

Blockchain offers decentralized middleware for multi-stakeholder applications. Data producers can store and share data through the blockchain ledger. Still, it is not tailor-made for storing and sharing sensitive data, as the data has to be packaged appropriately and encrypted before storage. Several existing frameworks contribute approaches to sharing data through the blockchain [6, 7, 8, 9]. Still, they do not adequately discuss how the data producers can track their data, manage access, and control it throughout the life cycle. Similarly, existing literature lacks approaches to access data with owners' consent within a decentralized and blockchain-based ecosystem.

Data includes multiple attributes with varying sensitivity. For example, a doctor could look at a patient's historical health record, age, and prescription data, while the pharmacist could only access the prescription data. Only the data associated with the attribute "prescription" must be revealed to the pharmacist in this context. When leveraging blockchain for data sharing, existing approaches allow sharing of all attributes while lacking efficient mechanisms for selective sharing of attributes based on the data consumer and their authority with the data owner's consent.

In this work, we show how proxy re-encryption can be employed in a blockchain-based multi-stakeholder application to protect the privacy of the data owner while allowing the data owner to control and manage their data [10]. Our prior work [11] introduced only the architecture to share sensitive data between stakeholders in e-prescription applications. It did not use selective disclosure mechanism to allow the data owner to selectively share data with other stakeholders. We make the following contributions:

- Introduce a novel decentralized architecture for multi-stakeholder applications by combining blockchain, smart contracts, proxy re-encryption, and BBS signature.
- Employ proxy re-encryption to allow data owners to view data access requests and delegate data accesses, giving consent to the data consumers through an immutable ledger.
- Leverage BBS signature to share data attributes selectively with data consumers based on the authority and pre-established trust, protecting the privacy.
- Implement a proof of concept using CosmWasm, which uses Tendermint (a Byzantine Fault Tolerance–BFT consensus mechanism), Ethereum (a popular blockchain platform with smart contract support), and NuCypher pyUmbral [12] proxy re-encryption (PRE) library to estimate the performance and to demonstrate feasibility. Also, we compare the obtained results with distinct blockchain implementations such as Tendermint, Hyperledger Besu, and Ethereum.

Our evaluation results show that the proposed data governance framework introduces minimal overhead while letting data owners control and manage their data with transparency and trust guarantees. Although we discuss the data governance framework through an e-prescription use case, the proposed framework is suitable for any multi-stakeholder application, including supply chain management, dealing with digital and sensitive data.

II. Related Work

Electronic prescription systems operate in a multi-stakeholder environment. It requires the integrity and transparency of information to avoid illegal drug sales while preventing patients' health problems due to drug overdose. Besides, the application of privacy-preserving techniques for medical records is another requirement to avoid misuse of sensitive information present in prescriptions.

A. Blockchain-based data sharing

Some works explore blockchain technology to maintain data integrity and availability for sharing among multiple stakeholders. Dubovitskaya et al. proposes Action EHR [7], a solution based on blockchain and asymmetric cryptography to allow patients to manage records related to cancer treatment across multiple hospitals. The authors explored records management in a hybrid way where metadata such as file sharing permissions are stored on the blockchain, and patient data is encrypted and stored off-chain. Makhdoom et al. [6] proposed a framework to preserve privacy in data sharing in the smart cities use case. The authors used the Hyperledger Fabric permissioned platform using different channels to separate communication between organizations while preserving data privacy. Rajput et al. [13] presented a framework for managing personal health records (PHR) using the Hyperledger Fabric and Hyperledger Composer platforms. The patient defines the access rules to share the PHR through smart contracts in this framework.

B. Consent management in blockchain-based data sharing

The consent mechanism allows controlling access to records. This way, information can only be accessed with the data owner's consent. Kim et al. proposed DynamiChain [14], a dynamic consent system for sharing data between different organizations. In particular, the authors analyzed the sharing of medical examinations. In the proposed model, data utilizers can access data according to rules defined by data providers using blockchain technology. Similarly, Tith et al. [15] presented a prototype to allow patient consent to share their medical data with other healthcare organizations. The authors presented the model using the Hyperledger Fabric platform in which the patient can define sharing rules. Jaiman and Urovi [8], Albanese et al. [9] and Hu et al. [16] proposed a consent model for the patient to control their data sharing with other stakeholders.

C. Privacy in blockchain-based data sharing

The application of privacy-preserving techniques for medical records is another requirement to avoid the misuse of sensitive information present in prescriptions. Chen et al. [17] proposed a solution using proxy re-encryption to preserve privacy when sharing medical data. The authors used a hybrid approach in which medical data is stored in the cloud in an encrypted form, and metadata such as indexes and digital signatures are stored on the blockchain. Similarly, Zou et al. proposed SPChain [18] a framework for sharing medical data while preserving patient privacy through proxy re-encryption. Li et al. introduced the decentralized medication management system (DMMS) [19] to allow prescribers to create prescriptions and encrypt data using the patient's public key.

Manzoor et al. [20] proposed a scheme to manage Internet of Things (IoT) data sharing. In particular, the authors used proxy re-encryption to ensure confidentiality in distributed cloud storage. In terms of evaluation, the authors analyzed the system performance using the Ethereum platform. Similarly, Chen et al. [21] proposed an architecture with proxy re-encryption for secure IoT data sharing. From the evaluation point of view, the authors analyzed the operational costs without a specific blockchain platform.

D. Selective sharing in blockchain-based applications

The data selection mechanism allows the data owner to select the data to be shared without disclosing other information. Mukta et al. proposed CredChain [22] a blockchain-based solution that enables the creation, sharing and verification

TABLE I
RELATED WORK AND GAPS

Features	**Related Work**					**This work**
	[6, 7, 13]	[14]	[8, 9, 15, 16]	[17, 18, 19, 20, 21]	[22]	
Blockchain-based	Y	Y	Y	Y	Y	**Y**
Privacy Support	Y	Y	N	Y	Y	**Y**
Consent Mechanism	N	Y	Y	N	N	**Y**
Selective Disclosure	N	N	N	N	Y	**Y**
Tracking Mechanism	N	N	N	N	N	**Y**

of credentials. In particular, the authors explore academic credentials to allow flexibility in selective disclosure using redactable signatures. The user has full control over the credential data and can select the information to be shared with other institutions.

However, none of the above works allows the data owner to manage and select specific data to share with other stakeholders while maintaining the privacy of sensitive data. Furthermore, our solution allows the data owner to verify who uses their data through a tracking mechanism. Table I summarizes related work and gaps.

III. Background

This section introduces the technologies we use to preserve privacy while allowing the data owner to select the data to be shared among multiple stakeholders. In particular, we use proxy re-encryption to keep the data encrypted and only accessible through the data owner's consent and BBS signatures to select data to be shared with other stakeholders, maintaining the authenticity of records.

A. Proxy Re-Encryption (PRE)

Proxy re-encryption is an asymmetric encryption technique initially proposed by Blaze et al. [23] in which an entity A (delegator) can delegate the decryption rights to another entity B (delegatee) through a proxy server. In the PRE technique, we have three main actors:

- **Delegator**: data owner and delegates decryption rights to another user or entity (i.e., delegatee);
- **Delegatee**: data consumer and receive decryption rights to access the encrypted information;
- **Proxy**: performs a re-encryption using the delegation key (generated by the delegator) and the encrypted message to allow the delegatee to access the encrypted information;

Initially, a message m is encrypted using the delegator's public key, $C_A = Enc(pk_A, m)$, and stored in a database. If delegatee B needs to decrypt the message, he must initially request decryption rights for the delegator informing his public key pk_B. If the delegator agrees, it will produce a delegation key $rk_{A\rightarrow B}$ and send it to the proxy.

For delegatee B to be able to decrypt the information, the proxy server must use the delegation key $rk_{A\rightarrow B}$ to re-encrypt C_A, that is, $C_B = ReEnc(rk_{A\rightarrow B}, C_A)$. After re-encryption, the delegatee can use his private (i.e., secret) key sk_B and decrypt the message. From Blaze et al. [23] work, all steps are summarized by the expression to decrypt a message m: $Dec(ReEnc(rk_{A\rightarrow B}, Enc(pk_A, m)), sk_B) = m$.

At all stages, only the public key is shared between the participants. From the proxy's point of view, it does not learn or try to decrypt confidential information. It receives encrypted information C_A and sends other encrypted information C_B.

Our solution uses NuCypher Umbral PRE cryptosystem algorithm that introduces the capsules (CAP) implementation feature [12, 24]. Umbral uses a key-encapsulation-mechanism (KEM) and data-encapsulation-mechanism (DEM). Data is encrypted with a random symmetric key, so this is the encrypted bulk data (the DEM part), and that symmetric key is then encrypted, which is the capsule (the KEM part). Both the encrypted bulk data and the capsule are stored together. During all steps, the capsule must be kept securely in an encrypted form to prevent a stakeholder, even with the delegation key, from re-encrypting other unauthorized data.

In summary, proxy re-encryption allows the data owner to delegate part of the secure data sharing responsibility to a third party (denoted as a proxy in PRE) while getting data integrity and confidentiality guarantees. In our work, we use PRE to allow the patient to delegate access to encrypted prescription data.

B. BBS Signatures

Unlike traditional digital signature systems where for validation, the messages and signature must be in the entire form, BBS signatures initially presented by Boneh et al. [25] is a form of short group signature that allows a stakeholder A to sign multiple messages generating a single signature s as output. With the signature, stakeholder B can derive proofs $(proof_i)$ by selecting the messages to be disclosed while maintaining the verifiable properties of authenticity and integrity. Another stakeholder (C) can verify the authenticity of derived proof $(proof_i)$.

Decentralized Identity Foundation [26] provides a formal definition of BBS operations adopted by our work. From the private (sk) and public key (pk), a Stakeholder A can sign a set of messages (msg) producing a single signature s as output. That is, $s = Sign(sk, pk, (msg[0], ..., msg[n]))$. With the signature s, Stakeholder B can produce derived proofs $(proof_i)$ using stakeholder A's public key with the messages indexes to be revealed: $proof_i = ProofGen(pk, s, (msg[0], ..., msg[n]), indexes)$.

The new transaction with the revealed message has zero-knowledge proof to verify the existence of the signature and Stakeholder C can verify if the derived proof is valid for the signature created by the first stakeholder: $ProofVerify(pk, proof_i, msg)$. In the electronic prescriptions use case, the doctor is stakeholder A, the patient stakeholder B, and the pharmacy stakeholder C. In the example of Figure 1, messages with indexes 1 and 2 were selected. Each message was sent in different transactions. However, selected messages can be added to the same transaction.

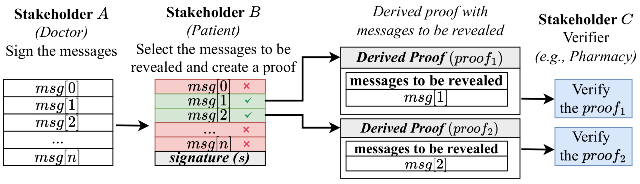

Fig. 1. BBS signatures and message selection with derived proofs

In summary, BBS signatures allow the data owner to selectively reveal certain pieces of data attributes to specific stakeholders in a reliable and scalable way. In our work, we use BBS signatures to allow the patient to select specific attributes of the prescription to share with other organizations.

IV. Decentralized Architecture with Consent Management and Privacy Protection

A. System Model and Threats

We assume a system comprising of patients, doctors, and pharmacies. When a patient visits a doctor, the doctor creates a new medical record that includes diagnostic data, personal details such as name and age, and prescriptions.

We focus on the following threats:

- Privacy threat: The patient's medical record includes sensitive data, which should not be revealed to unauthorized third parties without the patient's consent.
- Lack of accountability (or Illegal drug sales in an e-prescription use case): The lack of visibility into the medication supply chain leads to illegal medication sales, resulting in drug overdoses.
- Selective disclosure: Medical records include multiple fields such as patients' personal information, historical health data, and medical prescriptions. Only the recent medical prescription must be revealed to the pharmacist when a patient visits the pharmacy. Such scenarios highlight the need to share certain data attributes with specific stakeholders selectively.

Given these threats, this work aims to develop a solution by answering the following research questions.

Question 1: We aim to develop a transparent medical prescription system based on the blockchain without revealing sensitive data to unauthorized third parties. Note that the data stored on the blockchain is visible to the public on a blockchain platform. *How do we allow the patients to store and manage medical data in a tamper-proof ledger without violating patients' privacy?*

Question 2: When the data get stored on a digital system, doctors and health care agencies can access the data for diagnostic and survey purposes. Under this circumstance, it is important to let patients or data owners have visibility into data usage. *How do we allow the data owners to track and govern data usage by other parties?*

Question 3: When a medication supply chain operates without a regulator, pharmacies can easily acquire drugs from the manufacturer and sell them to patients without a valid prescription. *How do we prevent illegal drug sales by involving a regulator in a decentralized prescription system? To generalize it to other multi-stakeholder applications, how do we allow the regulatory body to access data for accountability and compliance verification within a decentralized data governance infrastructure?*

Question 4: All parties need not require access to all data. Certain stakeholders may require access to specific sensitive data belonging to other parties. For example, the regulator may want to audit the data for compliance check. *How do we selectively share specific data attributes with certain stakeholders in a reliable and scalable manner?*

B. Proposed Solution

We propose a decentralized data governance framework using blockchain technology and smart contracts to help patients (data owner) manage their data more efficiently. When a patient consults a doctor (data producer), the doctor creates prescription data by recording the diagnosis, recommended medications, and dosage. Then, the prescription data is encrypted using a changeable public key of the patient and stored on the blockchain via smart contracts. We assume that the patient shares her public key with the doctor when she makes an appointment to see the doctor. The patient needs to allow the pharmacist (data consumer) access to the prescription data to receive medication from the pharmacy. In particular, the patient selects data to be shared with other stakeholders. Through selective sharing, the patient has the flexibility to create different versions with allowed data while keeping unauthorized data secret. Besides, the data consumer requires proof to authenticate the integrity of the data as the data owner may share incorrect or modified data with the data consumer.

We propose a data access tracking mechanism within the contract state to monitor data accesses. In this way, any query or update in the status of the records will be registered. Note that the existing blockchain-based systems support writing data to an immutable ledger. Still, they do not monitor or provide support for governing data usage, which is necessary for data provenance and privacy. Our framework not only records the data on an immutable ledger in a privacy-preserving manner but also logs access requests to govern data usage.

Besides, we allow the data owner to decide which information she wants to share with which parties through a blockchain and smart contracts. The data consumer can also get proof showing the authenticity of the data. In the e-prescription use case, the doctor creates the prescription data for the patient, which means when the patient shares the prescription data with the pharmacy, she needs to prove that the doctor created the prescription data. Our framework lets the data owner create proof while enabling the data consumer to validate the data integrity by checking the data stored on the ledger with the data owner's consent.

To prevent illegal medications sales by the pharmacy, the regulatory agency will count, through a control contract, the number of drugs supplied to the pharmacy with the number of drugs sold (in the sales contract). Patients can report the pharmacy that sells unlawful drugs and receive rewards (tokens) through a smart contract through this approach.

We assume that the blockchain can hold encrypted prescription data for brevity. We can extend the framework by storing the encrypted prescription data on off-chain storage while maintaining the hash on-chain, which we leave for future work.

C. Overview of Smart Contracts

Figure 2 shows the contracts among stakeholders for the proposed model:

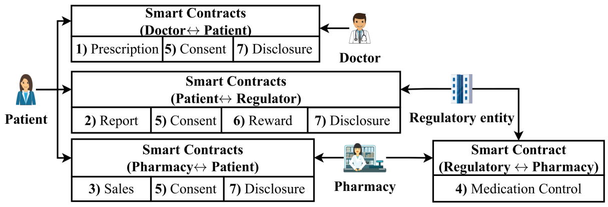

Fig. 2. Stakeholders and smart contracts

1) **Prescription Smart Contract**: the doctor creates the prescription data and invokes the *create_prescription* smart contract method. The contract state will be updated with the patient's personal information, recommended medications, and diagnosis. Each instance has the address of the doctor (sender) and patient (recipient). The *create_prescription* method only accepts transactions from the doctor. In this sense, only transactions signed by the doctor will be valid. For data usage tracking, any query to the contract, the stakeholder address will be updated in the *last_access* state.
2) **Report Smart Contract**: in case of illegal actions performed by the pharmacy, such as selling medication without a valid prescription (i.e., issued by the doctor), the patient can report it to the regulatory authority. If the report is valid, the patient can receive tokens as a reward. The contract instance contains the source address and the destination (regulator). From the *create_report* method, the current contract state will be updated with the source address and the data (description) of the denunciation with medication sold and pharmacy address. Personally identifiable information (PII) will not be stored in the contract state.

TABLE II
ACCESS TO INFORMATION BY STAKEHOLDERS AFTER PATIENT PERMISSION

Stakeholder	**Medical Records**		
	Personal Information (e.g., Name, Age)	**Diagnosis**	**Medication and Dosage**
Doctor	Yes	Yes	Yes
Patient	Yes	Yes	Yes
Pharmacy	No	No	Yes
Regulator	No	No	Yes

A note on reporting feature: in this work, we proposed a mechanism using a smart contract to allow the participants, particularly the patients, to create a report about illegal activities performed by the pharmacy. However, in future work, we will explore in detail how to protect users' privacy in case of denunciation and how to integrate data report verification and token economics in the prescription use case.

3) **Sales Smart Contract**: allows the pharmacy to sell medications to the patient. The pharmacy creates a contract instance with the patient to send sales transactions. Each sales transaction will be sent to the *sell_medication* contract method, and the current state is updated with the sales data (i.e., medication name, dosage, and price).
4) **Medication Control Smart Contract**: allows the regulator to account for the number of drugs supplied and sold by the pharmacy. The legal amount will be sent by the regulator and received by the pharmacy. Only the pharmacy will notify the number of medications sold, and the smart contract method will automatically update the number of available medications. In this way, the regulator will account for the relationship between drugs supplied and drugs sold.
5) **Patient Consent Smart Contract**: allows stakeholders to request the right to decrypt the patient's prescription data. The contract state includes the request origin address and the patient's consent, authorizing or not the data decryption. In Section IV-E, the request mechanism using smart contracts is presented.
6) **Reward Smart Contract**: used for the regulator to transfer tokens to the patient in case of complaint proof. We plan to develop and detail the token economy in our future work.
7) **Disclosure Smart Contract**: allows the patient to share selected data with other stakeholders. The contract keeps the shared items encrypted with the derived proof.

Figure 3 shows the steps for creating an instance and sending transactions. After uploading the contract to the network, an instance of the contract is created among the stakeholders with the definition of who sends and receives a transaction. The stakeholder (sender) sends a transaction to the network, informing the instance address and the transaction data described in Section IV-C. For example, the doctor sends a transaction to the network informing the instance address and the prescription data (i.e., patient personal information, medication, and diagnosis). The contract method will verify the transaction's validity by checking if the sender is the same as defined by the instance. To be added to the blockchain, transactions will follow the consensus steps of the underlying blockchain. If it is a valid transaction, the current contract state will be updated with the transaction data.

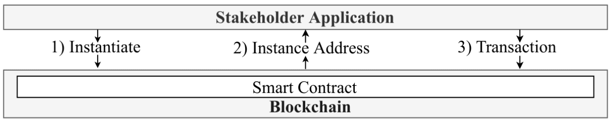

Fig. 3. Smart contract instance and transaction validation

D. Proxy Re-Encryption Mechanism and BBS Signatures

In this work, we use the proxy re-encryption technique to ensure data privacy and BBS signatures for selective disclosure. In this way, stakeholders can decrypt prescription data only with the patient's consent via the delegation mechanism. Table II shows the information that stakeholders can consult after the patient's permission, and the diagram in Figure 4 shows the architecture with the PRE and BBS operations:

1) From the appointment with the patient, the doctor creates a prescription containing the items: personal information (PI), medication (MED), and diagnosis (DIA) for future analysis. Before sending the prescription to the $create\_prescription$ smart contract method and being stored in the contract state, the prescription items are encrypted by the doctor application separately using the patient's public key (pk_P). The patient generates a public key for each appointment to ensure unlinkability, as the same public key may compromise anonymity. Although this approach increases the management complexity as the patient needs to maintain several private keys, the literature shows that the user can rotate a limited number of keys to maintain anonymity. *Therefore, the patient has flexibility and can consent to data sharing.*
2) The doctor creates a BBS signature with the encrypted prescription items.
3) The doctor application creates a transaction containing the encrypted items separately and submits it to the prescription contract. Without re-encryption, only the patient can decrypt the items using their private key.
4) If a stakeholder such as a doctor, pharmacy, or regulator needs to access some prescription information, it will be necessary to send a request to the patient informing the public key (pk) through the consent contract.
5) If the patient agrees to share some information, the application will produce a delegation key (rk) from the received public key (pk). Before sending the transaction, the delegation key will be encrypted using the stakeholder's public key to allow access to the delegation key.
6) The patient will request the data in the prescription contract and, through the selective disclosure mechanism using BBS implementation, select the item to be shared following the privacy filters in Table II. The selected item remains encrypted with the patient's public key. Thus, the re-encryption operation using the delegation key is necessary for the stakeholder to be able to decrypt the information. Similarly, the patient application will decrypt the capsule using their private key and encrypt with the stakeholder's public key. *Note that decrypting the capsule is not decrypting the data itself [24]. The capsule only refers to the item allowed by the patient and not all items in the prescription.*
7) The patient application will create a transaction containing a derived proof with the encrypted data and capsule allowed by the patient. The transaction will be submitted to the disclosure contract. As an example, Figure 4 shows the sharing of the medication item.
8) The stakeholder application will query the transaction in the disclosure contract and verify the authenticity of the derived proof generated by the patient application. This step verifies if the derived proof contains the same item issued by the doctor.
9) To access the items shared in the disclosure contract, it is necessary to perform the data re-encryption (RE) operation using the delegation key. The stakeholder (i.e., doctor, pharmacy, or regulator) will query and decrypt the delegation key through the consent contract and decrypt the capsule through disclosure contract. For example, Figure 4 shows the medication (CAP_{MED}) to be consulted by the pharmacy or regulatory agency. At this stage, the doctor's application will be able to re-encrypt all prescription data (i.e. personal information (CAP_{PI}), medication (CAP_{MED}), and diagnosis (CAP_{DIA}) as shown in Table II.
10) After the re-encryption step, stakeholders can decrypt the item with their respective private key (s_k) and analyze the information allowed by the patient.
In Figure 4, we use the Umbral cryptosystem algorithm [12] for proxy re-encryption operations where the capsule is used for step 9 and both (capsule and ciphertext) for step 10.

Note about patient revisiting doctor: From the architecture in Figure 4, if the patient revisits the doctor, it will be necessary to request the decryption rights (step 4) for the doctor to consult the prescription history. With the rights allowed and the data selected by the patient (steps 5, 6, and 7), the doctor application must verify the proof (step 8), re-encrypt the prescription capsules (i.e., CAP_{PI}, CAP_{MED} and CAP_{DIA}) and decrypt the re-encrypted items (i.e., $C_{PI_{RE}}$, $C_{MED_{RE}}$ and $C_{DIA_{RE}}$) in steps 9 and 10. Steps 4 to 10 are repeated for the other stakeholders, differing only in the information shared, as shown in Table II.

Note that sensitive prescription items are encrypted before being stored in the blockchain. In this way, records are private and immutable. Other organizations will only be able to decrypt the information with the patient's permission. The proxy re-encryption mechanism is a privacy software module

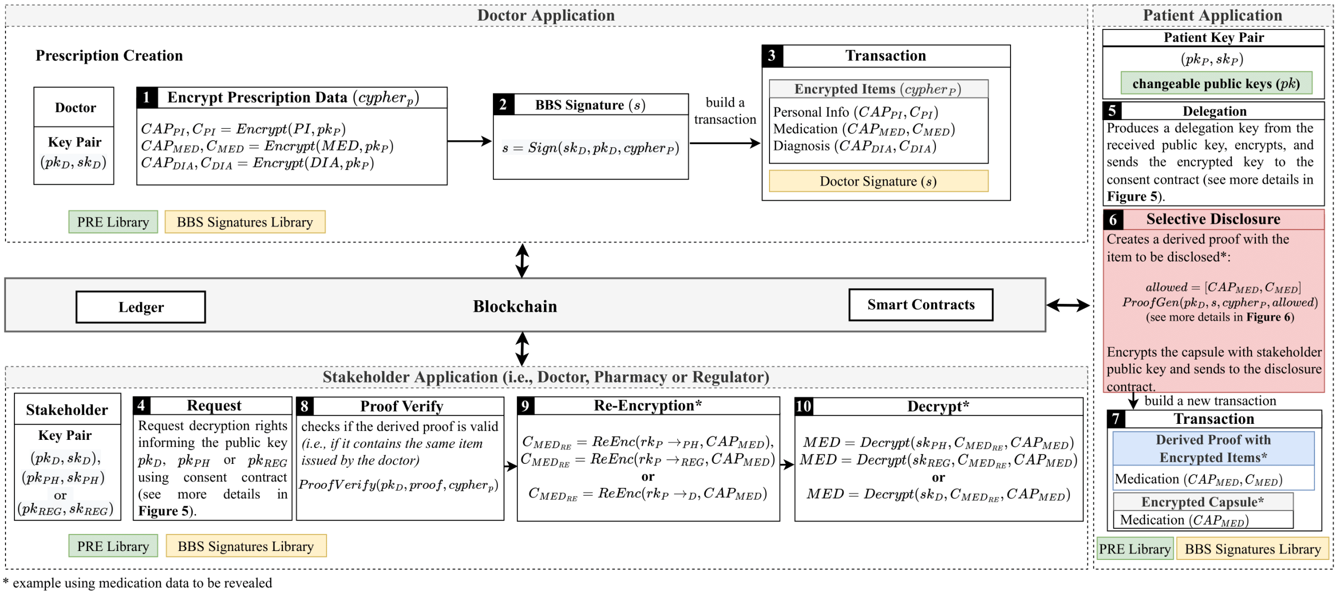
Doctor Application
Prescription Creation
Doctor
Key Pair
(pk_D, sk_D)
1 Encrypt Prescription Data $(cypher_p)$
$CAP_{PI}, C_{PI} = Encrypt(PI, pk_P)$
$CAP_{MED}, C_{MED} = Encrypt(MED, pk_P)$
$CAP_{DIA}, C_{DIA} = Encrypt(DIA, pk_P)$
2 BBS Signature (s)
$s = Sign(sk_D, pk_D, cypher_P)$
build a transaction
3 Transaction
Encrypted Items $(cypher_P)$
Personal Info (CAP_{PI}, C_{PI})
Medication (CAP_{MED}, C_{MED})
Diagnosis (CAP_{DIA}, C_{DIA})
Doctor Signature (s)
PRE Library
BBS Signatures Library
Ledger
Blockchain
Smart Contracts
Stakeholder Application (i.e., Doctor, Pharmacy or Regulator)
Stakeholder
Key Pair
(pk_D, sk_D),
(pk_{PH}, sk_{PH})
or
(pk_{REG}, sk_{REG})
4 Request
Request decryption rights informing the public key pk_D, pk_{PH} or pk_{REG} using consent contract (see more details in Figure 5).
8 Proof Verify
checks if the derived proof is valid (i.e., if it contains the same item issued by the doctor)
$ProofVerify(pk_D, proof, cypher_p)$
9 Re-Encryption*
$C_{MED_{RE}} = ReEnc(rk_P \rightarrow_{PH}, CAP_{MED})$,
$C_{MED_{RE}} = ReEnc(rk_P \rightarrow_{REG}, CAP_{MED})$
or
$C_{MED_{RE}} = ReEnc(rk_P \rightarrow_D, CAP_{MED})$
10 Decrypt*
$MED = Decrypt(sk_{PH}, C_{MED_{RE}}, CAP_{MED})$
$MED = Decrypt(sk_{REG}, C_{MED_{RE}}, CAP_{MED})$
or
$MED = Decrypt(sk_D, C_{MED_{RE}}, CAP_{MED})$
PRE Library
BBS Signatures Library
Patient Application
Patient Key Pair
(pk_P, sk_P)
changeable public keys (pk)
5 Delegation
Produces a delegation key from the received public key, encrypts, and sends the encrypted key to the consent contract (see more details in Figure 5).
6 Selective Disclosure
Creates a derived proof with the item to be disclosed*:
$allowed = [CAP_{MED}, C_{MED}]$
$ProofGen(pk_D, s, cypher_P, allowed)$
(see more details in Figure 6)
Encrypts the capsule with stakeholder public key and sends to the disclosure contract.
build a new transaction
7 Transaction
Derived Proof with Encrypted Items*
Medication (CAP_{MED}, C_{MED})
Encrypted Capsule*
Medication (CAP_{MED})
PRE Library
BBS Signatures Library
* example using medication data to be revealed

Fig. 4. Our decentralized data governance framework with support for PRE mechanism, consent management, data provenance, and privacy.

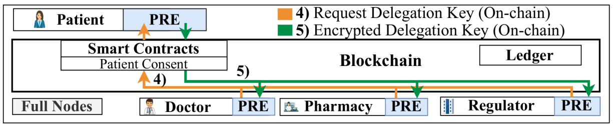

Fig. 5. On-chain request mechanism (step 4) and sending the encrypted delegation key (step 5) using smart contracts

implemented in the stakeholder application to act on confidential data sharing operations.

Note about proxy: Proxy is a software that only re-encrypts information. The proxies do not store any private keys and do not see any message from the ciphertexts. From their perspective, they only see an incoming ciphertext and the result after re-encryption, which is also a ciphertext.

A note on the security of unauthorized data: Our solution does not use a trusted third party to manage delegation keys. Instead, delegation keys are sent to stakeholders in encrypted form through a consent contract. Even with the delegation key, it is impossible to re-encrypt all other prescription items using a single capsule. In our solution, a capsule represents each prescription item. For a stakeholder to re-encrypt all items, the patient must provide all capsules. We emphasize that the delegation key is shared in encrypted form through the consent contract. In the case of prescription data, each capsule represents a single item of the prescription that is shared through the disclosure contract also in encrypted form.

E. Consent mechanism and delegation key

Figure 5 represents steps 4 and 5 of Figure 4 where each stakeholder is a full node (i.e., containing the PRE operations and blockchain). In step 4, the stakeholder sends a request to the patient through a consent contract. In step 5, the patient will create and encrypt the delegation key using the stakeholder's public key. In this way, requests are transparent to all network participants, and only the stakeholder can decrypt the delegation key.

F. BBS for selective disclosure

The BBS module acts as a piece of software that allows the patient to select the items to be shared with other stakeholders through transactions, as in Figure 6. From the application's point of view, a new transaction will be created containing a derived proof with only the allowed data (kept in encrypted form). For example, transaction 1 (in Figure 6) contains a derived proof with only the medication and can be shared with the pharmacy and regulator (as in the privacy levels in Table II). Transaction 2 contains a derived proof with all prescription data and is shared only with the doctor. Note that for stakeholders to be able to decrypt the data, the delegation key is required to perform steps 9 and 10 in Figure 4.

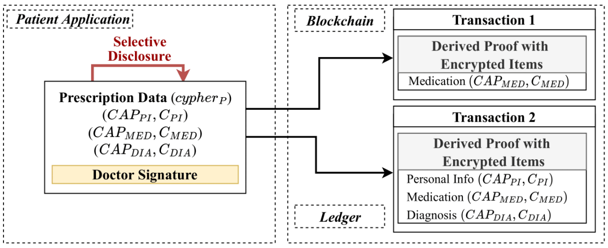
Patient Application
Selective Disclosure
Prescription Data $(cypher_P)$
(CAP_{PI}, C_{PI})
(CAP_{MED}, C_{MED})
(CAP_{DIA}, C_{DIA})
Doctor Signature
Blockchain
Ledger
Transaction 1
Derived Proof with Encrypted Items
Medication (CAP_{MED}, C_{MED})
Transaction 2
Derived Proof with Encrypted Items
Personal Info (CAP_{PI}, C_{PI})
Medication (CAP_{MED}, C_{MED})
Diagnosis (CAP_{DIA}, C_{DIA})

Fig. 6. Selective disclosure performed by patient application after stakeholder request

G. Patient Consent and Provenance

In our framework, the consent mechanism allows the patient to allow or revoke access to information using a decentralized, transparent, and tamper-proof architecture. Following Table II, the patient will be able to select the data to be shared with other healthcare organizations. The use of information must be transparent and traceable [27], that is, the patient must know what information is being used and by whom. Therefore, all requests and permissions are performed using smart contracts. From the data owner's point of view, there is a data lineage where it is possible to authorize and audit data sharing.

A note on PRE and BBS signatures: In our model, we use PRE for the patient to authorize or revoke access to one or more prescription items through the delegation key and capsule. Using BBS, the patient can create versions with the items issued by the doctor, maintaining integrity and authenticity. For example, using BBS, the pharmacy can check if the version created in step 6 of Figure 4 contains the same medication prescribed by the doctor in step 1 without accessing the other prescription items.

H. How does the proposed solution meet the objectives?

Question 1 focuses on providing transparency to the prescription system while protecting patients' privacy — our solution stores the patients' data on the blockchain but in an encrypted form. The prescription data is made available to other parties only after the patient's consent.

Question 2 focuses on governing data usage - our solution tracks the data access requests of consumers and permissions of data owners through a smart contract and immutable ledger. Therefore, data owners can have visibility into their data and its usage. We understand that a malicious data consumer may access the patient's data with their permission and then post it on a black market or other digital platforms. We plan to investigate digital watermarking and steganography in our future work to overcome this problem [28].

Question 3 aims to prevent illegal sales of drugs — our solution includes a regulator in the prescription system, thereby providing transparency and accountability. By allowing the regulator to track the flow of goods through a tamper-proof ledger, our solution reduces illegal drug sales. Besides, the regulator still requires the permission of the data owner to view records, making the auditing process transparent and trustworthy.

Question 4 protects sensitive data and intellectual property in multi-stakeholder applications. By using BBS Signature, our solution allows the data owner to selectively reveal partial data to specific stakeholders while proving ownership through a smart contract.

I. How can our solution be used for other multistakeholder applications?

In our work, we present electronic prescriptions as a use case of multi-stakeholder application. However, our solution can be broadly used in applications where the data owner can authorize and create versions to share only part of all data with other institutions. Therefore, the proposed solution can generally be used for other applications with data producers, data owners, and data consumers. In the use case presented, the doctors are the data producers, and the patient is the data owner, allowing or revoking access to the data consumers (e.g., pharmacy). We specifically use proxy re-encryption (PRE) as consent to share the delegation key and capsule with the data consumer. In addition, we use BBS signatures to enable the data owner to create versions of the data issued by the data producer (with one or more items), maintaining authenticity and keeping unauthorized data secret. The data consumer can re-encrypt and decrypt only the item allowed by the data owner. In addition, the data consumer can verify if the item is the same issued by the data producer.

V. Threat Analysis

This section discusses how our framework overcomes privacy threats. Our solution guarantees the following properties:

Non-repudiation: The system should ensure that stakeholders cannot deny the data ownership or access. In a multi-stakeholder application such as e-prescription, the patient authenticates the doctor to store the data in the blockchain ledger in encrypted form. Therefore, the patient provides consent to the doctor to store the data in the ledger. From the data consumers' viewpoint, they request access to the data. Our framework stores the access request and the owner's response to the request in the ledger. Besides, for accountability purposes, we introduce a smart contract to track the medication sales, which helps the regulators audit the medication supply chain. It is important to note that pharmacies may sell drugs without submitting a digital entry to the ledger. We believe this problem requires a fully regulated medication supply chain wherein all the prescriptions flow through the regulators with full accountability. In summary, our framework records critical activities on blockchain to guarantee non-repudiation.

Anonymity and Unlinkability: All the transactions use only the public key of the stakeholders. Our system uses a changeable public key when patients visit the hospital, doctor, and pharmacy to strengthen anonymity and unlinkability. Although this approach increases the management complexity as the patient needs to maintain several private keys, the literature shows that a limited number of keys can be rotated by the user to maintain anonymity in digital systems [29]. Besides, our framework focuses on ensuring anonymity on the digital platform, while anonymity in physical areas is outside the scope of this work.

Confidentiality: The system should ensure that data is not made available to any party without the owner's authorization. The data owner needs to authorize access to their data. Besides, the data owner can selectively share specific attributes (keeping in encrypted form) with certain stakeholders in our framework, ensuring confidentiality.

Integrity: The system must prevent data tampering—our framework stores all the data in the blockchain ledger in encrypted form. The data cannot be modified in the ledger unless the stakeholders manage to convince the majority of the nodes that run the ledger.

Impersonation attack: The data owner must protect the private key to govern and protect data. When attackers manage to compromise the data owner's computing device, they can misuse data by impersonation attack. Like any blockchain-based solution, our framework also requires safe practices to manage private keys to ensure security and privacy, such as authentication is crucial to prevent unauthorized access to medical applications and key pairs. Newaz et al. [4] analyzed some types of authentication (e.g., single, multi-factor authentication, and continuous authentication) for healthcare applications.

VI. Proof-of-Concept Implementation and Evaluation

A. Privacy: Proxy Re-Encryption

Evaluation Goals: To understand the overhead and feasibility of PRE and BBS signature operations, we evaluate memory allocation and execution time for steps in Figure 4. We evaluated encrypting (step 1), creating a delegation key (step 5), re-encryption (step 9), and decrypting (step 10).

Evaluation Setup and Methodology: The PRE evaluation programs and scripts were implemented in Python programming language using NuCypher pyUmbral PRE technology, an open-source implementation that uses the *secp256k1* elliptic curve [30]. We use the module *tracemalloc*, a tool to trace memory blocks allocated by the evaluation program during the execution of the operations. For execution time, we use the *time* module to calculate the difference between the start and end of each operation. All software created for evaluation is available on GitHub [31]. We implement the smart contracts using the following BFT platforms: CosmWasm and Hyperledger Besu, discussed in the section VI-C. For comparison, we evaluated the contracts methods using the Ethereum platform, and will be discussed in the section VI-D.

To identify realistic file sizes for medications and dosage prescriptions, we used the English Prescribing Dataset [32]. Prescription items used for evaluation are represented in separate text files with different sizes ranging from 0.43 Kilobyte (kB) to 0.82 kB for personal information, 0.24 kB to 0.53 kB for medication, and 2.18 kB to 8975.74 kB ≈ 8.76 Megabyte (MB) for diagnosis. While the file sizes are inferred from [32], file contents are randomly generated by the evaluation software. In total, 1000 iterations were performed for different file sizes. We used a Linux virtual machine with an Intel Core i7-10510U 2.303GHz (Quad-Core) processor and 8GB of RAM for the evaluation.

Table III presents information about the maximum (Max.), minimum (Min.), and average (Avg.) size of each prescription item.

*1) **Memory Allocation Evaluation**:* Figure 7 shows the average memory allocated for application-level PRE operations using the data files of Table III. In all operations, most memory blocks allocated are for asymmetric cryptography operations using the *secp256k1* curve. In particular, pyUmbral uses OpenSSL via *Cryptography.io* library. We highlight the key findings below:

TABLE III
Maximum, minimum and average files size with prescription items

	Prescription Data (kB)		
File Size	**Personal Info**	**Medication and Dosage**	**Diagnosis**
Min.	0.43	0.24	2.18
Max.	0.82	0.53	8975.74 (≈ 8.76 MB)
Avg.	0.62	0.39	4538.57 (≈ 4.43 MB)

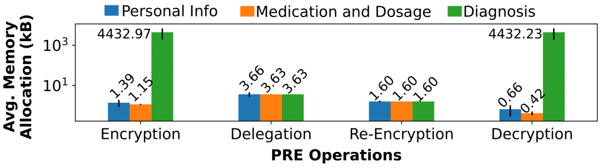

Fig. 7. Average memory allocation by *tracemalloc* in PRE operations

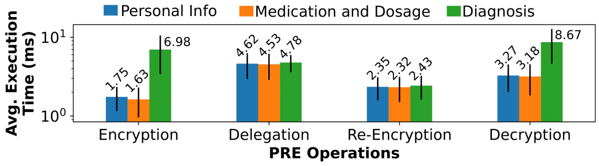

Fig. 8. Average execution time in PRE operations

- The diagnosis data's encryption (step 1 of Figure 4) consumes significant memory since it contains a large amount of data compared to other data items (see Table III). The diagnosis data requires an average allocation of 4432.97 kB (≈ 4.32 MB) with a standard deviation (Std.) of 2524.97 kB (≈ 2.46 MB). At the same time, for personal information and medication, it was 1.39 kB and 1.15 kB with Std. of 0.55 kB and 0.08 kB, respectively, as shown in Figure 7.
- In the delegation stage (step 5 of Figure 4), which is performed on the patient's application, the average memory allocation was around 3.66 kB (with Std. of 0.91 kB) for personal information, medication, and diagnosis. Similarly, in the re-encryption stage (step 9 of Figure 4), the average allocated memory was around 1.60 kB (with Std. of 0.01 kB).
- Similar to the encryption stage, the decryption (step 10 of Figure 4) also consumes significant memory due to the execution of computationally-intensive cryptography operations.

*2) **Execution Time Evaluation**:* Figure 8 shows the average execution time for application-level PRE operations using the data files of Table III.

To encrypt the diagnostic data (step 1 of Figure 4), it took an average of 6.98 milliseconds (ms) with a standard deviation of 3.56 ms for files ranging from 2.18 kB to 8975.74 kB ≈ 8.76 Megabyte (MB). The medication and dosage data took an average of 1.63 ms with a standard deviation of 0.67 ms for data files ranging from 0.24 kB to 0.53 kB. For personal information, it took 1.75 ms with a standard deviation of 0.59 ms for data files from 0.43 kB to 0.82 kB.

Similar to the average memory allocation evaluation, there were slight variations in the average processing times in the delegation and re-encryption stage for the prescription items. The delegation stage (step 5 of Figure 4) took an average of 4.78 ms (with Std. of 1.17 ms) for diagnosis data, 4.53 ms (with Std. of 1.65 ms) for medication and dosage, and 4.62 ms (with Std. of 1.66 ms) for personal information. In the re-encryption operation (step 9 of Figure 4), the average execution time was 2.43 ms (with Std. of 0.81 ms) for diagnosis, 2.32 ms (with Std. of 0.82 ms) for medication and dosage, and 2.35 ms (with Std. of 0.76 ms) for the patient's personal information. In the decrypt stage (step 10 of Figure 4), the item that obtained the highest execution average was the diagnosis with 8.67 ms (with Std. of 4.05 ms). In comparison, medication and dosage took 3.18 ms, and the patient's personal information took 3.27 ms with Std. of 1.36 ms and 1.25 ms, respectively.

These results show PRE operations' cost of memory allocation and execution time is relative to the data size. In our evaluation, even with text files with sizes in Megabyte (see Table III), the operations did not exceed 50 ms to be executed. In this sense, our proposed framework protects privacy and manages consent with a low operational overhead. We also believe PRE operations can run on platforms like Raspberry Pi or mobile phones.

B. Selective Disclosure: BBS Signatures

We evaluated, in a total of 100 executions, the execution time of BBS operations using the MATTR Jsonld implementation that uses *BLS12-381* key pairs [33]. We use the Performance API to evaluate the execution time between operations. Table IV shows the average (Avg.), standard deviation (Std.), minimum (Min.), and maximum (Max.) for the following BBS operations: signing all prescription data (messages), create a proof with all prescription data and verify the derived proof.

For a doctor to sign all the prescription data (i.e., personal information, medication, and diagnosis) performed in step 2 of Figure 4 took an average of 260.18 ms, with a maximum execution time of 341.93 ms and a minimum of 216.56 ms. For the patient application, producing a derived proof selecting all prescription items (step 6 of Figure 4) took an average of 340.11 ms with a maximum of 425.51 ms and a minimum of 279.22 ms. The average execution time to verify the derived proof (step 8 of Figure 4) with all prescription items took 192.17 ms, with a maximum time of 268.49 ms and a minimum of 149.39 ms.

C. Smart Contract: CosmWasm and HyperLedger Besu Implementation

We evaluate transaction time (with block creation) with encrypted prescription items (after the encryption step of Figure 4). We used two BFT platforms: CosmWasm using the Tendermint consensus engine and Hyperledger Besu with the Istanbul Byzantine Fault Tolerance 2 (IBFT2) consensus. For the CosmWasm platform, we used a test network provided by the platform with four validator nodes [34]. To evaluate Hyperledger Besu, we use a local network with four validator nodes and *web3.js* (a JavaScript API) to integrate it and send transactions. The steps to automate the sending of transactions to the network were implemented in a shell script. All software and contracts developed for model evaluation are available on GitHub [31]. Table V shows the evaluation setup between BFT platforms.

TABLE IV
EXECUTION TIME OF BBS OPERATIONS IN A TOTAL OF 100 EXECUTIONS FOR ALL PRESCRIPTION DATA

	Execution Time (ms)		
	Doctor Sign The Prescription Data	Patient Produces a Derived Proof	Verify the Derived Proof
Avg.	260.18	340.11	192.17
Std.	27.52	29.65	23.55
Min.	216.56	279.22	149.39
Max.	341.93	425.51	268.49

TABLE V
EVALUATION SETUP: COSMWASM AND HYPERLEDGER BESU

	CosmWasm	Hyperledger Besu
Consensus	Tendermint	IBFT2
Block Time	5 seconds	5 seconds
Version	CosmWasm: 1.0 wasmd: 0.23.0	22.1.3
Test Network	04 Validators (Non Local)	04 Validators (Local)

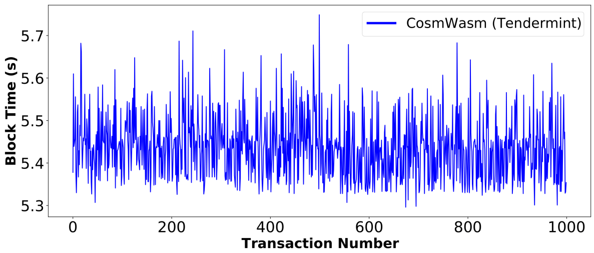

Fig. 9. Block time using the CosmWasm platform with four validator nodes in a total of 1000 transactions containing prescription data

A note on block time: In our evaluation, block time is the time required for the transaction created in step 3 of Figure 4 (with encrypted items) to be stored on the blockchain.

Block Time for Smart Contracts in CosmWasm: Figure 9 shows the block time for all transactions using the CosmWasm platform and Table VI shows the average, minimum, and maximum time to create a block containing around 1 Kb of prescription data (i.e., patient's personal information, medication, and diagnosis) in 1000 transactions. Time refers to the Tendermint consensus process and block creation. On average, the time for a transaction to be validated by contract method

TABLE VI
BLOCK TIME BETWEEN COSMWASM AND HYPERLEDGER BESU PLATFORMS FOR A TOTAL OF 1000 TRANSACTIONS

	Block Time (in seconds)	
	CosmWasm (Tendermint)	**Hyperledger Besu (IBFT2)**
Avg.	5.43	5.00
Std.	0.07	0.00
Min.	5.29	5.00
Max.	5.74	5.00

TABLE VII
CONTRACT DEPLOY INFORMATION USING THE ETHEREUM IMPLEMENTATION AND THE ROPSTEN TESTNET

Smart Contract	**Data Size (bytes)**	**Block Mining Time (s)**			**Txn. Fee (ETH)**
		Min.	**Max.**	**Avg.**	
Prescription	3021	1	105	11.69	0.00177497
Report	1419	1	60	10.44	0.00088486
Sales	2245	1	50	11.97	0.00136244
Medication Control	1771	1	50	11.54	0.00110506
Consent	3552	1	87	13.27	0.00205878
Reward	532	1	58	13.18	0.00042597
Disclosure	2096	1	123	32.77	0.00124628

and block creation took 5.43 seconds, with the minimum and maximum time being 5.29 seconds and 5.74 seconds, respectively. The variation in transaction time is due to the consensus delay, including peer-to-peer messaging between validator nodes.

Block Time for Smart Contracts in HyperLedger Besu: Unlike CosmWasm, in which the nodes are on different networks, the block generation time was kept at 5.00 seconds because the validator nodes shared the same local network with the minimum delay between the exchange of consensus messages.

A note about scalability: The complexity of exchanging consensus messages increases with the number of validator nodes. The Hyperledger Besu team analyzed that with up to 30 validator nodes, the network operates without performance loss at light loads with IBFT2 consensus [35]. Cason et al. [36] analyzed the performance loss of Tendermint with the increase in validator nodes. In particular, the authors analyzed a configuration for 16, 32, 64, and 128 nodes in a geographically distributed network.

D. Smart Contract: Ethereum Implementation

We measure the transaction (txn) fee and the average block mining time in the Ethereum platform using the Ropsten test network. We have implemented the smart contracts using Solidity programming language. To deploy and interact with contract methods, we used the Remix platform and *web3.js* library. MetaMask wallet was used to obtain transaction details (e.g., transaction fee and data size), and Etherscan to monitor transactions and blockchain information.

Table VII presents the contracts with the size in bytes with the costs necessary for the deployment on the network (from MetaMask wallet). We performed 100 iterations for each contract and method to calculate the block's average mining time. The average mining time remained between 10.43 seconds and 32.77 seconds. As a comparison, the average time to create blocks using the CosmWasm and Hyperledger Besu platforms was 5.43 seconds and 5.00 seconds, respectively.

Table VIII shows the information about sending transactions for the contract methods: the doctor creates a prescription, requests delegation, and for the patient to authorize access to the information according to the steps shown in Figure 4. Table IX shows transaction information about medication sales, creating a report, sending rewards, supplying medications, and updating medication sold.

Note on Blockchain-agnosticism: The evaluation with CosmWasm, Hyperledger Besu, and Ethereum shows that our data governance framework is agnostic to the underlying blockchain platform. One can implement our data governance framework on any blockchain platform that supports smart contracts.

E. End-to-end Execution Time

Table X shows the average total execution time in seconds of PRE and BBS operations following the steps of Figure 4. The results show that asymmetric cryptographic operations such as PRE and BBS kept below 1 second to be executed on all prescription data (i.e., personal information, medication, and diagnosis) following the file sizes in Table III. The main difference between the platforms is the block creation time with the transactions. BFT platforms like CosmWasm and Hyperledger Besu have a lower block time than Ethereum's Proof of Work. At the same time, Ethereum uses the computational effort mechanism to solve a cryptographic challenge.

F. Summary of Key Results

In summary, our evaluation shows that:

- Our proposed framework is agnostic to the underlying blockchain platform.
- PRE and BBS signature schemes introduce reasonable overhead while providing security and privacy guarantees.
- The blockchain can provide privacy-preserving and tamper-proof communication between data owners and data consumers.
- The data owner can selectively share specific attributes with specific stakeholders.

VII. CONCLUSIONS

Real-world multi-stakeholder applications such as medical e-prescription and supply chain deal with digital and sensitive data, demanding privacy protection, consent management, data provenance, and transparency. We have presented a decentralized data governance framework for e-prescription that uses proxy re-encryption and smart contracts to let data owners control and manage their data through a trusted and transparent blockchain platform. We have shown how the data

TABLE VIII
PRESCRIPTION AND CONSENT CONTRACT METHODS WITH FEE PER TRANSACTION IN ETHEREUM IMPLEMENTATION WITH ROPSTEN TESTNET

Steps of Figure 4	**Event/Contract Method**	**Hexadecimal Data Size (bytes)**	**Invoked by**	**Smart Contract**	**Block Mining Time (s)**			**Txn. Fee (ETH)**
					Min.	**Max.**	**Avg.**	
1, 2 and 3	create_prescription	7012	Doctor	Prescription	1	58	12.17	0.00159011
4	request_delegation	100	Doctor and Pharmacy	Consent	1	57	12.69	0.00012677
5	set_consent	612	Patient	Consent	1	48	10.43	0.00022021
6 and 7	set_disclosure	260	Patient	Disclosure	1	136	31.64	0.00026361

TABLE IX
SALES, REPORT, REWARD AND MEDICATION CONTROL CONTRACT METHODS WITH FEE PER TRANSACTION IN ETHEREUM IMPLEMENTATION WITH ROPSTEN TESTNET

Event/Contract Method	**Hexadecimal Data Size (bytes)**	**Invoked by**	**Smart Contract**	**Block Mining Time (s)**			**Txn. Fee (ETH)**
				Min.	**Max.**	**Avg.**	
sell_medication	356	Pharmacy	Sales	1	85	10.83	0.00013032
create_report	676	Patient	Report	1	55	11.78	0.00021107
send_reward	36	Regulator	Reward	1	52	11.24	0.00006627
supply_medications	36	Regulator	Medication Control	1	61	13.86	0.00009519
update_medications_sold	36	Pharmacy	Medication Control	1	102	14.18	0.00009589

TABLE X
SUM OF THE AVERAGE EXECUTION TIME OF THE PRE AND BBS OPERATIONS WITH THE BLOCK CREATION TIME FOR THE STEPS IN FIGURE 4

Steps of Figure 4	**1, 2 and 3**			**4, 5, 6 and 7**			**8, 9 and 10**
Total of Transactions	1			3			0
Total PRE (s)	0.01036			0.01393			0.02222
Total BBS (s)	0.26018			0.34011			0.19217
	CosmWasm	**Besu**	**Ethereum**	**CosmWasm**	**Besu**	**Ethereum**	-
Total Block Time (s)	5.43	5.00	12.17	16.29	15.00	54.76	-
Total (Sum) (s)	**5.70**	**5.27**	**12.44**	**16.64**	**15.35**	**55.11**	**0.21**

owners can record all the access requests and consents in an immutable ledger to monitor data lineage. Our proof-of-concept implementation uses CosmWasm, Hyperledger Besu, Ethereum, pyUmbral proxy re-encryption, and BBS signatures library to assess the feasibility and performance. Our evaluation results show that the proposed architecture can protect data owners' privacy and govern sensitive data access with minimal overhead. Our data governance framework is application-agnostic, and hence, it can be explored in any multi-stakeholder applications that deal with sensitive and private digital data.

ACKNOWLEDGEMENT

Jo Ueyama would like to thank the Brazilian Research Council (CNPq) and São Paulo Research Foundation (FAPESP) for financing the bulk of his research, FAPESP grant ID 2013/07375-0. Rodrigo D. Garcia would like to acknowledge the financial support by the Brazilian Research Council (CNPq, process 133470/2020-2) for his research.

REFERENCES

[1] Bader Aldughayfiq and Srinivas Sampalli. Digital health in physicians' and pharmacists' office: a comparative study of e-prescription systems' architecture and digital security in eight countries. *Omics: a journal of integrative biology*, 25(2):102–122, 2021.

[2] Mahnaz Samadbeik, Maryam Ahmadi, Farahnaz Sadoughi, and Ali Garavand. A copmarative review of electronic prescription systems: Lessons learned from developed countries. *Journal of research in pharmacy practice*, 6(1):3, 2017.

[3] Ismail Keshta and Ammar Odeh. Security and privacy of electronic health records: Concerns and challenges. *Egyptian Informatics Journal*, 22(2):177–183, 2021.

[4] Akm Iqtidar Newaz, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. A survey on security and privacy issues in modern healthcare systems: Attacks and defenses. *ACM Transactions on Computing for Healthcare*, 2(3):1–44, 2021.

[5] Jigna J Hathaliya and Sudeep Tanwar. An exhaustive survey on security and privacy issues in healthcare 4.0. *Computer Communications*, 153:311–335, 2020.

[6] Imran Makhdoom, Ian Zhou, Mehran Abolhasan, Justin Lipman, and Wei Ni. Privysharing: A blockchain-based framework for privacy-preserving and secure data sharing

in smart cities. *Computers & Security*, 88:101653, 2020.

[7] Alevtina Dubovitskaya, Furqan Baig, Zhigang Xu, Rohit Shukla, Pratik Sushil Zambani, Arun Swaminathan, Md Majid Jahangir, Khadija Chowdhry, Rahul Lachhani, Nitesh Idnani, et al. Action-ehr: patient-centric blockchain-based electronic health record data management for cancer care. *Journal of medical Internet research*, 22(8):e13598, 2020.

[8] Vikas Jaiman and Visara Urovi. A consent model for blockchain-based health data sharing platforms. *IEEE access*, 8:143734–143745, 2020.

[9] Giuseppe Albanese, Jean-Paul Calbimonte, Michael Schumacher, and Davide Calvaresi. Dynamic consent management for clinical trials via private blockchain technology. *Journal of ambient intelligence and humanized computing*, 11(11):4909–4926, 2020.

[10] Ahsan Manzoor, An Braeken, Salil S Kanhere, Mika Ylianttila, and Madhsanka Liyanage. Proxy re-encryption enabled secure and anonymous iot data sharing platform based on blockchain. *Journal of Network and Computer Applications*, 176:102917, 2021.

[11] Rodrigo Dutra Garcia, Gowri Sankar Ramachandran, Raja Jurdak, and Jo Ueyama. A blockchain-based data governance with privacy and provenance: a case study for e-prescription. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–5, 2022. doi: 10.1109/ICBC54727.2022.9805545.

[12] DAVID Nunez. Umbral: a threshold proxy re-encryption scheme. *NuCypher Inc and NICS Lab, University of Malaga, Spain*, 2018.

[13] Ahmed Raza Rajput, Qianmu Li, and Milad Taleby Ahvanooey. A blockchain-based secret-data sharing framework for personal health records in emergency condition. In *Healthcare*, volume 9, page 206. MDPI, 2021.

[14] Tong Min Kim, Seo-Joon Lee, Dong-Jin Chang, Jawook Koo, Taenam Kim, Kun-Ho Yoon, and In-Young Choi. Dynamichain: development of medical blockchain ecosystem based on dynamic consent system. *Applied Sciences*, 11(4):1612, 2021.

[15] Dara Tith, Joong-Sun Lee, Hiroyuki Suzuki, WMAB Wijesundara, Naoko Taira, Takashi Obi, and Nagaaki Ohyama. Patient consent management by a purpose-based consent model for electronic health record based on blockchain technology. *Healthcare Informatics Research*, 26(4):265–273, 2020.

[16] Chaochen Hu, Chao Li, Guigang Zhang, Zhiwei Lei, Mira Shah, Yong Zhang, Chunxiao Xing, Jinpeng Jiang, and Renyi Bao. Crowdmed-ii: a blockchain-based framework for efficient consent management in health data sharing. *World Wide Web*, 25(3):1489–1515, 2022.

[17] Zeng Chen, Weidong Xu, Bingtao Wang, and Hua Yu. A blockchain-based preserving and sharing system for medical data privacy. *Future Generation Computer Systems*, 124:338–350, 2021.

[18] Renpeng Zou, Xixiang Lv, and Jingsong Zhao. Spchain: Blockchain-based medical data sharing and privacy-preserving ehealth system. *Information Processing & Management*, 58(4):102604, 2021.

[19] Patrick Li, Scott D Nelson, Bradley A Malin, and You Chen. Dmms: A decentralized blockchain ledger for the management of medication histories. *Blockchain in healthcare today*, 2, 2019.

[20] Ahsan Manzoor, Madhsanka Liyanage, An Braeke, Salil S. Kanhere, and Mika Ylianttila. Blockchain based proxy re-encryption scheme for secure iot data sharing. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 99–103, 2019. doi: 10.1109/BLOC.2019.8751336.

[21] Yingwen Chen, Bowen Hu, Hujie Yu, Zhimin Duan, and Junxin Huang. A threshold proxy re-encryption scheme for secure iot data sharing based on blockchain. *Electronics*, 10(19), 2021. ISSN 2079-9292. doi: 10.3390/electronics10192359. URL https://www.mdpi.com/2079-9292/10/19/2359.

[22] Rahma Mukta, James Martens, Hye-young Paik, Qinghua Lu, and Salil S Kanhere. Blockchain-based verifiable credential sharing with selective disclosure. pages 959–966, 2020.

[23] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *International conference on the theory and applications of cryptographic techniques*, pages 127–144. Springer, 1998.

[24] Threshold network. https://docs.nucypher.com. Accessed: 2022-08-05.

[25] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Annual international cryptology conference*, pages 41–55. Springer, 2004.

[26] Decentralized Identity Foundation. The bbs signature scheme. https://github.com/decentralized-identity/bbs-signature, 2022. Accessed: 2022-11-13.

[27] Prasanth Varma Kakarlapudi and Qusay H Mahmoud. A systematic review of blockchain for consent management. In *Healthcare*, volume 9, page 137. MDPI, 2021.

[28] Sijia Zhao and Donal O'Mahony. Bmcprotector: A blockchain and smart contract based application for music copyright protection. In *Proceedings of the 2018 International Conference on Blockchain Technology and Application*, pages 1–5, 2018.

[29] Seungyeop Han, Vincent Liu, Qifan Pu, Simon Peter, Thomas Anderson, Arvind Krishnamurthy, and David Wetherall. Expressive privacy control with pseudonyms. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, page 291–302, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320566. doi: 10.1145/2486001.2486032. URL https://doi.org/10.1145/2486001.2486032.

[30] NuCypher. pyumbral pre. https://github.com/nucypher/pyUmbral, 2022. Accessed: 2022-11-13.

[31] R.D Garcia. A decentralized e-prescription model. https:

//github.com/rodrigodg1/e-prescription, 2022. Accessed: 2022-11-13.

[32] English Prescribing Dataset. English prescribing dataset (epd). https://opendata.nhsbsa.net/dataset/english-prescribing-data-epd, 2022. Accessed: 2022-11-13.

[33] MATTR. Jsonld signatures bbs. https://github.com/mattrglobal/jsonld-signatures-bbs, 2022. Accessed: 2022-11-13.

[34] CosmWasm. Cosmwasm testnets. https://github.com/CosmWasm/testnets, 2022. Accessed: 2022-11-13.

[35] Hyperledger Besu. Maximum validator count for an ibft2 network. https://bit.ly/3Paxubv, 2022. Accessed: 2022-11-13.

[36] Daniel Cason, Enrique Fynn, Nenad Milosevic, Zarko Milosevic, Ethan Buchman, and Fernando Pedone. The design, architecture and performance of the tendermint blockchain network. pages 23–33, 2021.

Rodrigo Dutra Garcia has a degree in computer engineering (2019). Currently, he is an MS candidate in Computer Science and Computational Mathematics at the University of São Paulo (USP). His research interests include blockchain technology, privacy, and computer networks.

Gowri Sankar Ramachandran received his M.Sc. from Malardalen University, Sweden and Ph.D. degree from KU Leuven, Belgium. He is a research fellow in distributed systems, blockchain, and the Internet of Things at Queensland University of Technology, Australia. He was a postdoctoral researcher and senior research associate at University of Southern California, USA, between 2017 and 2020. His research interests include blockchain, IoT, and distributed computing. He has published over 50 peer-reviewed publications. He serves on the Organizing Committees of top international conferences, including ICBC, IPSN, IoTDI, and SenSys. He was the general Vice-Chair of ACM BlockSys-2022, a co-located workshop with SenSys-2022.

Raja Jurdak received his M.S. and Ph.D. degrees from the University of California, Irvine. He is a professor of distributed systems and Chair in Applied Data Sciences at Queensland University of Technology, and Director of the Trusted Networks Lab. He previously established and led the Distributed Sensing Systems Group at CSIRO's Data61. He also spent time as a visiting academic at MIT and Oxford University in 2011 and 2017. His research interests include blockchain, IoT, trust, mobility, and energy efficiency in networks. He has published over 230 peer-reviewed publications, including two authored books most recently on blockchain in cyberphysical systems in 2020. His publications have attracted over 11,000 citations, with an h-index of 47.

He serves on the Editorial Boards of Ad Hoc Networks and Nature Scientific Reports, and on the Organizing and Technical Program Committees of top international conferences, including Percom, ICBC, IPSN, WoWMoM, and ICDCS. He was TPC Co-Chair of ICBC in 2021. He is a conjoint professor with the University of New South Wales and a Visiting Researcher with CSIRO Data61.

Jó Ueyama is a Full Professor at the Institute of Mathematics and Computer Science (ICMC) of the University of São Paulo (USP). Prof. Ueyama is also a Brazilian Research Council (CNPq) fellow. He completed his Ph.D. in computer science at the University of Lancaster (England) in 2006. Before joining USP, he was a research fellow at the University of Kent at Canterbury (England). Jó has published 60 journal articles and more than 100 conference papers. His main research interest includes Computer Networks, Security, and Blockchain.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Applications that involve multiple stakeholders, such as healthcare, the internet of things, and supply chain management, require privacy protection and management of private data. This research presented a blockchain-based governance system investigating e-prescription to provide privacy and enable consent management and selective sharing. The study was organized in a collection of articles with the following order of contributions: smart contract implementation and evaluation using BFT-based platforms such as Tendermint consensus, Hyperledger Fabric, and Hyperledger Besu comparing the operation costs to Ethereum PoW; Implementation and evaluation of proxy re-encryption operations in e-prescription use case using NuCypher pyUmbral PRE library to enable patient consent; Implementation and evaluation of BBS signatures to enable selective sharing by the patient using MATTR JSON-LD library.

To answer RQ1 regarding how blockchain and smart contracts can be utilized to secure and manage sensitive data in a tamper-proof ledger, Chapter 2 and 3 provides a study for electronic prescription use cases in BFT platforms. The first article proposed a decentralized e-prescription system using smart contracts. The proposed system aims to securely manage electronic medical records using tamper-proof and transparent blockchain features to prevent fraudulent activities such as data tampering. Moreover, the contracts were implemented, evaluated, and compared using two BFT-based platforms: CosmWasm (Tendermint consensus) and Hyperledger Fabric. The results indicate that a higher number of validator nodes enhances the system's fault tolerance. However, it also leads to increased latency due to BFT consensus message traffic. In addition to the first work, the second article presented an extended evaluation and privacy-preserving discussion in multi-stakeholder applications such as e-prescription. In particular, the study compared the operational costs of smart contracts in CosmWasm and Hyperledger Besu regarding CPU and memory used by validator nodes during consensus. Moreover, the work evaluated the block time of BFT platforms with Ethereum mining time. The results show that BFT-based platforms are feasible for healthcare applications compared to PoW solutions.

In addition to the second article, the third article proposed a blockchain-based system with privacy protection and consent management for multi-stakeholder settings. The system aims to respond to the second research question (RQ2) about preserving data owners' privacy and how data owners can track and govern data usage by other parties. The system keeps all the data encrypted and requests stored in a transparent and tamper-proof ledger allowing the data owner to delegate access to other parties. Particularly, the study employed proxy re-encryption to provide data privacy and enables data owner consent in data sharing. The evaluations show that proxy re-encryption protects patients' privacy and enables data governance with minimal overhead. Furthermore, to address RQ3 concerning how can a regulatory entity access data for accountability and compliance verification in a decentralized data governance system, the fourth article using patient permission, includes smart contracts to enable the regulator entity to track the flow of goods through a tamper-proof ledger reducing illegal drug sales. Regarding RQ4 about how data owners can selectively share specific attributes with certain stakeholders in a reliable and scalable manner, the fourth article includes BBS signatures on top of proxy re-encryption to allow data owners selectively share data while maintaining the encrypted form on the blockchain. In particular, the patient acts as a data owner and creates a verifiable derived proof using zero-knowledge proof to share specific attributes with selected stakeholders. The evaluations show that the blockchain-based governance system using proxy re-encryption and BBS signatures libraries can be explored in any multi-stakeholder system with private data with minimal overhead.

6.1 Limitations

Despite the potential benefits of the proposed blockchain-based privacy-preserving system, some limitations should be considered. Firstly, concerning data storage, the system adopts an on-chain approach, where all data is encrypted and stored on the blockchain. While this approach provides privacy and security for the data, it may raise concerns regarding scalability as the amount of data stored on the blockchain increases. Furthermore, the on-chain approach may also lead to increased costs associated with maintaining and updating the blockchain infrastructure. Regarding cryptographic functions using proxy re-encryption, BBS signatures, and the operations over encrypted data are performed in an off-chain fashion. Therefore, the proposed solution stores the encrypted data on the blockchain, and the cryptographic operations are on the client side. The research did not evaluate address privacy offered by anonymity protocols like zero-knowledge proofs used in Zcash and ring signatures utilized in Monero. These protocols maintain the anonymity of transactions concerning the sender, receiver, and the amount involved.

On the other hand, the presented study focuses on preserving sensitive data privacy using smart contracts with the delegation and selective sharing mechanism. From an implementation perspective, this study utilized open-source cryptography tools and blockchain platforms, particularly NuCypher pyUmbral PRE with OpenSSL and Cryptography.io, for proxy re-encryption operations. For BBS functions, it employed MATTR JSON-LD library using BLS12-381 key pairs. However, the research did not examine the quantum resistance of these algorithms.

Additionally, the rise of electronic communications in healthcare necessitates the development of international standards for electronic prescriptions. These standards will ensure safe medication dispensing and administration, accommodate international travel and adhere to various jurisdictional laws. ISO 17523:2016 defines the requirements that apply to electronic prescriptions (International Organization for Standardization, 2023). It describes generic principles that are considered important for all electronic prescriptions. However, this research did not strictly follow this standard in the smart contract design.

6.2 Future Work

In addition to the proposed system, different approaches to sharing sensitive information can be evaluated for compliance with the General Data Protection Regulation (GDPR). One potential scheme is to combine blockchain, smart contracts, and proxy re-encryption with peer-to-peer off-chain storage, such as the InterPlanetary File System (IPFS). It enables the design of a system for sharing different file sizes in the healthcare sector, such as Digital Imaging and Communications in Medicine (DICOM) while protecting the data owner's privacy. In addition, different privacy technologies can be investigated and compared, such as homomorphic encryption, differential privacy, and multi-party computation (MPC) protocols. Moreover, using these protocols, different consensus algorithms can be evaluated in terms of scalability.

From an identity ownership perspective, self-sovereign identity (SSI) using zero-knowledge proofs integrated with the above technologies can provide identity management in sensitive applications that require a high level of privacy. In addition, other research opportunities involve integrating blockchain and machine learning algorithms, such as federated learning to provide a decentralized computation enabling the data owners' consent to share information with machine learning models while ensuring compliance with GDPR. Furthermore, incorporating these privacy technologies with token economics (*tokenomics*) in the healthcare ecosystem can allow stakeholders to offer new services and products to clients in a decentralized manner.

BIBLIOGRAPHY

ALDUGHAYFIQ, B.; SAMPALLI, S. Digital health in physicians' and pharmacists' office: a comparative study of e-prescription systems' architecture and digital security in eight countries. **Omics: a journal of integrative biology**, Mary Ann Liebert, Inc., publishers 140 Huguenot Street, 3rd Floor New . . . , v. 25, n. 2, p. 102–122, 2021. Available: <https://doi.org/10.1089/omi.2020.0085>. Citation on page 15.

BUTERIN, V. *et al.* A next-generation smart contract and decentralized application platform. **white paper**, v. 3, n. 37, p. 2–1, 2014. Accessed: 2023-02-01. Available: <https://cutt.ly/l91KF7R>. Citation on page 14.

HEWA, T.; YLIANTTILA, M.; LIYANAGE, M. Survey on blockchain based smart contracts: Applications, opportunities and challenges. **Journal of Network and Computer Applications**, v. 177, p. 102857, 2021. ISSN 1084-8045. Available: <https://www.sciencedirect.com/science/article/pii/S1084804520303234>. Citation on page 14.

International Organization for Standardization. **Health informatics — Requirements for electronic prescriptions**. 2023. <https://www.iso.org/standard/59952.html>. Accessed: 2023-08-05. Citation on page 61.

KAKARLAPUDI, P. V.; MAHMOUD, Q. H. A systematic review of blockchain for consent management. **Healthcare**, v. 9, n. 2, 2021. ISSN 2227-9032. Available: <https://www.mdpi.com/2227-9032/9/2/137>. Citation on page 13.

KSIBI, S.; JAIDI, F.; BOUHOULA, A. A Comprehensive Study of Security and Cyber-Security Risk Management within e-Health Systems: Synthesis, Analysis and a Novel Quantified Approach. **Mobile Networks and Applications**, p. 1–21, 2022. ISSN 1383-469X. Available: <https://doi.org/10.1007/s11036-022-02042-1>. Citation on page 13.

MUKTA, R.; PAIK, H. young; LU, Q.; KANHERE, S. S. A survey of data minimisation techniques in blockchain-based healthcare. **Computer Networks**, v. 205, p. 108766, 2022. ISSN 1389-1286. Available: <https://www.sciencedirect.com/science/article/pii/S1389128622000044>. Citation on page 13.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. May 2009. Available: <http://www.bitcoin.org/bitcoin.pdf>. Citation on page 14.

PENG, L.; FENG, W.; YAN, Z.; LI, Y.; ZHOU, X.; SHIMIZU, S. Privacy preservation in permissionless blockchain: A survey. **Digital Communications and Networks**, v. 7, n. 3, p. 295–307, 2021. ISSN 2352-8648. Available: <https://www.sciencedirect.com/science/article/pii/S2352864819303827>. Citation on page 14.

QAHTAN, S.; YATIM, K.; ZULZALIL, H.; OSMAN, M. H.; ZAIDAN, A.; ALSATTAR, H. Review of healthcare industry 4.0 application-based blockchain in terms of security and privacy development attributes: Comprehensive taxonomy, open issues and challenges and recommended solution. **Journal of Network and Computer Applications**, v. 209, p. 103529, 2023. ISSN 1084-

8045. Available: <https://www.sciencedirect.com/science/article/pii/S1084804522001709>. Citation on page 13.

QI, M.; WANG, Z.; HAN, Q.-L.; ZHANG, J.; CHEN, S.; XIANG, Y. Privacy protection for blockchain-based healthcare iot systems: A survey. **IEEE/CAA Journal of Automatica Sinica**, p. 1–20, 2022. Available: <https://doi.org/10.1109/JAS.2022.106058>. Citation on page 14.

SZABO, N. Formalizing and securing relationships on public networks. **First monday**, 1997. Available: <https://doi.org/10.5210/fm.v2i9.548>. Citation on page 14.

UDDIN, M. A.; STRANIERI, A.; GONDAL, I.; BALASUBRAMANIAN, V. A survey on the adoption of blockchain in iot: challenges and solutions. **Blockchain: Research and Applications**, v. 2, n. 2, p. 100006, 2021. ISSN 2096-7209. Available: <https://www.sciencedirect.com/science/article/pii/S2096720921000014>. Citation on page 13.

VEJDANI, M.; VARMAGHANI, M.; MERAJI, M.; JAMALI, J.; HOOSHMAND, E.; VAFAEE-NAJAR, A. Electronic prescription system requirements: a scoping review. **BMC Medical Informatics and Decision Making**, BioMed Central, v. 22, n. 1, p. 1–13, 2022. Available: <https://doi.org/10.1186/s12911-022-01948-w>. Citation on page 15.

WANG, D.; ZHAO, J.; WANG, Y. A Survey on Privacy Protection of Blockchain: The Technology and Application. **IEEE Access**, v. 8, p. 108766–108781, 2020. ISSN 21693536. Available: <https://doi.org/10.1109/access.2020.2994294>. Citation on page 14.

WAZID, M.; DAS, A. K.; MOHD, N.; PARK, Y. Healthcare 5.0 Security Framework: Applications, Issues and Future Research Directions. **IEEE Access**, v. 10, p. 129429–129442, 2022. ISSN 2169-3536. Available: <https://doi.org/10.1109/access.2022.3228505>. Citation on page 13.

ICMC USP

SÃO CARLOS