A P2P Optimistic Fair Exchange (OFE) Scheme For Personal Health Records Using Blockchain Technology

By

Nasim Al Goni

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2020

A P2P Optimistic Fair Exchange (OFE) Scheme For Personal Health Records Using Blockchain Technology

by

Nasim Al Goni

APPROVED BY:

B. Balasingam
Department of Electrical and Computer Engineering

S. Samet
School of Computer Science

S. Saad, Advisor
School of Computer Science

Jan 17, 2020

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I hereby declare that this thesis incorporates material that is a result of research conducted under the supervision of Dr. Sherif Saad (Advisor). Dr. Ahmed Ibrahim contributed in revising the publication. In all cases, the key ideas, primary contributions, experimental designs, data analysis, interpretation, and writing were performed by the author, and the contribution of co-authors was primarily through providing feedback on the refinement of ideas and editing of the manuscripts.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

This thesis includes one original paper that has been accepted for publication in a conference:

Publication title/full citation	**Publication status**
Nasim Al Goni, Sherif Saad, and Ahmed Ibrahim, "A P2P Optimistic Fair Exchange (OFE) Scheme For Personal Health Records Using Blockchain Technology", in WIDECOM 2020	Accepted

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada

Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

In today's digital world, it is common to exchange sensitive data between different parties. There are many examples of sensitive data or documents that require a digital exchange, such as banking information, insurance data, health records. In many cases, the exchange exists between unknown and untrusted parties. Therefore, it is essential to execute the data exchange over a fair non-repudiation protocol. In digital communication, non-repudiation is undeniable evidence of one's responsibility regarding the validity of any data he shares/receives. Usually, this is achieved by the use of a cryptographic digital signature. In this case, the parties cannot deny the authenticity of their digital signature. The protocol satisfies the fairness property if and only if it does not give the sender any advantages over the receiver or vice versa, at any step during the exchange process. Combining fair exchange and non-repudiation for digital exchange is critical in many applications and can be acquired with or without the involvement of any trusted third party (TTP). However, without the involvement of TTP, fairness becomes probabilistic, and the involvement of TTP can cause significant dependency on the third party. Therefore, a peer-to-peer (P2P) (aka offline) fair non-repudiation protocol that does not require a trusted third-party is desirable in many applications. Blockchain is designed in such a way that the network can handle the trustless environment and deliver the correct result. Thus, if the exchanges are done leveraging Blockchain, it will ensure true fairness, and at the same time, none of the participants have to deal with the trust issue. In this thesis we propose a P2P fair non-repudiation data exchange scheme by leveraging Blockchain and distributed ledger technology. The scheme combines on-chain and off-chain communication patterns to enable the exchange of personal health records between patients and healthcare providers. We provide an informal reasoning of the proposed scheme. Moreover, we propose a design and implementation agnostic to existing Blockchain platforms to enable unbiased evaluation of the proposed scheme. Finally, we make a comparative analysis of the result derived from our approach with the existing one.

DEDICATION

I want to dedicate this thesis to my beloved family (my parents and my dear brother), who always encouraged me to go for my dream and always beside me through thick and thin. I also want to dedicate this work to two of my dearest friends who always motivated me in stressful situations.

ACKNOWLEDGEMENTS

First of all, thanks to Almighty for helping me in pursuing my Master's degree. Secondly, I want to express my sincerest appreciation to my supervisor, Dr. Sherif Saad, for his constant support, patience, and inspiration during my Master's studies at the University of Windsor. I am deeply honored working under his supervision and be a part of his research team. Thank you, Professor, so much for allowing me to learn from you. Next, I would also like to express gratitude to my thesis committee members, Dr. Balakumar Balasingam and Dr. Saeed Samet, for their beneficial advice and valuable suggestions to my thesis. Also, I am greatly thankful to my colleagues and friends at the University of Windsor for helping me in various aspects of the research and my life as a student during the past one and a half years.

Finally, I would like to express my highest appreciation to my family for all the unconditional love, support, patience, encouragement, and kindness they gave me during my whole life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

A healthcare system can be characterized as the technique by which healthcare is financed, sorted out, and conveyed to a population. It incorporates issues of access (for whom and to which services), uses, and resources (healthcare workers and offices). The objective of a healthcare system is to improve the health of the population in the best way conceivable, considering a society's available resources and contending needs. By the start of the twenty-first century, access to healthcare had come to be addressed by most nations as fundamental human rights.

A healthcare system, is hence, more than the pyramid of publicly possessed offices that convey individual health services. It incorporates, for instance, private caregivers, behavior change programs, vector-control campaigns, medical coverage associations, and occupational health and safety legislation.

The advancement of innovations has profoundly affected conventional healthcare practices. Among the upsides of innovation in healthcare, the decrease in preventable death cases, general improvement of patient well-being, a significant reduction in treatment and recovery time, and the rise of new employment opportunities for medical workers are noteworthy. A by-product of technological advancement that affected the healthcare system in a positive manner is the digitalization of health records, which leads to higher patient care, improved public health, and ease of workflow.

Although healthcare services has customarily been structured around the care providers, situated in organizations, for example, medical clinics, hospitals or specialists' offices, thanks to the technological advancement, and because of the demands, the practice of keeping the services to institution-centric is now gradually shifting towards patient-centric (Emily et al. [42] referred as "shift left"). By creating community-based treatment services, mobile units providing care when and where required, and telemedicine, with shift left, the care goes to the patients instead of patients coming to the care-providers. The paradigm of shifting the services in the left direction will not only reduce the cost but also improve the overall quality of life (figure 1.0.1).

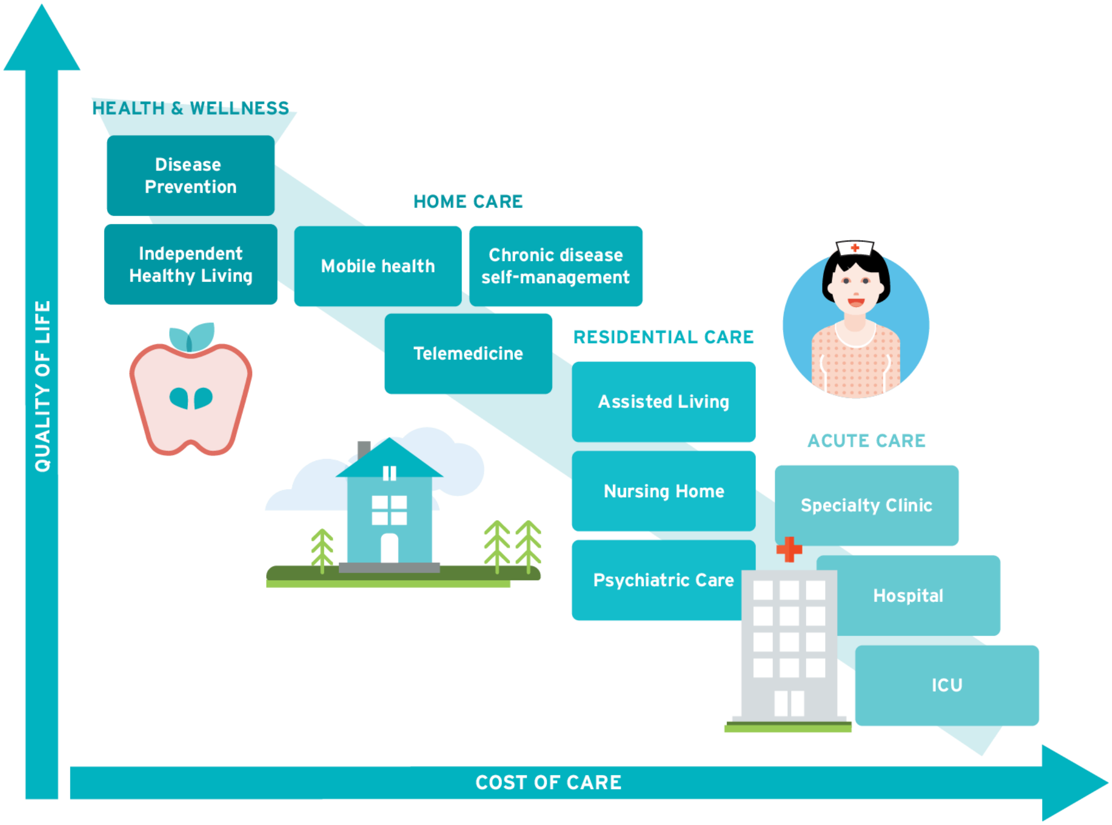


Fig. 1.0.1: Shifting left will help in both ways, reduce the cost of care as well as improve the quality of life [42]

Nevertheless, the confrontation of numerous technological progressions postures new difficulties. Some of the critical challenges, which can be categorized into non-technical and technical in general, currently faced by the healthcare industry are:

Non-Technical Challenges:

- *Patient Experience.* Patients are now looking forward to having a streamlined patient experience so that they avail "self-service" to find the answers of most questions, issues, or concerns (e.g., downloading an immunization record, booking an appointment, taking care of their bills, or checking their record/insurance status) at whenever, wherever, and however is most convenient for them. For healthcare organizations offering services from various areas, it is additionally significant for each employee to have the current patient data. Not only will it convey a superior patient experience but also help to avoid fatal mishaps such as drug interactions.

- *Invoicing and Payment Processing.* To meet patient desires and improve the client experience, healthcare providers have to ensure the billing system is patient-friendly. They should offer paperless articulations, and a diversify payment methods (e.g., eCheck, Visa) through an online patient portal and use the most modern payment systems, for example, mobile and text-to-pay. However, at the same time, healthcare providers are required to pursue strict rules to protect patient information. They have to guarantee that their payment portal and processing system are entirely compliant; otherwise, they risk incurring a huge penalty.

- *Price Transparency.* Numerous patients are now examining price estimation for different services before settling on a choice. Any system that does not make their pricing public might be dropped from even consideration at first. Price transparency became an important buzzword in 2019 and probably be a mighty issue to look into in the coming years.

Technical Challenges:

- *Cybersecurity.* Because of the exceptionally delicate patient data gathered by healthcare organizations, the industry has become an ideal target for cybercriminals. In 2017, the US medical and healthcare sector experienced more than 350 information ruptures, uncovering 4.93 million patient records [106]. Sadly, this pattern gives no indications of backing off. In the first half of 2019, there as of now were 32 million patient records broke [31]. At the point when a breach happens, in addition to the fact that one is compromising classified patient data, healthcare providers additionally face a hefty punishment if they found to have disregarded the many compliant standards of the industry.

- *Medical records accessibility.* Even though more medical information is being produced regularly, it is dissipated over various parties and their systems, including payers, suppliers, and patients. There is no single "source of truth" that a healthcare provider can use to enhance the patient experience.

  For example, when patients switch insurance plans or healthcare providers, most practices depend on patients' self-reporting to reproduce their records. Accordingly, not all the data are transferred appropriately, and it is challenging to harness the power of data for generating accurate insights. Also, information originates from numerous sources in a variety of formats. At present, there is no single framework to recover, store, and break down information from different sources at scale.

  So, to completely use all the patient information from various sources, healthcare organizations need to actualize the non-relational information system. Thus, information from different sources can be used regardless of whether the datasets come in various formats. To do so, a recognized patient identifier is required to avoid patient data mismatch. At the same time, a thorough and transparent procedure of sharing the information is highly demanding since it will ensure the responsibility regarding the safety of the data.

Despite the mentioned challenges, the new paradigm, for arrangement of pervasive health services at reasonable costs, has been embraced by nations, for example, USA[1], Canada[1], U.K.[89], Korea [22], and European Union ([47], [55], [30]).

1.1 Healthcare Applications

Over the last decades, like all other industries, the healthcare domain is also prioritizing applications or software over the paper-based system for data management.

There can be different types of Healthcare Applications, such as:

- *Medical Practice Management (MPM) Application:* Focus on smoothing the day-to-day task of a medical facility. The goal is to make sure practitioners spend less time on administrative paperwork and more time on patient treatment.

  Example: Prime Suite [59], eClinicalWorks [41].

- *Health Records (HRs) Management Application:* Contains detail information of a patient such as demographics, medical history, laboratory results, allergies, etc. This information is required to be shared among the physicians in order to provide more accurate treatment.

  Example: CureMD [26], IO Practiceware [73]

- *E-prescribing Management Application:* Instead of sending ambiguous handwritten notes, it allows medical providers to send clear, accurate, and understandable prescriptions.

  Example: DrChrono [39], DrFirst [40]

- *Hospital Management Application:* Deals with the management of patient data, doctor and medical staff information, and hospital billing.

  Example: SoftClinic [105], Practo's Insta [67]

- *Healthcare Customer Relationship Management (CRM) Application:* Helps healthcare organizations collect, progress, and manage customer relationships more efficiently and effectively.

  Example: Deskera [34], DocEngage [36]

Several healthcare providers and insurance agencies today utilize one or the other form of electronic version of medical record systems, which is why we are focusing on healthcare management applications.

1.1.1 e-Health, EHRs, and PHRs

There has been a ton of research in the electronic healthcare area with an emphasis on using the electronic patient records for patient monitoring and diagnosis. The arrangement of health services utilizing digital innovation has been named as *e-Health* [102]. Also, conventional clinical settings with paper-based medical records and remedies have likewise progressed to the *Personal Health Records* (PHRs) and the *Electronic Health Records* (EHRs), an electronic adaptation of patient healthcare data. The PHRs are constrained by patients themselves [69] while; the EHRs are overseen by the healthcare providers [61]. The e-Health requires the entire remaking and digitization of the healthcare infrastructure, including generation, supply, and management [22].

1.2 Requirements of Healthcare Applications

1.2.1 General Requirements

Since modern healthcare systems moving towards paperless, they tend to reduce significant workforce requirements and become a very cost-effective way of treating patients. Also, physicians can have all the information in one place, which leads to better treatment and thus increases efficiency.

A healthcare management application should possess functionalities like Data management, Patient history, Patient scheduling, E-prescribing, etc.

- *Data Management:* Medical practitioners would be allowed to add and store patient information electronically. Also, other physicians would be able to view and/or modify it.

- *Patient History:* The application should stores information about existing problems, allergies, medications, etc.

- *Patient Scheduling:* It may allow the medical providers to schedule patients easily, register them, and choose a reason for their visit.

- *E-prescribing:* Instead of sending ambiguous handwritten prescriptions, the application may allow medical providers to send prescriptions to pharmacies electronically.

1.2.2 Security and Privacy Requirements

A patient may have numerous healthcare service providers, including primary care physicians, specialists, and therapists. Moreover, a patient may enroll with several medical coverage organizations for various sorts of insurance, for example, medical, dental, and vision [116]. Subsequently, the health records of a patient may exist in the database of different caregivers in the healthcare services community. From the clinical point of view, it is essential to get to the present-day patient health data [114]. Nonetheless, sharing and coordination of the records, that are overseen by several service providers are slow and expensive [116] and requires viable, secure, and minimal cost to share among the providers.

Recent trends in healthcare, fixating on getting to the data anytime and anywhere, energize moving the healthcare systems towards more patient-centric rather than institution-centric. Even though records sharing offers a considerable advantage, it also entails threats as far as privacy and security [38] of data. The idea of privacy-preserving is extensively more than merely keeping up the confidentiality of information. Metri et al. [80] argue that threats to data protection include spoofing identity, messing with the data, denial of data access, and data divulgence.

In spoofing, the assailant claims to be a legitimate client, while information altering includes noxious adjustments and change of the content. Repudiation is concerned with the users who deny his role after performing an action within the system. Data divulgence is the presentation of data to the entities who have no privilege to get the data [80]. Thus parties involving data sharing should be aligned with the governmental rules and regulations regarding that. In the United States, for instance, use and exposure of the Protected Health Information (PHI) ought to be as per the necessities of the Health Insurance Portability and Accountability Act (HIPAA). The HIPAA requires that keeping up the privacy of the healthcare information is not an alternative, yet a commitment [1]. In Canada, according to the Personal Information Protection and Electronic Documents Act (PIPEDA), institutions must get an individual's consent if they collect, use or disclose that individual's personal information [88] (among others, medical records also fall under the umbrella of personal information).

In case of storing or trading health records over the internet, according to Abbas et al. [1], eight security and privacy criteria are required, namely, Integrity, Confidentiality, Authenticity, Accountability, Audit, Non-repudiation, Anonymity, and Unlinkability.

- *Integrity.* The data stored in the system is the exact representation of the intended information and is authentic when presented to someone. It is not altered and, if required, edited by an authorized person for the right reason in a corrected manner, keeping the record of edition for auditing purposes.

- *Confidentiality.* The system should have a robust mechanism to keep the record in a secured position, which is inaccessible to an unauthorized party.

- *Authenticity.* The system should make sure that only the authentic individual is getting access to the information, and the information provided to the requester is accurate.

- *Accountability.* Accountability means the obligation of an individual on entrusted property. The individual will be accountable for his/her activities, accept responsibility, and transparently disclose the results.

  The healthcare applications should have a mechanism of assigning individuals who will take responsibility for his/her actions. The patient should be able to monitor who has accessed their data and up to what level. If agreed upon, the patient can decide who can have his/her data.

- *Audit.* Audit is a systematic review of the security of an organization's data by estimating how well it adapts to a set of established criteria.

The healthcare applications should ensure that all the healthcare data is secure, and all the data access activities are being monitored.

- *Non-repudiation.* The system ensures that there is no denial of action by any person after performing any activity with the stored data. It is a by-product of accountability; one should be liable for his action.

- *Anonymity.* The system should have a technique to hide the identity of a patient or participant. No outsider (or even insider in some particular cases) can trace back any individual based on the generated health data stored in the system.

- *Unlinkability.* To ensure that, no matter how many times a requester requests for health data, he cannot be traced back by others with the help of the request to send it the first time. In other words, the information flow will not derive any idea about the user to a third party.

Table 1.2.1 stated the requirements for common healthcare applications.

Security and Privacy Requirement	**Medical Practice Managemen Application**	**Health Records Management Application**	**E-prescribing Management Application**	**Hospital Management Application**	**Healthcare Customer Relationship Management Application**
Integrity	Required	Required	Required	Required	Required
Confidentiality	Required	Required	Required	Required	Required
Authenticity	Required	Required	Required	Required	Required
Accountability	Required	Required	Required	Required	Required
Audit	Required	Required	Required	Required	Required
Non-repudiation	Required	Required	Required	Required	Required
Anonymity	NA[a]	Required	NA	NA	NA
Unlinkability	NA	Required	NA	NA	NA

[a] NA = Not Applicable

Table 1.2.1: Security and privacy requirements for common healthcare application

There is no specific classification on the approaches for privacy-preservation, which, in turn, can guarantee the fulfillment of one or more above mentioned requirements. However, Abbas et al. [1] classified the approaches into two groups at the top level.

- *Cryptographic.* Handles the privacy risks utilizing specific encryption schemes and cryptographic primitives.
- *Noncryptographic.* Uses policy-based authorization models which allows the data having access control policies.

The cryptographic approaches are again subdivided into three groups.

- Public Key Encryption (PKE)
- Symmetric Key Encryption (SKE)
- Alternative cryptographic primitives

Figure 1.2.1 illustrates the whole taxonomy.

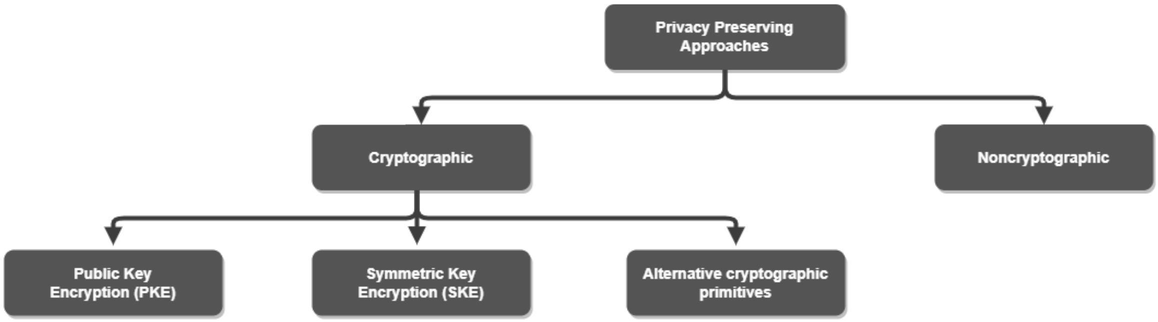

Privacy Preserving Approaches
Cryptographic
Noncryptographic
Public Key Encryption (PKE)
Symmetric Key Encryption (SKE)
Alternative cryptographic primitives

Fig. 1.2.1: Taxonomy of the privacy preserving approaches [1]

1.3 Non-repudiation and Fair Exchange

1.3.1 Non-repudiation

While issues, for example, integrity, confidentiality, authentication, and access control, have been considered seriously, enthusiasm for non-repudiation conventions has just come as of late [71].

As stated, Non-repudiation is the ability to provide irrefutable evidence of one's responsibility regarding the validity of any data he shares/receives. Because patients want to access their medical records and exchange that with other care providers for their benefit, the role in the exchange and the usage of that data is an important issue here. Thus non-repudiation can make sure participants (Alice and Bob), engaging in the exchange of medical records, will have evidence of their participation in the procedure. This evidence will justify and balanced out all the petition in the future. Thus, along with sharing the information, participants will provide *Proof of Origin* (POO) and *Proof of Receipt* (POR) of the information as an indication of their engagement. If there should arise an occurrence of disagreement (e.g., Alice denying having sent given information or Bob denying having it), an adjudicator can assess these confirmations and decide for one of the parties without any ambiguity.

- *Proof of Origin.* A proof of origin is a non-repudiation origin of data which, when presented to any adjudicator, will unambiguously guarantee that whether or not the data was indeed originated by Alice (the sender).

- *Proof of Receipt.* A proof of receipt is a non-repudiation receipt of data which, when presented to any adjudicator, will unambiguously guarantee that whether or not the data was indeed received by Bob (the receiver).

Nevertheless, the exchange of POO and POR, in order to maintain the non-repudiation criteria, must be done in a fair manner.

1.3.2 Fair Exchange

A protocol that ensures, during the exchange of items between two parties, neither of the party will be in any favorable situation. That is, the exchange has to be atomic. At any point during the exchange, the process will ensure that either both the parties will get what they want, or none of them will get anything.

In terms of medical records exchange, Alice will send her records and POO of that records if and only if she gets or have the guarantee to get the POR from Bob, and Bob will send the POR of that records only when he gets the records and POO of that records.

1.4 Problem Definition

In a paper-based scenario, the exchange of non-repudiation evidence is easy to achieve because both parties will be physically available at the same time. However, it is not as simple when the same exchange has to be done over a computer network. As a matter of fact, Even et al. [98] proved that achieving fairness in a deterministic two-party signing protocol is impossible because information exchange over the computer network is non-simultaneous.

Researchers have proposed many alternative solutions to maintain a strong fair exchange protocol in digital exchange. A strong fair exchange protocol does not require a human judge, and if any disputes occur, it will be handled within the scope of the transaction. On the other hand, a weak fair exchange can not offer any such solution. However, it can gather proof so that a misbehaving party can be identified. The protocol assumes that the misbehaving party can be brought to justice [93].

Strong fair exchange can be achieved in many ways, with or without involving any third party. However, without any involvement of a third party, it mostly remains probable while the involvement of the third party brought the dependency and trust issue.

A P2P (aka offline) fair non-repudiation protocol that does not require a trusted third-party while electronically exchanging large-sized sensitive medical records concurrently maintaining the confidentiality of the records is desirable in many healthcare applications, which cannot be served by either of the above proposition.

1.5 Contribution

One solution to achieve the above goal (a fair non-repudiation protocol that does not require a trusted third party) can be done by involving distributed ledger technology (aka Blockchain). Blockchain, by design, come up with a technique where one does not have to depend on a single party or intermediary to hold up the data. Thus, instead of putting trust in a third party, the participants can take advantage of using a trustless Blockchain network. Even though the researchers have already started working on this issue, none of them are focused on maintaining fairness while exchanging medical records. Thus, the contributions to this study are,

- Proposing a scheme that will highlight maintaining true fair-exchange policy without any involvement of trusted third party while exchanging medical records.

- Utilizing off-chain communication protocol to enable the exchange of personal health records in a P2P manner which in turns reduces the storage overhead on the shared distributed ledger.

- Use a platform-agnostic approach while implementing the scheme which will provide a reference implementation that could be used by other research teams to test new fair-exchange schemes that will take advantage of Blockchain.

- A comparative analysis of the result derived from the proposed approach with the existing one.

1.6 Organization of the rest of the thesis

The rest of the thesis is organized as follows:

- Chapter 2 layouts some basic cryptographic techniques which are necessary to ensure fairness in digital exchange along with a detail description of ways to maintain fairness in digital exchange.
- Chapter 3 presents the related and previous works in the field.
- Chapter 4 explains the concepts of the Blockcahin and how it works.
- Chapter 5 introduces the proposed approach and model it at a high level followed by an informal reasoning of each step of the proposed approach.
- Chapter 6 presents the design and implementation of the proposed scheme and experimental results obtained from that.
- Chapter 7 lists the conclusion and discusses future work.

CHAPTER 2

Preliminary

Consider a naive protocol, for example, where Alice sends a signed message (M) to Bob, who answers with a signed receipt for the given message. If none of them trust each other, this convention is not appropriate, as Bob may not send the subsequent message. The convention could be modified in an accompanying manner: Alice sends a guarantee to the message to Bob, who answers with a receipt, and, in the third step, Alice sends the message itself to Bob. Here, we have another issue, as in this time, it is Alice who is in a favorable position, being the first to acquire her total proof, and henceforth could decline to send the last message.

Even though ensuring fairness over a digital exchange is hard to achieve, we can meet the goal by leveraging some facts. Nevertheless, in order to understand how the different approaches of ensuring fairness for digital exchange works, we need to know some cryptographic techniques which are used in these approaches. Thus, in this chapter, those cryptographic techniques followed by a well-detailed explanation of the different approaches of ensuring fairness are clarified.

2.1 Cryptographic techniques required in ensuring fairness

2.1.1 Secure Hash Algorithms

A secure hash algorithm converts the arbitrary input-length data to a predefined fixed-length output. No matter how big or small in size, the original data is, the size of the hash value of that original data is always the same. Two essential characteristics of a secure hash function are:

- A hash algorithm will always generate the same hash value for the same input data. However, it will generate a completely different value even if a slight modification is made on the original data.

- A secure hash function is a one-way function. One can only get the hash value from the original data, but will not get (or even guessed) the original data from the hash value. That is, the function can not be reversed engineered.

The following examples will give a better understanding of secure hash.

Example 1.

Original Text: Hello

Hash value:

185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969

Example 2.

Original Text: This is a hash value

Hash value:

b3118a3288630e2591f2f05f5f78e39e2f44257f8de3f409f2af2fa8c13d53bb

Example 3.

Original Text: hello

Hash value:

2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

Here, three original texts are applied over a *SHA-256* hash function. Even though the original texts are different in size (see example 1 and 2), the size of the hash value as output is always the same. Also, because of a slight modification in the original text of example 3 from 1 (H in example 1 is capital, whereas, in 3, it is small), two completely different hash values are generated.

A hash algorithm is suppose to give one output for one input. If a hash algorithm generates same output for two different inputs, it is called *collision*. Unfortunately, recently, some of the algorithms failed to be total collision resistant such as MD5 and SHA1 [82].

2.1.2 Symmetric Encryption

Symmetric encryption is a technique where both encryption and decryption can be done by the same key. By applying any symmetric encryption algorithm, data will be converted into a form that is not human-understandable. To have the original data from the encrypted data, one has to have the encryption key and decipher the encrypted records with that key (figure 2.1.1). Examples of symmetric encryption algorithms are AES, DES, and Blowfish.

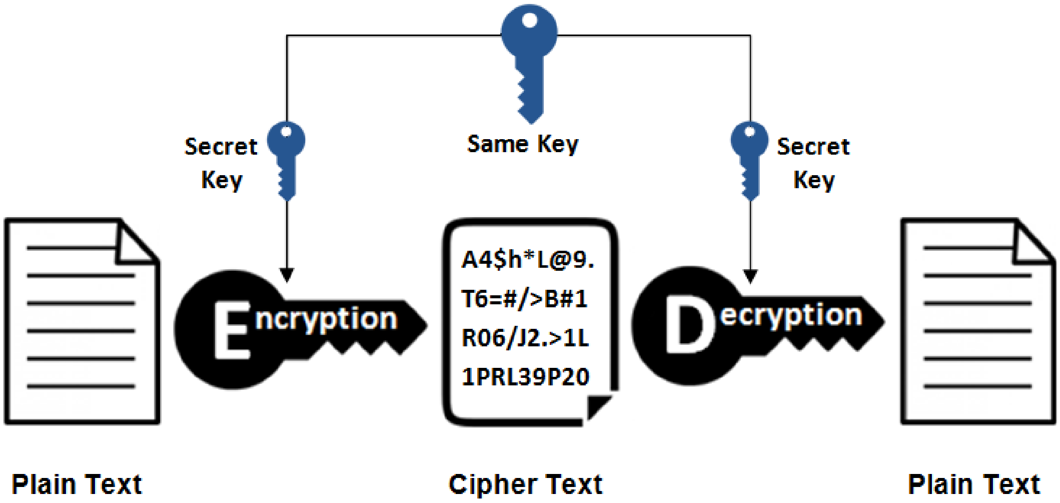


Fig. 2.1.1: How symmetric encryption works [66]

The below example explains symmetric encryption technique. In the example, we encrypted an original text “Hello” with a symmetric key AES-256 (base64 encoded). Thus, we get a cipher text “rVjJBGgF6FIKmQO2UANOkQ==”. Later, we decrypt the cipher text with the same key and get the original text “Hello”.

Encryption:

Original Text: Hello

Key (AES-256): 17Brx9O3eP44AEV84TTmBVxTCs7TCSJg1uMVpcDNlwI=

Cipher text: rVjJBGgF6FIKmQO2UANOkQ==

Decryption:

Cipher Text: rVjJBGgF6FIKmQO2UANOkQ==

Key (AES-256): 17Brx9O3eP44AEV84TTmBVxTCs7TCSJg1uMVpcDNlwI=

Original text: Hello

2.1.3 Asymmetric Encryption

Asymmetric encryption is a technique where encryption and decryption are done by two different keys. Here an asymmetric key pair, commonly known as public-private key pair, originated by using some mathematical formulas, is used for encryption and decryption. Every public key has its correspondent private key, which is extremely difficult to guess. If a message is encrypted with a public key, it can only be decrypted by its corresponded private key and vice-versa. Thus, if someone kept the private key to himself and shares the public key to others, the sender of the message can encrypt the message with the public key, and only the right recipient (owner of the private key of that correspondent public key) can decrypt the message. In this way, the message can be exchanged securely. RSA, El Gamal, and ECC are few examples of asymmetric encryption algorithms. Figure 2.1.2 illustrates how asymmetric encryption works.

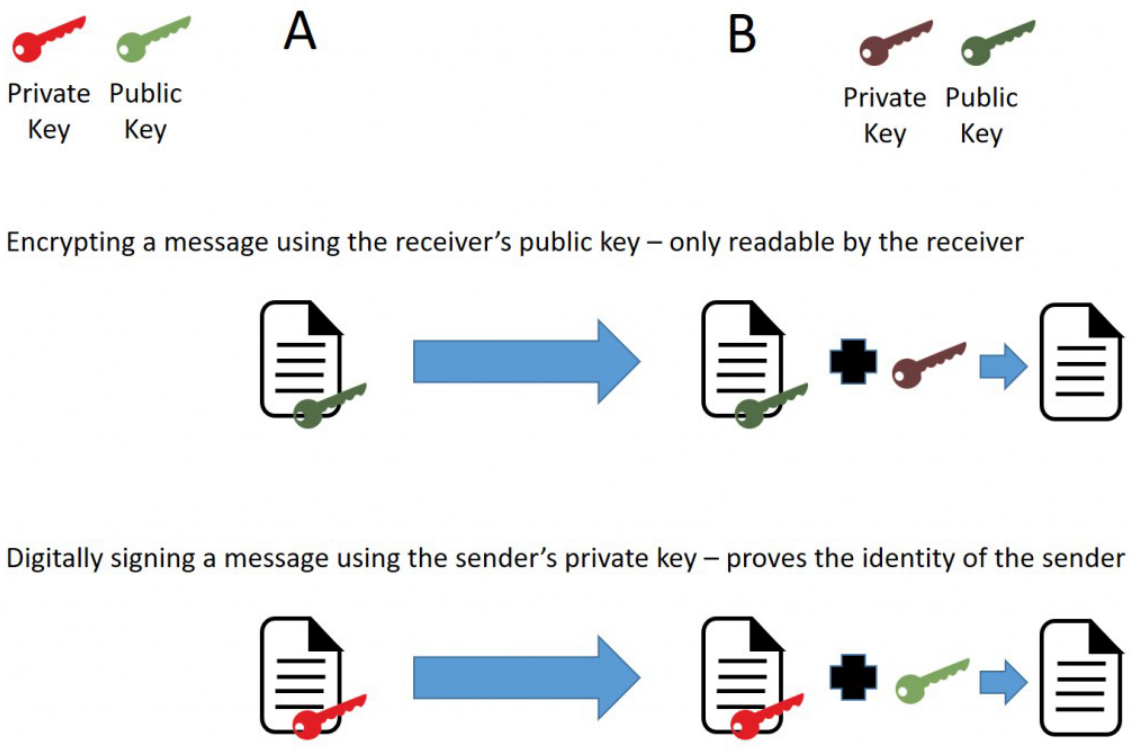

Private Key
Public Key
A
B
Private Key
Public Key
Encrypting a message using the receiver's public key – only readable by the receiver
Digitally signing a message using the sender's private key – proves the identity of the sender

Fig. 2.1.2: How asymmetric encryption works [115]

For convenience, we are explaining the asymmetric encryption with the below example as well. Here, two keys (public and private) are generated by using the RSA encryption algorithm. The public key is used for encrypting the original text "Hello", which provides us a cipher text. Later on, the cipher text is deciphered with the private key to get the original text again.

Encryption:

Original Text: Hello

Public Key:

——BEGIN PUBLIC KEY——
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBALPLfv2T06sMfx7CRweN4AW
vtdL91qUlJeCgVt1ryTSiOjQXaFMTLXdIrkl9Wj4nzUAJRtrPjnu8dfVhwY+ogQ
8CAwEAAQ==
——END PUBLIC KEY——

Cipher text:

pmEUxc/PS+YyZkjnbT1j/pF62WWKTMJ7ZXDixrm7Bx+PgDwOPWd1ur

Hmwd0eSE/hcS9yL0D4NNDU7bbnqNHJbg==

Decryption:

Cipher Text:

pmEUxc/PS+YyZkjnbT1j/pF62WWKTMJ7ZXDixrm7Bx+PgDwOPWd1ur

Hmwd0eSE/hcS9yL0D4NNDU7bbnqNHJbg==

Private Key:

——BEGIN RSA PRIVATE KEY——

MIIBOgIBAAJBALPLfv2T06sMfx7CRweN4AWvtdL91qUlJeCgVt1ryTSiOjQXaFMT

LXdIrkl9Wj4nzUAJRtrPjnu8dfVhwY+ogQ8CAwEAAQJAFMuImJOse7AqU8hsprcG

HiJAiXLKjLdLNjvVUC7TSr72rMvRwN//g9k0XL70vYz6KVXIE7ULQIO3lTkzYujK

oQIhAPOf1YNq9dZqBLNgg/7Z+Nl/uXekWiuFJHQcy5vO96l7AiEAvO2asuYqBoq9

9h2u3vRE2Mos6Gy6UZ5pzWYAcc5UQH0CIGNLVsOcWZxNU6MkiEfb4VAMfbQkuVeZ

iXUFs3rKjSh3AiEAjUpUyz3+Z+4SqqEASpT7d/WbKCdIIAoMriN+aZ4YvDECIEtL

1K8f4t9/ERxa8gJgQyQGnXLqAiSAML86x9X/EXgh

——END RSA PRIVATE KEY——

Original text: Hello

2.1.4 Digital Signature

Using the concept of public-private key pair, a digital signature works to ensure the originality of any message. It works as follows:

1. Sender (aka signer) will generate his/her public-private key pair. Keep the private key to himself/herself and shares the public key with the receiver (aka verifier) of the message.

2. The sender generates a hash value of the message with a hash function and encrypts the hash value with his/her private key. He/She then sends both encrypted hash value and the message to the receiver.

3. The receiver decrypts the hash value with the public key he/she has. At the same time generates a hash value of the received message using the same hash function. Then compares the two hashes. If those hashes match, it means the message has not been forged on its way, and indeed it is originated from the sender.

Figure 2.1.3 clarifies the concept in a more helpful manner.

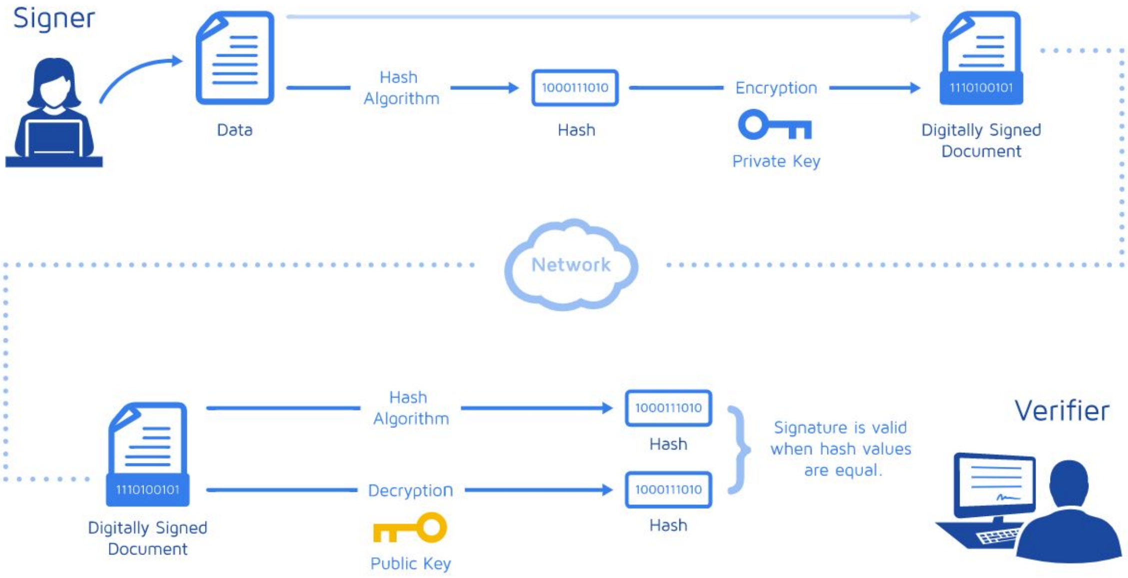


Fig. 2.1.3: How digital signature works [37]

2.2 Fair exchange protocols

Now that we have a better understanding of the required cryptographic techniques, let us deduce the approaches of maintaining fairness with these techniques. At a high level, fair exchange protocols can be categorized in two broader categories, and then one of the categories can again be subdivided into three sub-categories. Figure 2.2.1 depicted the categories.

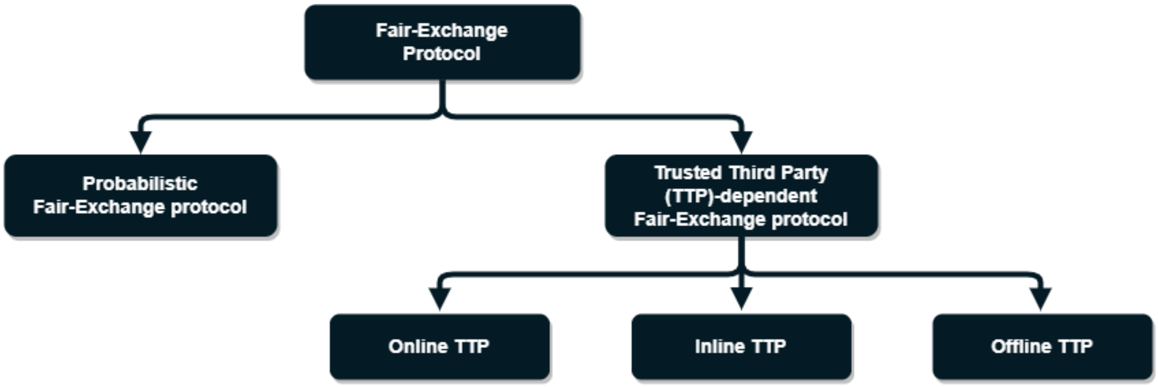

Fair-Exchange Protocol
Probabilistic Fair-Exchange protocol
Trusted Third Party (TTP)-dependent Fair-Exchange protocol
Online TTP
Inline TTP
Offline TTP

Fig. 2.2.1: Ways of the achieving fairness in digital exchange

2.2.1 Probabilistic fair-exchange protocols

The sender will divide the main message into n parts (where only the sender knows the actual value of n) and sends one part at a time along with his/her digital signature of that part (which will count as POO of that part). After receiving each part of the original message and verifying the POO, if satisfied, the receiver will provide his POR for that part. The whole process will keep running until each of the participants gets their desired document (see figure 2.2.2).

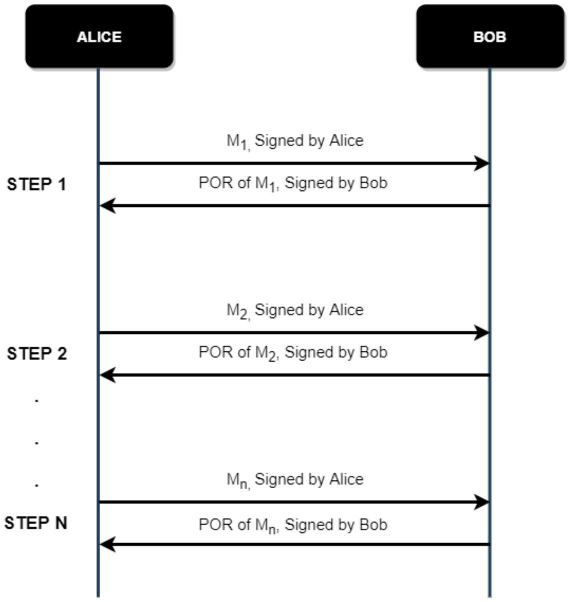


Fig. 2.2.2: Probabilistic fair-exchange protocol. Message M is divided up to n parts (M_1, M_2...... M_n).

The Probabilistic fair-exchange protocol is called probabilistic because, instead of “yes” or “no” as an outcome, the result of fairness comes as a probable manner. However, this protocol does not require a third party to interfere.

2.2.2 Trusted Third Party (TTP)-dependent fair-exchange protocol

The TTP-dependent fair-exchange protocol ensures true fairness provided that a third party, trusted by both sender and receiver, is involved during the exchange procedure. It can again be subdivided into three categories [71]:

2.2.2.1 Online TTP

The exchange of all the items (token of interest/commitment in the exchange, original message, POO, and POR) is done through TTP. There is no P2P communication between the sender and the receiver of the message. The step-by-step procedure of this protocol is as follows which is also drawn by figure 2.2.3:

1. Alice sends her message (M) with her digital signature on it to TTP.
2. TTP shares the hash of the message as a proof to Bob that he/she has the message from Alice, without revealing the original message
3. Bob sends his POR with his digital signature on it.
4. If satisfied, TTP shares the message to Bob and POR to Alice.

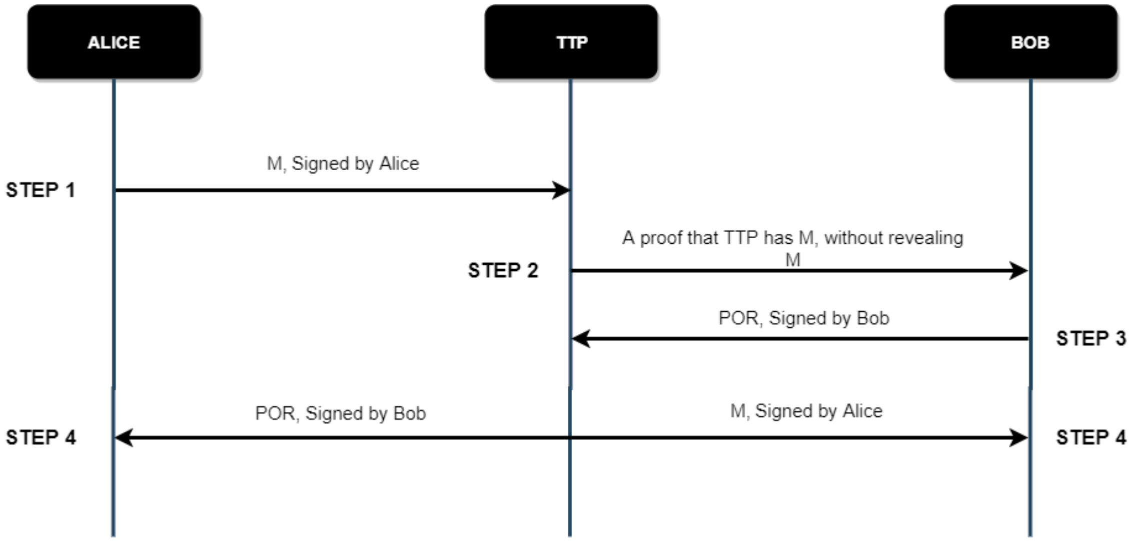


Fig. 2.2.3: Trusted Third Party (TTP)-dependent (online) fair-exchange protocol

2.2.2.2 Inline TTP

The exchange of the essential items (original message, POO, and POR) is done through TTP. However, the sender and receiver can have peer-to-peer communication and share a token of interest/commitment between them without the involvement of TTP. A general guideline of how the protocol works are as follows and also explained in figure 2.2.4:

1. Alice sends the hash of message (M), as her commitment in the exchange, without revealing the original M, to Bob.

2. Bob sends the hash of POR, as his commitment in the exchange, without revealing the original POR, to Alice.

3. Alice sends her M with her digital signature on it to TTP.

4. Bob sends his POR with his digital signature on it to TTP.

5. TTP shares M with Bob and POR with Alice.

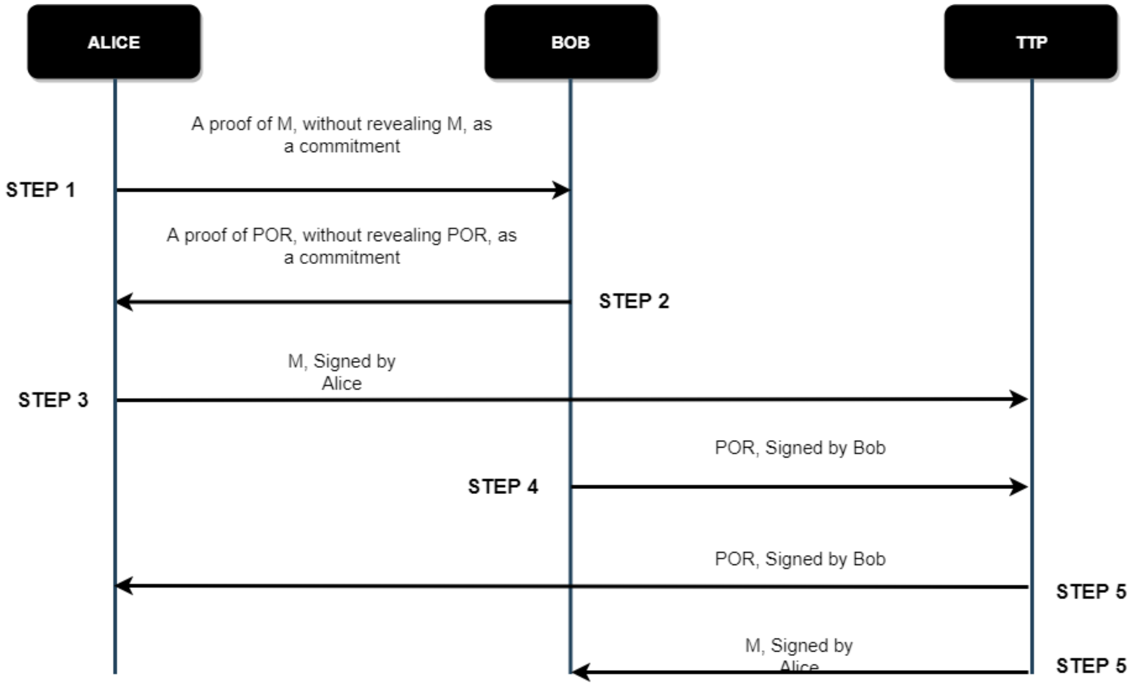


Fig. 2.2.4: Trusted Third Party (TTP)-dependent (inline) fair-exchange protocol

2.2.2.3 Offline TTP

In this protocol, the involvement of TTP is least, only when a dispute needs to resolve. Otherwise, sender and receiver can exchange all items in a P2P manner. In a nutshell, the protocol works as follows:

1. Alice encrypts the message (M) with TTP's public key, puts her digital signature on the encrypted message, and sends the encrypted digitally signed message to Bob.

2. Bob verifies the signature and makes sure that the encrypted message indeed came from Alice. Then sends his encrypted POR (encryption is done by TTP's public key) to Alice, with his digital signature on the encrypted POR.

3. Alice verifies the signature and makes sure that the encrypted POR indeed came from Bob, She then sends the original message to Bob.

4. Upon receiving the message, Bob sends the original POR to Alice with his digital signature attached.

If a dispute occurs, such that step 4 was not completed or Bob sends incorrect POR,

5. Alice presents the encryted POR she received from Bob along with her message to TTP
6. TTP decrypts the POR, sends it to Alice and at the same time sends the message to Bob

Since the participants directly communicate with each other unless any dispute occurs and at the same time, it ensures fairness, fair-exchange with offline TTP is also known as *Optimistic Fair Exchange.* Figure 2.2.5 depicts the offline fair exchange in a nicer manner.

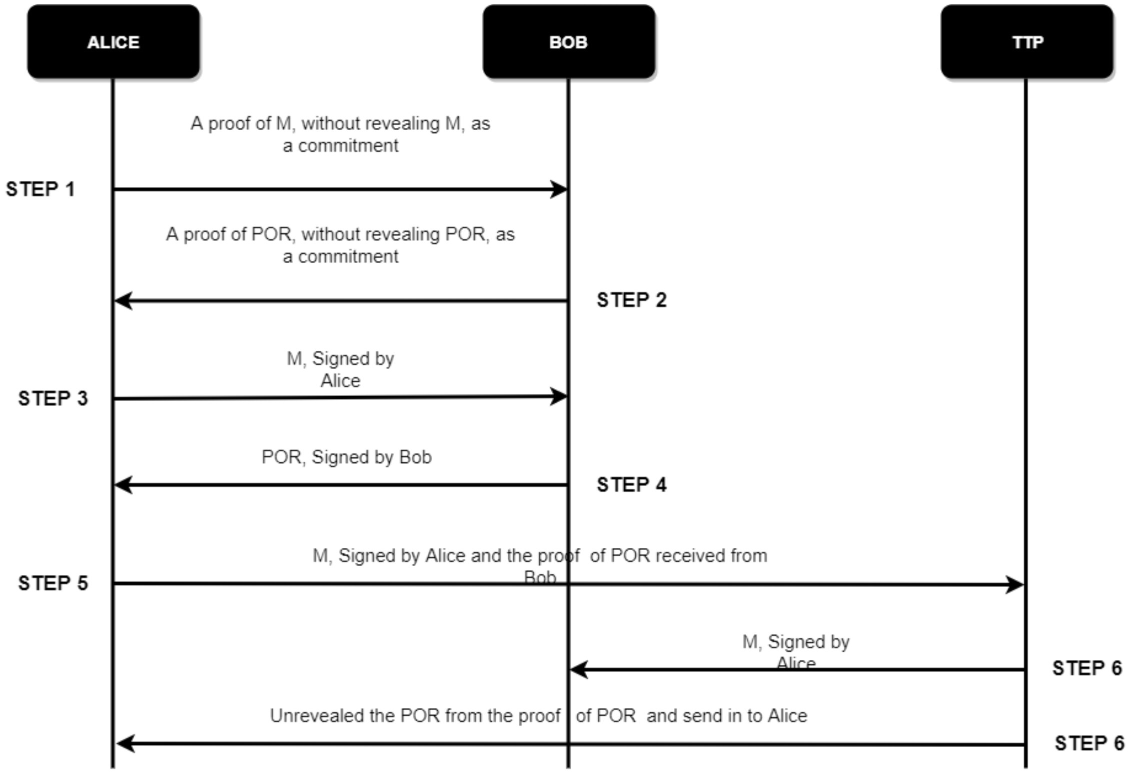

ALICE
BOB
TTP
A proof of M, without revealing M, as a commitment
STEP 1
A proof of POR, without revealing POR, as a commitment
STEP 2
M, Signed by Alice
STEP 3
POR, Signed by Bob
STEP 4
M, Signed by Alice and the proof of POR received from Bob
STEP 5
M, Signed by Alice
STEP 6
Unrevealed the POR from the proof of POR and send in to Alice
STEP 6

Fig. 2.2.5: Trusted Third Party (TTP)-dependent (offfline) fair-exchange protocol

Though a TTP-involved fair-exchange protocol ensures fairness every time, irrespective of their type, dependency on a trusted third party is a major drawback here.

CHAPTER 3

Literature Review

As healthcare services move towards a patient-centric approach, rather than institution centric, the exchange of the Personal Health Records (PHRs) among the patient and the service provider(s) is a vital factor for providing proper treatment. At the same time, the electronic exchange of information should maintain all the security and privacy criteria (according to Abbas et al. [1], Non-repudiation is one). Though a significant amount of work has been done on maintaining non-repudiation i.e., fair exchange policy for electronic exchange in some areas like contract-signing protocols ([12], [29], [98]), certified e-mail systems ([60], [65], [118]), and e-payment schemes in electronic commerce ([18], [49], [79], [90]), the PHR exchanges have not received sufficient attention.

Electronically exchanging personal health records while maintaining a fair exchange policy is trickier and cannot be achieved with the existing schemes of contract signing protocols, certified e-mail systems, or e-payment schemes in electronic commerce. This is mainly because healthcare data are *large-sized*, stored in different formats, as well as *sensitive*. That is why, unlike others, we must also consider the challenge of exchanging large-sized data and maintaining its confidentiality.

The literature review section is thus categorized into the following three groups, and in each group, the previous works in that arena are discussed.

- Fair exchange without the involvement of Blockchain
- Use of Blockchain in healthcare
- Fair exchange with the involvement of Blockchain

3.1 Fair exchange without Blockchain

3.1.1 Related works which follow the probabilistic approach

Since the fair exchange is a fundamental problem in digital exchange, many researchers have already considered this issue. Among them Damgard et al. [29], Even et al. [46], Goldreich et al. [54], Luo et al. [75] and Markowitch et al. [76] examined the probabilistic approach where the whole message is exchanged chunks-by-chunks. This approach is beneficial in the sense that it does not require any third party; thus, the bottleneck issue can be resolved. The main disadvantage of this approach is that it never guarantees true fairness and has the "unsatisfactory property of uncertain termination" ([12], [113]). Another problem is that this protocol assumes that the two parties have "equivalent computational resources", which is unrealistic in most cases [113].

3.1.2 Related works which follow online/inline TTP based approach

In case of the protocols which are designed to take extensive help from an online/inline TTP, all or some of the messages are exchanged via a third party (Ben-Or et al. [12], Coffey et al. [23], Deng et al. [33], Jianying et al. [68], Kremer et al. [71]). This protocol always ensures fairness throughout the exchange. Of course, the protocol also has drawbacks, namely excessive trust towards a third party and the bottleneck issue. Since every message is transferred via TTP, the process is slow, especially when many users put trust in the same TTP. Another limitation of this protocol is a single point failure issue. If the TTP is compromised by an attacker, the adversary would have the exchanged items instead of the intended recipients.

3.1.3 Related works which follow offline TTP based approach

A better alternative to the above approach could be the use of offline TTP (Asokan et al. [4], Ateniese et al. [6], Feng et al. [48], Maruyama et al. [78], Park et al. [90], Wang et al. [113]). This approach is more practical and can handle the bottleneck issue quite easily. Nevertheless, in some scenarios where an offline fair-exchange policy is used, Bob can take advantage by using Alice's partial confirmation (or commitment) generated at the first step as a bargaining chip (as this step confirms that the encrypted message is indeed coming from Alice which means she is interested in this business deal). To mitigate this issue, Huang et al. [62] works on *Ambiguous Optimistic Fair Exchange*. Based on the idea of fully anonymous group signature and with the help of non-interactive witness indistinguishable (NIWI) proof along with non-interactive zero-knowledge (NIZK) proof, their scheme shows that the partial confirmation will seem ambiguous and no outsider can deduce whether Alice or Bob generated this partial confirmation.

Though offline TTP is widely accepted and indeed an optimistic solution since it can solve the problems that arise by the probabilistic approach and online/inline TTP approach, it still has a significant drawback, putting trust in a third party. A convenient solution to this can be the replacement of the TTP with a "trustless" Blockchain network.

3.2 Use of Blockchain in healthcare data

Seeing its potentiality, the research community has started to realize the utilization of Blockchain beyond the financial applications. This decentralized technology can be immensely useful in developing applications in different domains such as healthcare, logistics, supply chain management, and the Internet of Things (IoTs), among others ([5], [10], [81]).

Kumar et al. [72] discussed the overall utilization of Blockchain in the healthcare industry. They acknowledged the fact that a Blockchain-based solution could significantly help in areas like clinical data sharing, global data sharing, maintaining medical history, research and clinical trials, healthcare data access control, drug supply chain management, and billing/payers of the industry. However, to do so, they also pointed out the key requirements such as nationwide interoperability, data security, data consistency/integrity/immutability, and cost/resources effectiveness, which should be considered before diving into a Blockchain-based solution.

In [101], Shen et al. proposed an efficient way of sharing healthcare data with Blockchain. Their method creates a bridge between immutable small-sized trace records of health data and mutable records of large-sized original data. This way, the patient can grant access to their medical records to a requester, and the requester can retrieve them from another healthcare provider who holds the data for the patient, and each peer-to-peer exchange can be logged into Blockchain as an immutable snippet.

Azaria et al. [7] demonstrates the utilization of Blockchain for accessing medical data. Their decentralized framework (called MedRec) handles EHRs of patients and offers a way to recover their data from various healthcare service providers. The framework, based on the Ethereum platform, offers two distinct incentives for the medical stakeholder to take an interest in the Blockchain network. One is cryptocurrency Ether itself, which is required to execute exchanges in an Ethereum Blockchain. Ether coins should be bought from the cryptocurrency market either by the patients or service providers. The subsequent incentive is the aggregated anonymized medical data, which is essential for the research in the industry. The system performs based on the principle of *smart contracts*, which contains metadata about the ownership of records, data integrity, and permissions.

Mikula et al. [83] proposed a system for identity and access management for electronic health records with Blockchain. Using Hyperledger Fabric's smart contract, their proposed system authenticates the identity of the user before granting access to a database containing medical records.

However, none of the above works cover a specific issue: ensuring fair exchange policy while maintaining the confidentiality of the data.

3.3 Fair exchange with Blockchain

Even though the idea of using Blockchain in ensuring fairness in digital exchange is fairly new, some researchers have already worked on this.

A solution for secure certified electronic mail exchange following the fair exchange protocol is proposed in [60] by Hinarejos et al. In their design, they assume that, though a participant has the decryption key of a POR, it will only be effective when it is published on the Blockchain. However, the drawbacks of that design are they have used the Bitcoin Blockchain as a message board. Bitcoin is a payment system and thus should not be overloaded by using it as a message board. Also, In our case, one of the parties involved in the exchange could be patients who may not possess bitcoin. Eventually, he or she could not use their protocol. Another issue is that bitcoin is a public Blockchain. Thus, if a pending transaction, due to any reason, could not convert to a valid transaction, the receiver of the message would be in an advantageous situation that contradicts the rule of fair exchange. In [79], Meng et al. proposed a fair exchange policy for exchanging physical goods that leverage crypto-currencies. In this design, a common public escrow account is opened on a themis Blockchain from which currency can only be withdrawn if someone has both the secret keys. Both buyer and seller share their encrypted secret keys to their selective mediators of the network using the Shamir secret sharing approach [100] and then provides the ciphertext to the other party (Alice to Bob and vice versa). If no dispute occurs, the buyer will send his original secret key to the seller. The seller can use that to unlock the escrow account. If a dispute arises, the seller will notify the mediators of the network, and if more than half of the mediators work, the seller can recover the private key from them. However, their protocol is particularly designed to ensure fair exchange policy while exchanging cryptocurrency for physical goods, which is not our area of work since we are working on exchanging the data over the digital platform. At the same time, the use of mediators introduce a trusted party and consequently shifted to a central model. The protocol proposed in [49] by Ferrer-Gomila et al. offered true fairness without the involvement of any TTP which is very specific to contract signing. In the solution, the contract will only be valid if it receives approval from the Blockchain network even though the unencrypted contract is shared between the parties without any involvement of Blockchain. It may not be an issue for contract signing protocol, but in our case, we cannot explicitly rely on their design. If a malicious party gets the unencrypted health data outside the chain without providing any partial or full POR in the first step, he/she will terminate the exchange protocol immediately. Also, they designed their scheme, considering that one of the users must have crypto-currency from the Blockchain to support their scheme.

As none of the proposed fair exchange solutions with Blockchain technology is designed for exchanging sensitive large-sized PHRs; we are motivated to fill this crucial gap.

CHAPTER 4

Blockchain

Blockchain, a Distributed Ledger Technology (DLT), is a network for sharing and maintaining the database of static or dynamic data (records/transactions) amongst all the participants in the network. Viriyasitavat et al. [110] defined Blockchain as:

"A technology that enables immutability, and integrity of data in which a record of transactions made in a system are maintained across several distributed nodes that are linked in a peer-to-peer network."

Although most of the current Blockchains are used for financial exchanges, this is not the only usage of Blockchain. In the most generic case, exchanges could be seen just as atomic changes to the system state, and thus Blockchain can be utilized to timestamp archives and secure them from adjustments.

This chapter demonstrates the overall concepts of Blockchain and its functionality.

4.1 Blockchain

Blockchain, as the name suggests, is a combination of multiple blocks containing data. Each block is connected with others through a strong cryptographic technique (a hash value) and finally formed into a chain. Thus, it is called the Blockchain. Figure 4.1.1 and 4.1.2 demonstrate what a block, in Blockchain, contains in general and how the chain is formed in an illustrative manner.

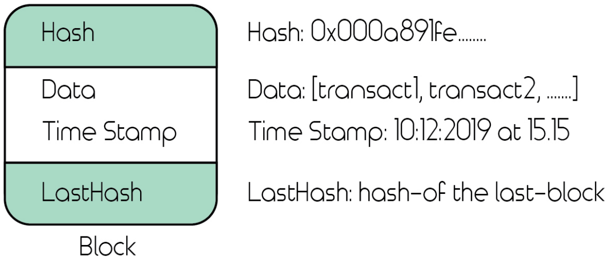

Hash
Data
Time Stamp
LastHash
Block
Hash: 0x000a891fe........
Data: [transact1, transact2, ...]
Time Stamp: 10:12:2019 at 15.15
LastHash: hash-of the last-block

Fig. 4.1.1: What a block, in blockchian, contains in general

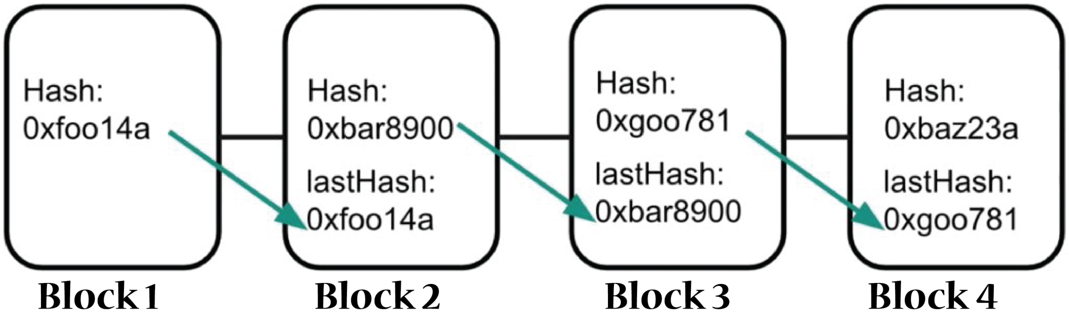


Fig. 4.1.2: How the blocks are chained up

Each data/transaction in the block is signed (following the protocol of digital signature) by the initiator of that data/transaction, which makes sure the data/-transactions are not generated by a malicious person and, as such, protects the block and later on the chain to have false data. The hash of each block is generated by passing the data that block contains and "LastHash" (hash of the previous block) to a hash function. In that way, the hash ensures that, once a block is added to the chain, it is extremely difficult for anyone to alter any of the data it contains. The reason is, changing data would lead to generating a new hash and will not be matched with the "LastHash" of the next block. Consequently, if anyone wants to change the hash of a block, he/she has to change the hash of all the blocks next to it. For changing a hash, for some of the Blockchains demands computationally powerful machines, which is costly.

The Blockchain network is entirely decentralized and designed in such a manner that peers do not have to deal with a middle-man while making any transaction. In other words, no one has to put trust in a third party to deal with his or her data. That is why it is also known as a *trustless* network. In the Blockchain, not only the ledger is distributed, but also, there is no central authority to manage the network like a traditional distributed database system. The power of maintaining the overall network is hand out to all the participants in the network. Each participant holds a copy of a synchronized ledger by himself. As a result, participants can witness their data whenever they want. Since the ledger is consensually maintained and the network is not governed by a single participant, a successful tear-down of the network is highly unlikely. Where in a centralized system, the attacker has to penetrate only one point, in a decentralized distributed ledger system, he/she has to modify the data of more than half of the total nodes of the network to make the system obsolete. Figure 4.1.3 demonstrates how a Blockchain network works with a sample diagram.

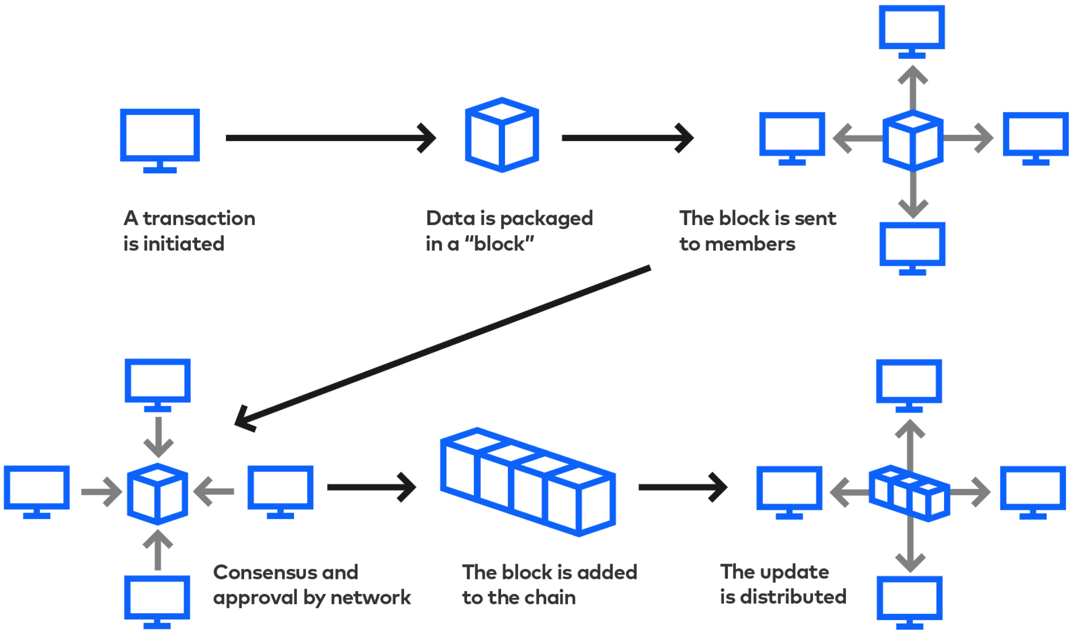


Fig. 4.1.3: How Blockchain works [77]

Blockchain can be categorized based on two points [58]: a) Access to data and b) Access to transaction processing. Based on access to data, a Blockchain can be *public* or *private* and based on access to transaction processing, a Blockchain can be *permissionless* or *permissioned*.

- *Public Blockchain.* No restriction on reading Blockchain data and submitting transactions for the inclusion into the chain.

- *Private Blockchain.* Direct access to Blockchain data and submitting transactions are limited to a predefined party.

- *Permissionless Blockchain.* No restriction on any node for transaction processing.

- *Permissioned Blockchain.* The power of processing a transaction is limited to a predefined party.

4.1.1 Characteristics of Blockchain

A Blockchain system can be characterized based on multiple factors such as Decentralization, Transparent, and Consensus Driven [9].

- *Decentralization.* In a conventional centralized database system, exchanges are trusted or supported through central trusted mediators that assurance legitimacy, which acquires extra cost, and as a consequence the performance turns into a major issue when utilizing central servers [110]. Blockchain is a promising answer for the distributed decentralized transaction management problems [35], being managed among peers in a P2P network.

  A Blockchain system operates without a central hub, and the decision power is distributed amongst the different entities/participants in the network. Thus it can handle a single point of failure issue very well.

- *Transparent.* Blockchain offers both "privacy" and "transparency" in an aggregated way. The privacy of a user is maintained with a powerful cryptographic mechanism (typically the address of the user is kept hidden with a hash function/public key or pseudo address). Thus a user can be anonymized. At the same time, the transaction of an address is open to view by anyone in the network at any time, depending on the type of the system (public/private). This level of transparency is not always supported by a centralized system where central authority in the name of “security purpose” can cut down the rights of the users of accessing their data.

- *Consensus Driven* As there is no central authority, any change in the ledger has to be timely maintained by all the nodes in the network; thus, the role of the consensus mechanism comes. Due to the fact that the ledger has to be synchronized to maintain the data integrity, all the participants must come to an agreement with the inclusion or exclusion of data. As a result, the consensus mechanism is the backbone of any valid transaction to be executed and stored in Blockchain. The idea of consensus mechanism, different types, and how each of the type works is explained in an elaborated fashion in section 4.2.

4.1.2 Smart Contract

A unique feature of Blockchain is *Smart Contract.* It is an automated version of the traditional contract with a set of procedures designed by the Blockchain network to process the input and generates output. The terms and conditions of the contract, agreed by several involved parties, will be written on code, and then the contract will be included in the transaction and submitted to the network. Once added to the chain, the contract will be automatically triggered when the pre-written conditions are met. A smart contract helps to reduce transaction costs by omitting the involvement of third-party in a contract. Some of the Blockchain platforms that support smart contracts are Ethereum and Hyperledger. Figure 4.1.4 illustrates how smart contracts works.

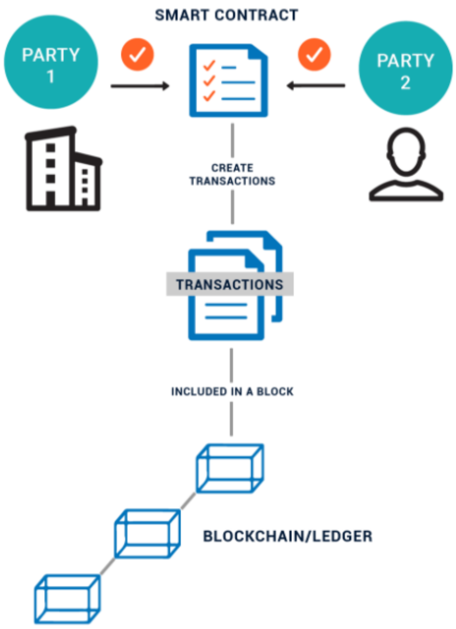


Fig. 4.1.4: How smart contract works

4.2 Consensus Mechanism

The generation of the agreement between all the nodes for any state of the ledger is called a *consensus* mechanism. There are many ways of achieving this agreement, and the mechanism is chosen based on the type and use of the network. However, each consensus mechanism ensures one thing; in every transaction, one node or a group of nodes is acting as a miner/validator and is always responsible for validating that transaction. That node, to prove its worthiness of being a miner/validator, has to bear some fees. These fees could come in the form of requiring huge computational power to find the solution of a problem, lodging some crypto-assets in the network as a stake, having huge computational space to store solutions of a problem, etc., depending on which consensus algorithm is picked up for that network. This entire process confirms, even if someone does not trust any node in the network, he/she can trust the chain. In a broader way, the mechanisms can be classified into four types: a) Work-based mechanisms, b) Stake-based mechanisms, c) Byzantine Fault Tolerance (BFT) based mechanisms, and d) Other mechanisms (which does not fall into any of the three previous types).

4.2.1 Work-based consensus mechanisms

A computationally expensive, yet easy to prove once solved, the puzzle is presented among the nodes. The answer to the puzzle is a hash value, which acts as the hash address for that block. All the nodes in the network will compete with each other to find (aka mine) the solution of that puzzle. The mining is done by a "trial and error" basis. Out of an enormous amount of possibilities, only one value is the answer to the puzzle. Thus every node has to try several times with different possible values and check whether that value is the correct answer or not. This process is called *mining*. The node who finds the answer first (aka miner) gets the right to add the new block to the chain and in return, gets the incentive for solving the puzzle. All the blocks carry the hash value of its own, as well as the hash value of its previous block. As a result, once a block is added to the chain, it is challenging in terms of cost to change the data since all the blocks have to be added again with a new hash address. This is practically unmanageable because of the extremely high expense of solving the puzzle in order to generate the hash. Keeping this concept as a foundation, some of the consensus algorithms which have been forked up from there are: a) Proof of Work [103], b) Proof of Meaningful Work [99], c) Semi-Synchronous Proof of Work [87], d) Delayed Proof of Work [56], and e) Proof of Participation and Fees [50].

4.2.2 Stake-based consensus mechanisms

To confirm a transaction, unlike work-based, in the stake-based mechanism, the node (here mentioned as validator) is chosen deterministically. All the nodes which are going to participate in the competition to get selected as a validator have to lock up some crypto-assets in the network as a stake. The selection of validator is made with some algorithmic way, and it depends on the amount a node deposits as the security payment. For example, if one node deposited ten cryptocurrencies while another deposited a hundred, the chance of being selected as a validator for the second node will be ten times more. This way, unlike Proof of Work or any other work-based mechanisms, one does not have to spend money/cryptocurrency on computationally powerful machines to mine a solution that could go in vain if he/she could not mine first. Some of the examples of consensus mechanisms which are derived from the concept as mentioned above are: a) Proof of Stake [17][57], b) Delegated Proof of Stake [107], c) Proof of Stake Time [96], d) Proof of Stake Velocity [97], and e) Proof of Importance [25].

4.2.3 Byzantine Fault Tolerance (BFT) based consensus mechanisms

Unlike Work-based and Stake-based consensus mechanisms, BFT based mechanisms usually work on private Blockchains. The advantage of using BFT based mechanisms is, it ensures that the system will work even in the presence of malicious/faulty nodes. Moreover, since in private Blockchain, every node knows each other at a certain level, it affirms accountability as well. The idea derives from the concept of Byzantine Generals' problem, a real historical situation. Here all the validator nodes have to give an opinion on the validity of a block. As long as the block gets positive feedback above a threshold value, it is valid. The validators do not have to put any asset as a stake or invest in machines having high computational power, so it is economically friendly as well. The different consensuses which are derived from this primary ground are: a) Practical Byzantine Fault Tolerance protocol [27], b) Delegated Byzantine Fault Tolerance protocol [24], and c) Federated Byzantine Agreement protocol [91].

4.2.4 Other consensus mechanisms

There are some other consensus mechanisms which doesn't follow any of the previous categories, such as:

- *Proof of Burn.*[104] Instead of spending money on computational power to get the selector/validator role, in proof of burn, one has to burn virtual coins. The idea can be similar to buying a mining rig with the virtual coin. The more one burns, the higher the possibility he has to be selected as a leader. The motivation is to encourage the nodes for the short term loss in order to gain the long term achievement. Also, once a node burns the coin, it will ensure a time frame within which he will validate the blocks. However, to prevent early adopters from benefiting too much, every time a node validates a block, a portion of the rig, i.e., the power of the burnt coin decays. So to maintain the hatching power, one has to burn coin periodically.

- *Proof of Elapsed Time.*[95] It is a lottery-like algorithm where a potential participant has to download the "trusted code" and broadcasts the request in order to join the network. The request has to be signed by specialized hardware, which generates a public/private key pair. Once joined in the network, in order to get selected as the "leader" to validate a transaction during each round of consensus, the participant has to wait a certain amount of time defined by the "trusted code." In order to make sure the fairness of the selection (i.e., distribution of the waiting time amongst the participants) specialized hardware is used, which confirms the authenticity of the "trusted code".

- *Proof of Authority.*[28] It uses the concept of PoS; however, instead of keeping stake, which has monetary value, here the participants stake their real identity. The selection of validator is made by rotation in order to reasonably select the validator. Since the real identity of the validator will be flashed, this consensus is usually used for permission Blockchain. It is environmentally friendly as well since there is no need for excessive power consumption in order to mine a block. A similar kind of consensus mechanism, which follows the PoA from a certain point, is *Proof of Reputation* [51].

- *Proof of Activity.*[13] Here both the PoW and PoS protocols are combined in order to provide an extra layer of security while validating the blocks. First, with PoW, the header of the block is found out. However, instead of transaction data, the block contains only the address of the miner. Then a random group of validators is selected following the Proof of Stake protocols who signs in that block and the last person to sign it, fills with transaction data, and then join it with the chain. The incentives are distributed among the miner and validators.

Table 4.2.1 shows the different Blockchain platforms which uses these consensus mechanisms.

Name of the consensus protocol	**Type of Blockchain suitable to use**	**Used in**
Proof of Work	Public & Private	Bitcoin [14]
Proof of Meaningful Work	Public & Private	Vrenelium [111]
Semi-Synchronous Proof of Work	Public & Private	Purple Protocol (under development) [92]
Delayed Proof of Work	Public & Private	Komodo [70]
Proof of Participation and Fees	Public & Private	Not used yet (Proposed on 2018) [50]
Proof of Stake	Public & Private	Ethereum [44]
Delegated Proof of Stake	Public & Private	BitShares [15]
Proof of Stake Time	Public & Private	VeriCoin [109]
Proof of Stake Velocity	Public & Private	Reddcoin [94]
Proof of Importance	Public & Private	NEM [85]
Practical Byzantine Fault Tolerance Protocol	Private	Hyperledger Fabric [63]
Delegated Byzantine Fault Tolerance Protocol	Private	Neo [86]
Federated Byzantine Agreement	Public & Private	BRAVO [19]
Proof of Burn	Public & Private	Slimcoin [104]
Proof of Elapsed Time	Private	Hyperledger Sawtooth [64]
Proof of Authority	Public & Private	Ethereum Kovan [45]
Proof of Reputation	Private	GoChain [53]
Proof of Activity	Public & Private	Decred [32]

Table 4.2.1: Different consensus mechanisms and their use in different Blockchain platforms

CHAPTER 5

Methodology

The electronic adaptation of medical records can be classified into Electronic Health Records (EHRs) and Personal Health Records (PHRs). Healthcare providers use EHRs, which is a collection of patients' health data, and maintained that throughout a specific time by storing accurately in a repository [3]. It helps the care providers (e.g., hospitals, research centers, or clinics) to have improved management of patient health information [20]. Unfortunately, care providers are not very interested in sharing these records among themselves. Even if they agreed to share, the sharing procedure would take much time because of all the regulatory and compliant issues. Also, these records usually are not put into in a similar format in various organizations, thus if exchanged, will hinder the desired goal. As a consequence, the interoperability issue comes into the scene [2]. To address these problems, the PHRs idea was proposed in 2006 [108] and was characterized as an ISO (International Organization for Standardization) standard (ISO/TR 14292) in 2012 [3].

The PHRs are a representation of health records, which, unlike EHRs, is managed by the patient [108]. Patients can decide either to allow the care providers having their PHRs or keep them private. Numerous EHRs for the same patient can coexist; however, only one PHRs of the same patient would exist. The PHRs can incorporate information from multiple sources, extending from gadgets connected to the patient to data stored in different care providers' system [108].

Figure 5.0.1 explains how the PHRs and EHRs, despite the fact, they can be integrated to exchange information for patient's wellbeing, can be different [3].

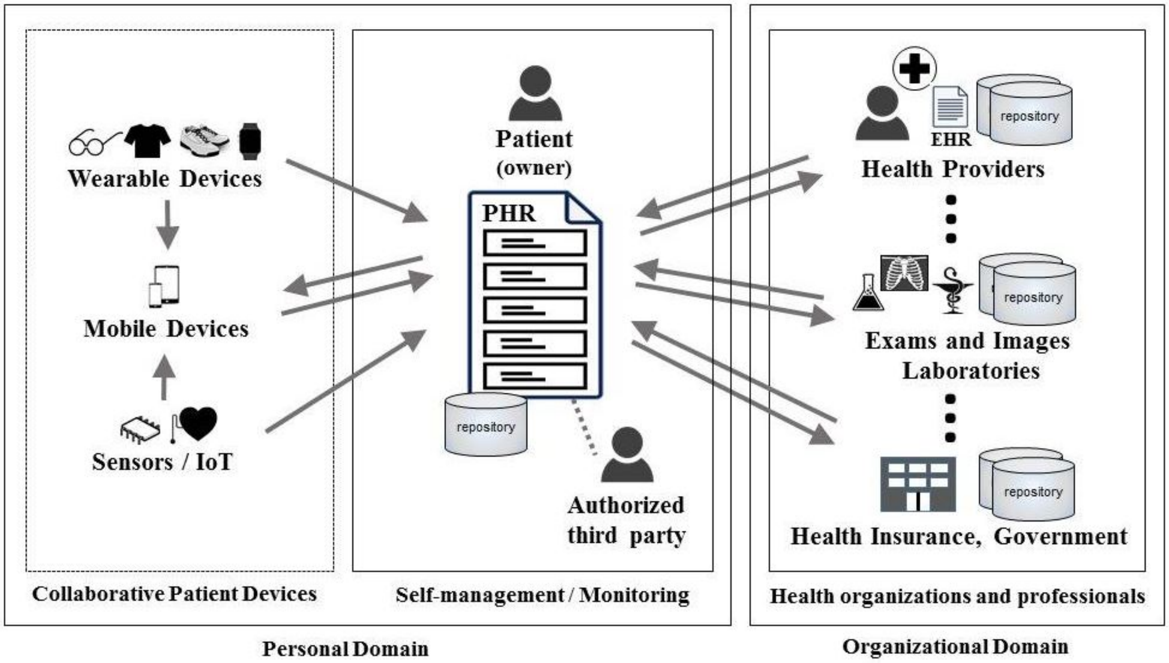


Fig. 5.0.1: Relationship between EHRs and PHRs [3]

Even though patients are the main stakeholders of their PHRs, as per figure 5.0.1, those data can be exchanged with different healthcare providers, research teams, and insurance companies. The exchange can happen in two ways.

1. Patient, himself/herself, share his/her records for his/her benefit.

2. Different care providers (hospitals, research teams, and insurance companies), with the proper consent from the patient, exchanged the data among themselves for the betterment of the patient.

We are focusing on point no. 1. Thus, in our exchanges, we have two main stakeholders in general; patient (sender of the medical records) and healthcare provider(s) (receiver of the medical records). Patients can share their record by hand-to-hand or electronically by using the application which supports the exchange. Nevertheless, any application which will be designed to help these exchanges (for both point 1 and point 2) should possess the privacy and security criteria mentioned in [1]. Out of the eight criteria, in this research, we are targeting to maintain the Non-repudiation criteria by ensuring fair-exchange policy while digitally exchanging the medical records from patients to a healthcare provider. In this chapter, we are going to deduce the problem statement and a detailed methodology to achieve the goal. As Blockchain is an essential tool in this thesis, we will highlight the requirements of the type of network and states the different rules of the network, which are crucial to fulfilling the target. Finally, we will present an informal reasoning of the logic of the proposed methodology based on our targeted criteria.

5.1 Design Goals and Assumptions

Goals:

1. *Fair Exchange:*

   (a) We will electronically transfer a medical record, M, from party A (say, Alice) to party B (say, Bob). After the completion of the exchange, neither Alice nor Bob can deny their role in the exchange, i.e., Alice cannot say she did not send M, and at the same time, party, Bob cannot deny that he did not receive M. Both the sender and receiver of M will have valid proof that will testify against anyone's falsified claims.

   (b) During the exchange, at any step, neither of the parties will be in any favorable situation. By favorable, we mean, if one party starts acting malicious and terminates the exchange protocol after a certain step, he/she will not have any document which can provide him/her any personal gain.

2. *Maintain data confidentiality:* M will remain confidential during the whole exchange procedure. Only the intended persons (Alice or Bob) can know the true meaning of M after the completion of the exchange procedure. Any other participants during or after the exchange cannot find out what M is.

3. *No requirement of any TTP:* During the whole exchange, neither Alice nor Bob has to trust any other party.

4. *Reduce storage overhead:* The medical records will be exchanged in a P2P manner (off-chain) which will significantly reduce the storage overhead for the participants in the Blockchain network.

Assumption: A constant and secure communication channel has been established among all the participants, whether it is a Blockchain network or a P2P communication channel. Table 5.1.1 shows the protocol notation.

Table 5.1.1: Protocol notation

A	Alice
B	Bob
BC	Blockchain
M	Medical Record
$E_K(M)$	Encrypted medical records using encryption key k_1 & k_2
S_X	Digital Signature of X
Y, S_X	Digital Signature of X on item Y
X_{pub}	Public key of X
X_{priv}	Private key of X
RS_X	Reputation score of X
Adr_X	Blockchain address of X
k_1	First symmetric key
k_2	Second symmetric key
H_X	Hash of item X
Z, t	maximum validity time, t, of an item Z
C	Hash of ($E_K(M)$, $B_{pub}(k_1)$, H_{k_2})
$\overline{POO}$	Partial Proof of Origin
POO	Full proof of Origin
$\overline{POR}$	Partial Proof of Receipt
POR	Full Proof of Receipt

5.2 Design rationale

- In our scheme, we are using Blockchain to maintain an immutable timestamp record, which will help to exchange the POO and POR fairly.

  It will help us to achieve goal no. 1(a).

- Here, we are going to use a private Blockchain, which will create a barrier to leaving the network at any time and without informing anyone. The reason is since the recipient in the Blockchain network does not have any leverage, and any malicious recipient can take the key from the waited transactions list and can leave the network before any pending transaction is approved following the consensus protocol, which in turn will keep him/her in an advantageous situation during the overall exchange procedure, we can not rely on a public Blockchain. As a general rule of any private Blockchain, any participant can only leave the network once he/she has fulfilled all the rules defined by the network. Consequently, the use of private Blockchain will help to ensure that neither of the party (Alice or Bob) will be in any favorable situation during the exchange.

  This makes sure to fulfill goal no. 1(b).

- The exchange of items will happen in two phases. The idea is, even though the encrypted message (medical records), 1$^{\text{st}}$ part of the decryption key, partial POO, and partial POR will be exchanged in a P2P fashion (in phase I; we will call if off-chain), without any dependency on TTP, the conversion of partial to full POO or POR will only happen when the exchange of the 2$^{\text{nd}}$ part of the decryption key and partial POO and partial POR are recorded in the Blockchain (in phase II; we will call it on-chain). The reason for splitting the overall exchanges into two phases is explained in subsection 5.2.3.

  Thus, we will be able to accomplish goal no. 2, 3, and 4.

The complete design scheme is presented in subsection 5.2.2.

5.2.1 Private Blockchain requirement

A private Blockchain will limit the Blockchain data to a pre-defied group [58]. Even though there exists some private Blockchain platforms, to justify our Proof of Concept (PoC), we are not going to use any current existing platforms, rather going to create one with policies specially tailored to our necessity. The reasons for doing so are:

- In compared to public Blockchain network, as per their design, private networks are not open. We simply can not get into any private networks and work with our PoC, as it has specific rules and regulations and designed for a specific purpose.

- In existing private Blockchain platforms, the participants in the network has to know each other up to a certain extent (as existing platforms are designed mainly for collaborating different business institutions/entities so that they can do their business in a transparent way). Hence, existing platforms mostly used BFT based consensus protocol, which works based on voting and trust. One of our design goals is to remove the dependency on a “trusted third party”, which does not allow us to go with the current platforms.

- Private Blockchains do not allow one to modify the rules according to users’ necessity, which is a significant hindrance to work with our PoC.

Due to the above reasons, for our PoC, we are are going to use a platform-agnostic approach for implementing the scheme, which will also provide a reference for implementation and testing for other research teams. We are going to create a private Blockchain with the following policies:

Policies for our Private Blockchain:

1. The private Blockchain is going to use Proof of Work (PoW) consensus protocol, a work-based protocol whose detail explanation is as follows:

   Any block in a Blockchain contains its ID, data, and the ID of its previous block. The ID is generated by hashing the combination of the data that the block contains, the previous block's ID, and a specific number (*nonce*). The addition of a nonce will require that the hash value be a unique hash (a hash starts with a certain number of zeros). Nodes have to mine out the nonce. Mining the nonce is computationally expensive since they have to get it through a trial and error basis. The node, who mine the nonce first, will add the new block into the chain, and because of his/her work, in return, will receive the incentives. Since any change in the data of a block will result in generating a completely new hash, an attacker has to mine the nonce of that block and all the blocks previous to that block in the chain. Mining nonce requires costly and computationally powerful machines, and no one would want to waste their powerful machines by going through the old blocks as long as the return on investment (personal gain by changing the data) does not suit them; which is the reason of using PoW in our design. Figure 5.2.1 offers a simple example of how the hash value in PoW is generated.

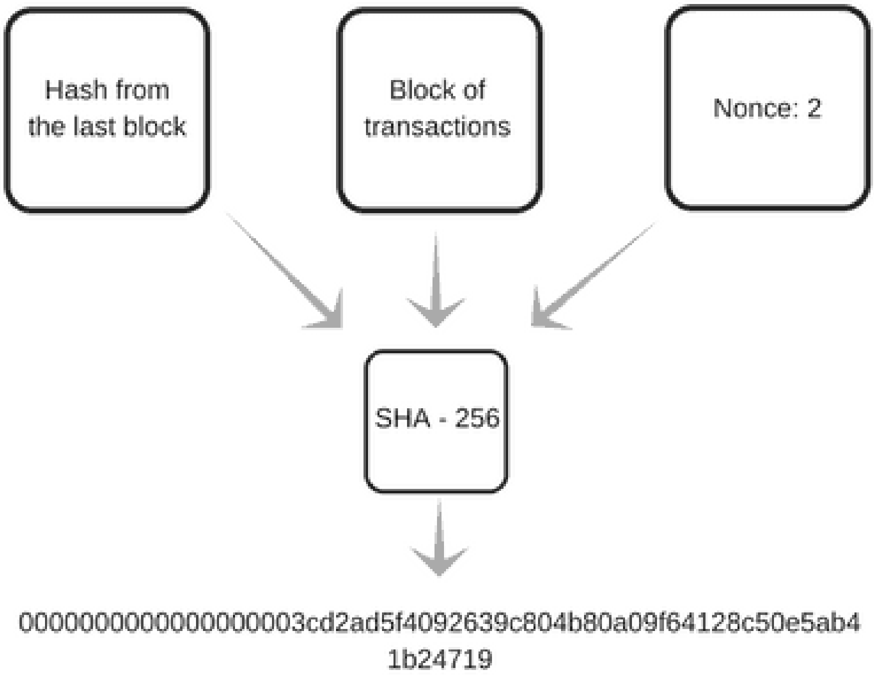


Fig. 5.2.1: Proof of Work: Finding out the unique hash of the current block, a hash starts with a certain number of zeros, by mining the nonce

2. Miners will be rewarded with a reputation score instead of cryptocurrency.

   As the participants in our Blockchain platform are patients and healthcare providers, instead of dealing with expensive cryptocurrencies, we are motivating the miners by providing a reputation score as a reward.

3. Each transaction will cost a specific reputation score (which is significantly less than the mining reward) and will be deducted from the sender's reputation score wallet.

   This will prevent from posting unnecessary transactions into the Blockchain network and will help to reduce the storage overhead.

4. Each node has to maintain a level of reputation score to stay in the network.

   This will ensure the active participation of the nodes in the network. As each transaction will cost a certain reputation score, to fill up those gaps in their wallet, the participants have to mine new blocks. Thus, the overall competitiveness between nodes in the network in finding the nonce value will increase which is crucial for any PoW to work unbiasedly.

5. A node has to post the "leave" request in the network, and once it is mined only then he/she can leave the network. From that point, that node will neither be able to get any new block to his chain nor can see the pending transactions in the pool.

   This will create an immutable snippet in the Blockchain network, and the participant in the network can check whether or not the recipient is in the network before posting any transaction against his/her name.

5.2.2 Design of the Scheme

We are denoting the transaction of data from one party to another by P $\xrightarrow{M}$ Q, which means P is sending certain information, M, to Q. Let us say Alice (patient) and Bob (healthcare provider) agreed to share a medical record. The original medical records will be encrypted with *cascade encryption* technique before sending. Gaži et al. [52] defines cascade encryption as:

> *"A simple and practical construction, used to enlarge the key space of a blockcipher without the need to switch to a new algorithm. Instead of applying the blockcipher only once, it is applied l times with l independently chosen keys."*

With cascade encryption technique, medical records will be encrypted twice with two different symmetric key. This will allow us to exchange the records securely following fair-exchange policy while providing enough information to an adjudicator who can assess the information to decide for one of the parties without any ambiguity should any occurrence of disagreement regarding the exchange of records happen.

The P2P OFE protocol will occur according to the following four steps which are also sketched in Figure 5.2.2.

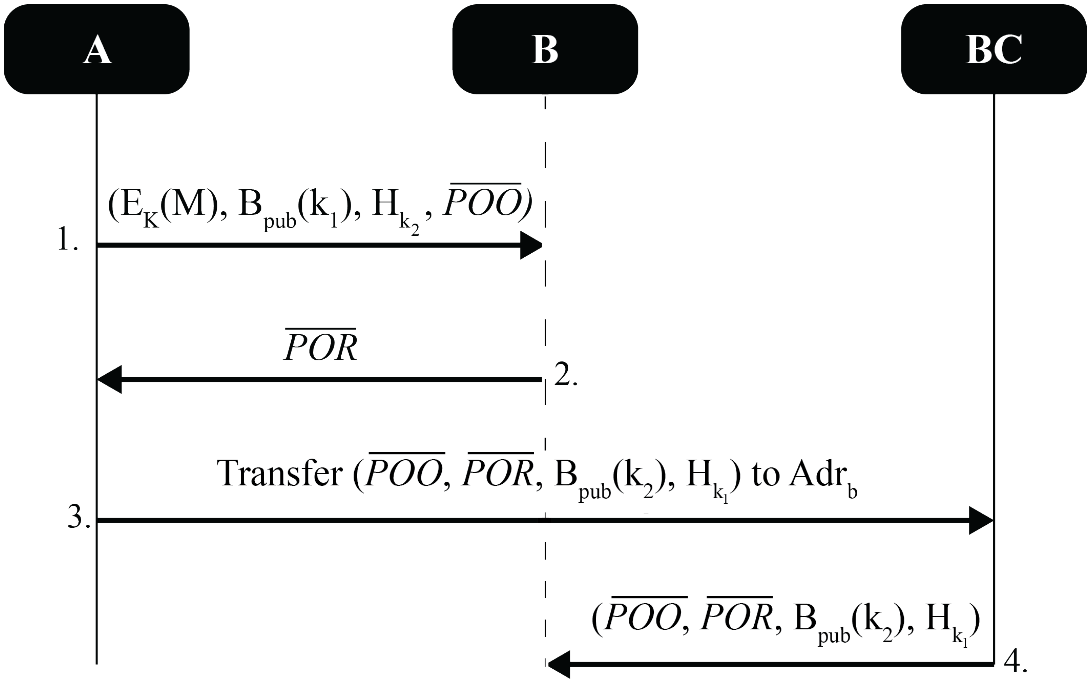

A
B
BC
1.
$(E_K(M), B_{pub}(k_1), H_{k_2}, \overline{POO})$
$\overline{POR}$
2.
3.
Transfer $(\overline{POO}, \overline{POR}, B_{pub}(k_2), H_{k_1})$ to Adr_b
$(\overline{POO}, \overline{POR}, B_{pub}(k_2), H_{k_1})$
4.

Fig. 5.2.2: Proposed P2P OFE scheme with Blockchain

1. $\mathrm{A} \xrightarrow{(\mathrm{E_K(M)},\ \mathrm{B_{pub}(k_1)},\ \mathrm{H_{k_2}},\ \overline{POO})} \mathrm{B}$

   where $\overline{POO} = (\mathrm{C},\ \mathrm{S_A})$ and $\mathrm{C} =$ Hash of $(\mathrm{E_K(M)},\ \mathrm{B_{pub}(k_1)},\ \mathrm{H_{k_2}})$.

   In this step, Alice will encrypt M two times; first with $\mathrm{k_2}$ (let's say the result of encrypting M with $\mathrm{k_2}$ is $\mathrm{M'}$) and then will encrypt $\mathrm{M'}$ with $\mathrm{k_1}$ (let's say the result of encrypting $\mathrm{M'}$ with $\mathrm{k_1}$ is $\mathrm{E_K(M)}$). She will encrypt $\mathrm{k_1}$ with Bob's public key, $\mathrm{B_{pub}}$. $\mathrm{k_2}$ will stay secret known only by Alice. Then, she will generate $\overline{POO}$ which is a combination of the hash of the tuple ($\mathrm{E_K(M)}$, $\mathrm{B_{pub}(k_1)}$ and $\mathrm{H_{k_2}}$) along with her digital signature on that hash.

   Then, Alice will send the (a) encrypted message $\mathrm{E_K(M)}$, (b) symmetric key ($\mathrm{k_1}$) encrypted with Bob's public key, $\mathrm{B_{pub}(k_1)}$, (c) hash of the second symmetric key, ($\mathrm{H_{k_2}}$), and (d) partial proof of origin, $\overline{POO}$ to Bob.

2. $\mathrm{B} \xrightarrow{\overline{POR}} \mathrm{A}$; where, $\overline{POR} = ((\overline{POO},\ \mathrm{t}),\ \mathrm{S_B})$. t starts from now.

   In the second step, Bob computes the partial proof of receipt, $\overline{POR}$, which is a tuple of $\overline{POO}$ and t, signed by his digital signature. Then, sends $\overline{POR}$ to Alice.

3. $\mathrm{A} \xrightarrow{\text{transfer } (\overline{POO},\ \overline{POR},\ \mathrm{B_{pub}(k_2)},\ \mathrm{H_{k_1}}) \text{ to } \mathrm{Adr_b}} \mathrm{BC}$

   In the third step, Alice will send the transaction of transferring the token of their exchanges at the first and second steps along with the second symmetric key encrypted with Bob's public key, $\mathrm{B_{pub}(k_2)}$ and a hash of $\mathrm{k_1}$, $\mathrm{H_{k_1}}$ to Bob's address, in the Blockchain.

4. $\mathrm{BC} \xrightarrow{(\overline{POO},\ \overline{POR},\ \mathrm{B_{pub}(k_2)},\ \mathrm{H_{k_1}})} \mathrm{B}$

   At step 4, from the Blockchain network, a miner, after mining the nonce, will validate and broadcast that transaction to the rest of the nodes by adding a new block containing that transaction on the current chain, and in returns, will be rewarded with a reputation score.

Steps 1 and 2 are the phase I (off-chain) of the overall exchange, where as steps 3 and 4 are phase II (on-chain). As, the full POO or full POR will contain two things;

$\overline{POO}$ or $\overline{POR}$ and the block id, which includes the token of off-chain exchange, after step 4, the $\overline{POO}$ and $\overline{POR}$, exchanged at step 1 and 2, will convert into full POO and full POR.

In Blockchain, all the nodes will be assigned an ID (the public key of that node) which will be counted as the address of the node. The list of available nodes in the network (their IDs) can be found by calling an Application Programming Interface (API). Every node, before posting a transaction, can check whether the recipient is available in the network or not through that list.

5.2.3 Off-chain and On-chain Exchange

One important decision when building a Blockchain solution is to decide what data would be stored on-chain and what will be stored off-chain. It is essential to ensure that only the necessary data will be stored on-chain. Since any data stored on-chain will remain permanently on the network and readable by any member/node in the network. Moreover, Blockchain is not designed to store documents. In particular, storing large documents such as medical images and medical test results is not recommend for two main reasons. First, every node/member in the network is required to keep a permanent copy of the ledger, the ledger size increase over time and never decrease. Therefore, storing massive and large data on the ledger will result in eventually eliminating nodes that could not provide the required storage. Second, the ledger is readable by all the nodes/members of the network. Therefore, there are many privacy and compliance issues if the data is stored on a shared ledger. For that reason, we do not store or exchange any readable medical records on the chain. All the medical records and health-related data are exchange over secure P2P connections off-chain. A detailed record and digital fingerprint for all the exchanged data off-chain is stored on-chain as a tokenized exchange transactions.

The tokenization and digital fingerprint of the health data is a straight forward process. The data is randomly salted and hashed using a secure hashing algorithm. Then the signed hashed data, and the original data are sent to the receiver off-chain over a secure P2P connection. Upon receiving the data, the receiver will validate the data and sign the validation results using the receiver's private-key and send it back to the sender. Finally, the hashed data, receiver's acknowledgment, and the decryption key is signed with the sender's private-key and posted on-chain.

5.3 Informal reasoning of the scheme

Logic in security-sensitive applications demands careful reasoning to find out any loophole in the design. By analyzing different security aspects, one can find out whether or not the tested design/protocol can meet all the claimed objectives, even in the presence of an active adversary. Researchers have pointed out numerous useful ways ([11], [16], [21], [74]) of reasoning different cryptographic protocols to formally analyze the performance in maintaining information security, such as confidentiality, authentication, integrity or non-repudiation, in an insecure network, should the tested protocol is applied. Since the goal of this thesis is to propose a framework which can ensures fairness in digitally exchanging medical records, we are not analyzing the strength or weakness of a particular cryptographic primitive. Instead, we are analyzing the logic of the overall framework based on two of our targeted criteria, Fairness and Confidentiality.

5.3.0.1 Fairness

Our primary objective is to make sure neither the sender nor the recipient at any point during the overall exchange procedure should be in a favorable situation. Here, we are going to analyze the fairness of the proposed scheme during each step of the procedure and find out whether or not the scheme is offering the desired goal.

Step 1:
In this step, A is sending the encrypted medical records, along with the 1st symmetric key and the partial proof of origin, $\overline{POO}$. Thus, at the end of the step, B cannot decrypt the record since he does not have the 2nd symmetric key. He can only have the 2nd symmetric key if he sends the partial proof of receipt, $\overline{POR}$, in the second step.

Result: At the end of the step, true fairness retained since neither of the parties is in an advantageous/disadvantageous situation.

Step 2:

In this step, B will send his partial proof of receipt, $\overline{POR}$ provided that he received, and checked the originality of the different items received in step 1. Even though he has the encrypted medical records now, he still needs the 2nd symmetric key to decipher the records. On the other hand, after receiving the partial proof of receipt, $\overline{POR}$, A has to post their P2P exchanged items as a token along with the 2nd symmetric key in Blockchain, to convert the partial proof of receipt, $\overline{POR}$, into full proof of receipt, POR.

Result: At the end of the step, true fairness retained since neither of the parties is in an advantageous/disadvantageous situation.

Step 3:

In this step, A posted the transaction (token of their exchanges in steps 1 and 2 along with the 2nd symmetric key) in the Blockchain network. In this stage, the posted transaction now should be in the waiting pool, waiting to be mined to have a block of its own, assuming A followed the requirement of posting a transaction correctly. At this point, A still cannot claim to have POR as the transaction is in the waiting pool; thus, she does not have the block ID, which is an essential element to convert a partial proof of receipt, $\overline{POR}$, into full proof of receipt, POR. Likewise, B cannot claim to have full proof of origin, POO, as it also requires the block ID. Even though, he can get the 2nd symmetric key from the transaction in the waiting pool and decrypts the message now, he cannot leave the network as he has to post the leave request in the network. If he posts the request before A posts the items in the network, A will not post the items. Moreover, if he posts after A posts the items in the network, before mining the leave request of B, the items posted by A, will be mined.

Result: At the end of the step, true fairness retained since neither of the parties is in an advantageous/disadvantageous situation.

Step 4:
In this step, the waited transaction, posted by A in step no. 3 has been mined, and the transaction has the block ID. Thus, both A and B can claim to have the full proof of origin, POO, and full proof of receipt, POR.

Result: At the end of the step, true fairness retained since neither of the parties is in an advantageous/disadvantageous situation.

5.3.0.2 Confidentiality

One of the key requirements while exchanging medical records, digitally or not, is to maintain its confidentiality from an unauthorized party. In the proposed scheme, the records are exchanged after performing cascade encryption. To get the original message, an attacker has to have both the decryption keys. Both the symmetric keys, k_1 and k_2, are encrypted with B's public key, B_{pub}, before sending. Thus only B can have both the keys and can decrypt the exchanged records.

Another issue could be the sending of incorrect 2^{nd} symmetric key by A in step no. 3. In that case, B will have both of the symmetric keys and their hashes with A's signature on it, after step 4. B then can submit the keys and hashes, to any regulator, who eventually can discover that A is guilty.

CHAPTER 6

Experiments and Results

Evaluating the results of a scheme will help to give a detailed insight into the behavior of a scheme. However, before gauging, one could use some benchmark, which will help to determine the performance. In this chapter, the details about the experimental setup, its results, and the analysis of those results based on our goal is presented.

6.1 Key points for comparison

According to our design goals (section 5.1), we have following targets to achieve:

1. *Fair Exchange:*

   (a) At the end of the exchange no party, participated in the exchange, can deny their role.

   (b) During the exchange, at any step, under no circumstances, can either of the party be in an advantageous situation.

2. *Maintain data confidentiality:* Only the intended recipient of the message can know the true meaning of Message, M, during or at the end of the exchange.

3. *No requirement of any TTP:* Neither sender nor recipient has to depend on any trusted third party.

4. *Reduce storage overhead:* By exchanging the original records off-chain, the storage overhead for the participants in the Blockchain network will be reduced.

Based on our design goals, before going to have a performance comparison, we are addressing some comparison points and prioritize them on table 6.1.1.

Comparison Points and their priorities:

1. Fair Exchange: Whether the scheme can always ensure true fairness at any point in the exchange.

2. Requirement of trusted third party: Is there any requirement of a trusted third party for the scheme to work.

3. Requirement of Blockchain: Is there any requirement of a Blockchain for the scheme to work. If yes, what type of Blockchain is needed there.

4. Data Confidentiality: Whether the data are exchanged confidentially or not. By confidential, we mean, only the sender and the receiver, or a party trusted by both of them, can know the real meaning of the exchanged data.

5. Data Exchange: Whether the actual data are exchanged on Blockchain or not. If Blockchain is not involved in actual data exchange, we call it off-chain, otherwise on-chain

6. Scalability of the network: In the case of Blockchain-based approach, how large the network was at the time of testing

7. Consensus Mechanism: What consensus is used in the Blockchain

Table 6.1.1: Priority of the key comparison points

Sl no.	**Comparison Point**	**Priority**
1	Fair Exchange	Primary
2	Requirement of TTP	Primary
3	Data Confidentiality	Primary
4	Data Exchange	Primary
5	Requirement of Blockchain	Primary
6	Scalability of the network	Secondary
7	Consensus Mechanism	Secondary

The top five comparison points (marked as a “primary priority”) will decide whether or our design goals (section 5.1) can be fulfilled. On the other hand, the comparison point with “secondary priority” (Sl no. 6-7) will help us to analyze the performance of our design.

Before presenting the result of our work and comparing them with the previous works (works mentioned in section 3.1, 3.2, and 3.3), we are stating the status of those works in tables 6.1.2, 6.1.4, and 6.1.6 according to the comparison points. If a previous work can get “Yes”, “No”, “Maintained”, “Off-chain” and “Yes” on the primary comparison points, *Fair Exchange*, *Requirement of TTP*, *Data Confidentiality*, *Data Exchange* and *Requirement of Blockchain* as an outcome respectively, only then we will consider that, that work may achieve our design goals.

6.1.1 Previous works on fair exchange without Blockchain:

As there has been much work done already on fair exchange without Blockchain, for the convenience while comparing the schemes, here we grouped the papers worked with the same approach and gave them a common name. So, ([29], [46], [54], [75] and [76]) is referring as probabilistic approach. Similarly, ([4], [6], [12], [23], [33], [48], [62],

[68], [71], [78], [90], [113]) is referring as TTP involved approach. Table 6.1.2 shows the status of these works on the comparison points and table 6.1.3 shows whether we can achieve our design goals with these works.

Table 6.1.2: Comparison of different schemes which works on fair exchange without Blockchain

Comparison Point	**Probabilistic Approach**	**TTP Involved Approach**
Fair Exchange	No	Yes
Requirement of TTP	No	Yes
Data Confidentiality	Maintained in some works	Maintained in some works
Data Exchange	Off-chain	Off-chain
Requirement of BC[a] (Type)	No	No
Scalability of the network	Not Applicable	Not Applicable
Consensus Mechanism	Not Applicable	Not Applicable

[a] BC = Blockchain

Table 6.1.3: Fulfillment of the design goals

Design Goal	**Probabilistic Approach**	**TTP Involved Approach**
Goal 1(a)	✗	✓
Goal 1(b)	✗	✓
Goal 2	✗	✗
Goal 3	✓	✗
Goal 4	✓	✓

According to table 6.1.3, both the probabilistic approach and TTP involved approach failed to achieve all of our goals because of not showing the desired output in one or more primary comparison points in table 6.1.2.

6.1.2 Previous works on use of Blockchain in healthcare data:

None of the works mentioned in table 6.1.4 focused on ensuring fairness. As a result, the schema of these works can not help us to achieve all of our goals (table 6.1.5).

Table 6.1.4: Comparison of different schemes which works on use of Blockchain in healthcare data

Comparison Point	**Kumar et al.**	**Shen et al.**	**Azaria et al.**	**Mikula et al.**
Fair Exchange	No	No	No	No
Requirement of TTP	No	No	No	No
Data Confidentiality	Not Applicable	Maintained	Maintained	Not mentioned
Requirement of BC[a] (Type)	Yes (Not Mentioned)	Yes (Not Mentioned)	Yes (Public)	Yes (Private)
Data Exchange	Not Applicable	Off-chain	Off-chain	Off-chain
Scalability of the network	Not Applicable	Tested upto 100 nodes	Tested on Ethereum	Tested on Hyperledger Fabric
Consensus Mechanism	Not Applicable	BFT-SMaRt[b]	Not Mentioned	PBFT[c]

[a] BC = Blockchain; [b] BFT-SMaRt = Byzantine Fault-Tolerant (BFT) State Machine Replication; [c] PBFT = Practical Byzantine Fault Tolerance protocol

Table 6.1.5: Fulfillment of the design goals

Design Goal	**Kumar et al.**	**Shen et al.**	**Azaria et al.**	**Mikula et al.**
Goal 1(a)	✗	✗	✗	✗
Goal 1(b)	✗	✗	✗	✗
Goal 2	Not Applicable	✓	✓	Not Mentioned
Goal 3	✓	✓	✓	✓
Goal 4	Not Applicable	✓	✓	✓

6.1.3 Previous works on fair exchange with Blockchain

Although the works of Meng et al. [79], Hinarejos et al. [60], and Ferrer-Gomila et al. [49] can ensure fairness without any involvement of TTP, their design schema is strictly specific to their targeted goals. For example, Meng et al. [79] work on ensuring fair exchange policy while exchanging physical goods for crypto-currency. Here, not all of the items are exchanged electronically, which is why we put "Not applicable" in the "Data Confidentiality" point (table 6.1.6). On the other hand, even though the protocol of Hinarejos et al. [60] is suitable in ensuring fairness in his use-case, his protocol will not help us to ensure fairness in ours. Hence, although, their work is showing all the expected outcomes on the comparison points, it failed to fulfill all of our goals as in their case they are using a public Blockchain which could help a malicious receiver being in an advantageous situation during the exchange. Finally, Ferrer-Gomila et al. [49] exchanged the items unencrypted, which violates our goal no. 2.

The overall status of their works based on the comparison points is in table 6.1.6 and based on fulfillment of our design goals is in Table 6.1.7.

Table 6.1.6: Comparison of different schemes which works on fair exchange with Blockchain

Comparison Point	**Hinarejos et al.**	**Meng et al.**	**Ferrer-Gomila et al.**
Fair Exchange	Yes	Yes	Yes
Requirement of TTP	No	No	No
Data Confidentiality	Maintained	Not Applicable	Not Maintained
Requirement of BC[a] (Type)	Yes(Public)	Yes(Public)	Yes (Not Mentioned)
Data Exchange	Off-chain	Off-chain	Off-chain
Scalability of the network	Tested with Bitcoin Network	Tested up to 128 nodes	Tested with Bitcoin's test network
Consensus Mechanism	PoW	DPoS[b]	PoW

[a] BC = Blockchain; [b] DPoS = Delegated Proof of Stake

Table 6.1.7: Fulfillment of the design goals

Design Goal	**Hinarejos et al.**	**Meng et al.**	**Ferrer-Gomila et al.**
Goal 1(a)	✓	✓	✓
Goal 1(b)	✗	✓	✓
Goal 2	✓	Not Applicable	✗
Goal 3	✓	✓	✓
Goal 4	✓	✓	✓

As a result, from table 6.1.3, 6.1.5, and 6.1.7, we conclude that neither of the previous works was able to maintain all of our mentioned design goals. Thus, rather than comparing the performance of ours (for both off-chain and on-chain exchange) with any existing fair-exchange protocol, we are analyzing our scheme by observing the behavior of the Blockchain network (whether it fulfills all the mentioned private Blockchain network policies or not) in addition to finding out the fulfillment of the key comparison points in section 6.3. We also calculate the time required to perform the various tasks in off-chain exchanges before exchanging the messages.

6.2 Experimental Setup

6.2.1 Equipment Specifications

We have experimented our design on a workstation with the following specifications (table 6.2.1):

Table 6.2.1: Specifications of the experimental equipment

Specification	**Value**
Processor	Intel(R) Core(TM) i7-5500 U
Frequency	2.40 GHz
RAM	12 GB
Operating System	Windows 8.1 (64-bit)

6.2.2 Network Creation

To implement the proposed scheme, we are using the following technologies: NodeJS, Redis (an in-memory data grid), and Python. Using Redis, we construct a cluster of nodes connected over Pub/Sub architecture and provides the backbone for the Blockchain communication channels. Python is using to write and executing all the logics for off-chain exchange (steps 1 and 2 of the scheme, see subsection 5.2.2). Finally, NodeJS is used to implement the business logic of the Blockchain network.

6.2.3 Dataset Description

In our experiment, we used a synthetic dataset of patient data from EMRBOTS.ORG [43]. The dataset had records of 10000 patients, which were organized into four different tables according to their criteria.

Patient Core Populated Table: It contains a unique patient ID along with personal information of the patient (gender, date of birth, race, marital status, language, and population percentage below poverty).

Admissions Core Populated Table: It contains a unique patient ID, admission ID each time a patient is admitted along with the start and end date of the admission.

Admissions Diagnoses Core Populated Table: It contains a unique patient ID, admission ID, diagnosis code, and diagnosis description each time a patient is admitted.

Labs Core Populated Table: It contains a unique patient ID, lab name (different health information such as White Blood Cell Count, Red Blood Cell Count, Hemoglobin, Hematocrit, Mean Corpuscular Volume, Mch, Mchc, Rdw, Platelet Count, Absolute Neutrophils, Absolute Lymphocytes, Neutrophils, Lymphocytes, Monocytes, Eosinophils, Basophils, Metabolism and Urinalysis) results of the each lab test (in value and unit) and date-time of the lab test.

Out of 10000 patient IDs in Patient Core Populated Table, we randomly selected 1000 IDs, and then based on those 1000 patient IDs (each ID represents different patient), we merged all the four tables and generated 1000 different files. Thus each file now contains a detail description of an individual patient along with multiple medical records (records every time that patient is admitted). In this way, we have created 1000 PHRs. The size of the records are ranging from 2 KB to 1 KB.

6.3 Experimental Results

We are presenting the results in two parts; off-chain and on-chain. In off-chain, we will measure the time taken to perform the required tasks before exchanging the items. In on-chain, we will analyze the behavior of the network by changing the number of participants in the network. We will monitor whether every time our designed network can maintain all the policies mentioned in subsection 5.2.1 along with fulfilling the primary and secondary priorities of key comparison points. Also, we will increase the *difficulty* level of PoW and measure the time required to mine a block for that difficulty. The difficulty level of PoW is determined by how many numbers of zeros there should be at the beginning of a hash. The network will start with a lower difficulty as, in this case, there is only one block (genesis block) with some pre-defined data in that, and will keep increasing in proportion to the number of the blocks in the chain. It will help us to measure the optimal difficulty of PoW for our network (where mining a block is neither too fast nor too slow).

6.3.1 Off-chain Exchange

We have measured the time required for performing each task in step 1 (performing cascade encryption on original medical records, $\mathrm{E_K(M)}$, encrypting the first symmetric key with B's public key, $\mathrm{B_{pub}(k_1)}$, Hashing the second symmetric key, $\mathrm{H_{k_2}}$, and creating the partial proof of origin, $\overline{POO}$) and step 2 (verifying partial proof of origin, $\overline{POO}$, and creation of partial proof of receipt, $\overline{POR}$) of the design scheme (subsection 5.2.2). Table 6.3.1 and 6.3.2 present the results.

Table 6.3.1: Time taken to perform each task of Step 1 of the scheme

Task	**For 1000 PHRs**	**For 1 PHR (avg.)**
$\mathrm{E_K(M)}$	27 sec	.027 sec
$\mathrm{B_{pub}(k_1)}$	8 sec	.008 sec
$\mathrm{H_{k_2}}$	2 sec	.002 sec
$\overline{POO}$	38 sec	.038 sec
Total	**75 sec**	**.075 sec**

Table 6.3.2: Time taken to perform each task of Step 2 of the scheme

Task	For 1000 PHRs	For 1 PHR (avg.)
Verify $\overline{POO}$	5 sec	.005 sec
$\overline{POR}$	15 sec	.015 sec
Total	**20 sec**	**.020 sec**

Here, we have calculated the time taken to perform the tasks for 1000 PHRs and then find the average for 1 PHR by dividing the result with 1000.

6.3.2 On-chain Exchange

We have analyzed our on-chain exchange several times by creating a Blockchain network with a different number of nodes at each time. In each case, our network was able to maintain all the primary priorities and thus fulfill our design goals (see table 6.3.3 and table 6.3.4).

Table 6.3.3: Status of our approach based on the comparison points

Comparison Points	Our approach (P2P OFE)
Fair Exchange	Yes
Requirement of TTP	No
Data Confidentiality	Maintained
Data Exchange	Off-chain
Requirement of Blockchain	Yes (Private)
Scalability of the network	Up to 250 nodes
Consensus Mechanism	PoW

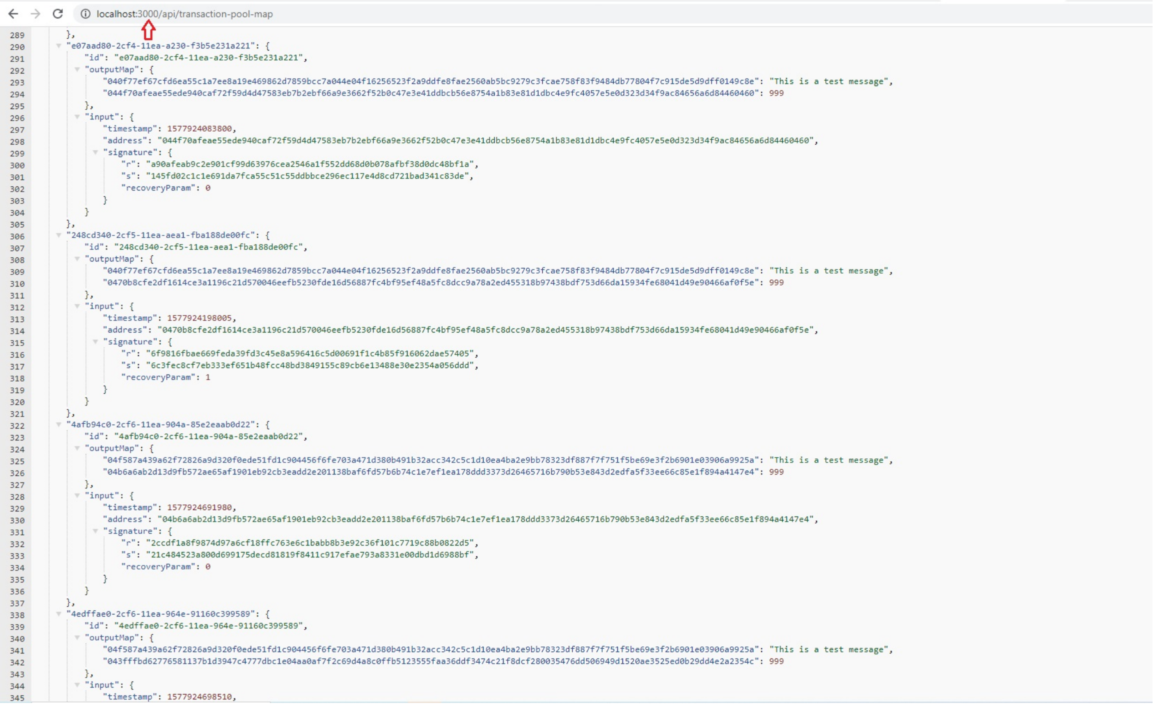
Table 6.3.4: Fulfillment of the design goals

Design Goal	**Our approach (P2P OFE)**
Goal 1(a)	✓
Goal 1(b)	✓
Goal 2	✓
Goal 3	✓
Goal 4	✓

In our Blockchain network, all the nodes have the right to mine, post or view transactions. Once a node posts a transaction in the Pub/Sub channel, the transaction will first enter the “transaction-pool-map” and distributed across the Blockchain network. “Transaction-pool-map” is a list, managed by each node, where every unmined transaction will be queued. A node can call the “transaction-pool-map” API and can check how many transactions are there waited to be mined (see figures 6.3.1 and 6.3.2 where two nodes, running at two different ports, calling the API and getting the same list of unmined transactions). When a node tries to mine, all the transactions in the “transaction-pool-map”, present at the time of mining, will be considered as the “data” of a block and that node will start finding the nonce value for that block (see figure 6.3.3 where a block is formed with 120 transactions in it). We have experimented with the on-chain exchange several times while each time considering some nodes in the Blockchain network as healthcare providers and rests as patients. We have posted numerous transactions by different patient nodes to different healthcare provider nodes (each patient node posts one transaction to one healthcare provider node) and tested the capacity of the “transaction-pool-map”. The waiting time of each transaction to be mined depends on the time required to mine a block. Table 6.3.5 states the result.

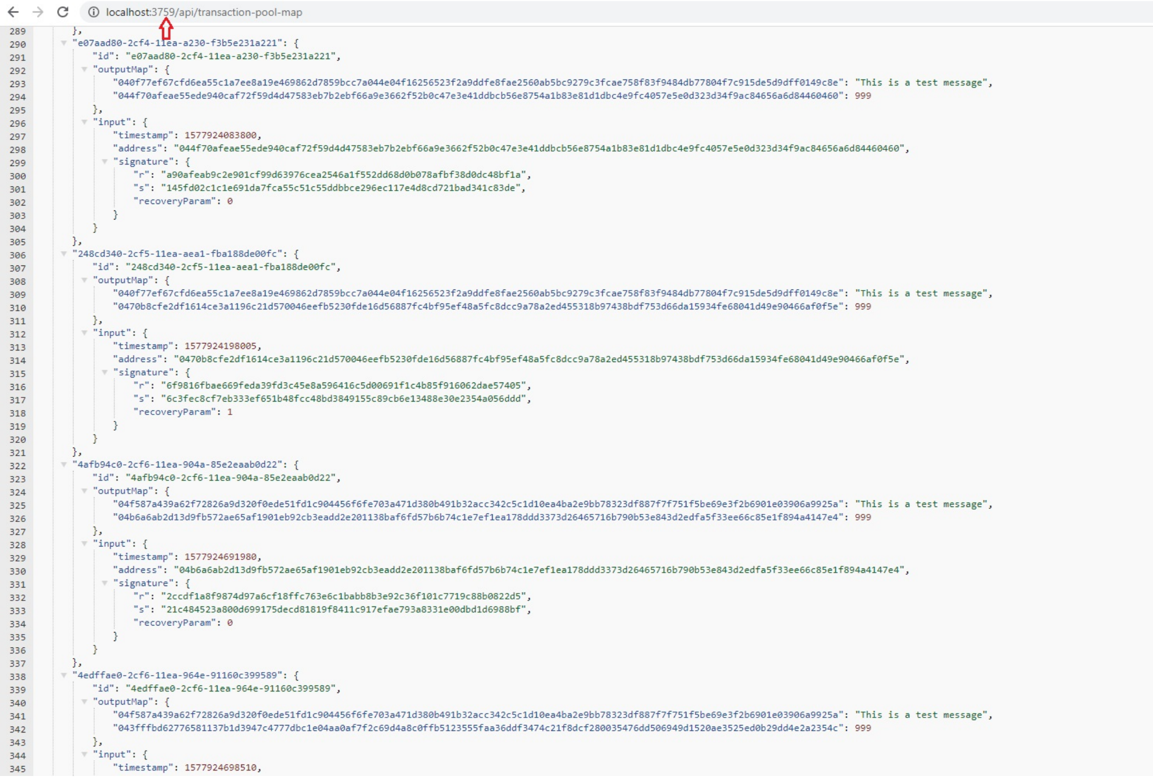
Table 6.3.5: Capacity of "transaction-pool-map"

No. of Healthcare provider nodes	**No. of Patient nodes**	**Total nodes**	**Total no. of transactions in the "transaction-pool-map"**
5	50	55	50
5	75	80	75
5	100	105	100
10	200	210	200


localhost:3000/api/transaction-pool-map

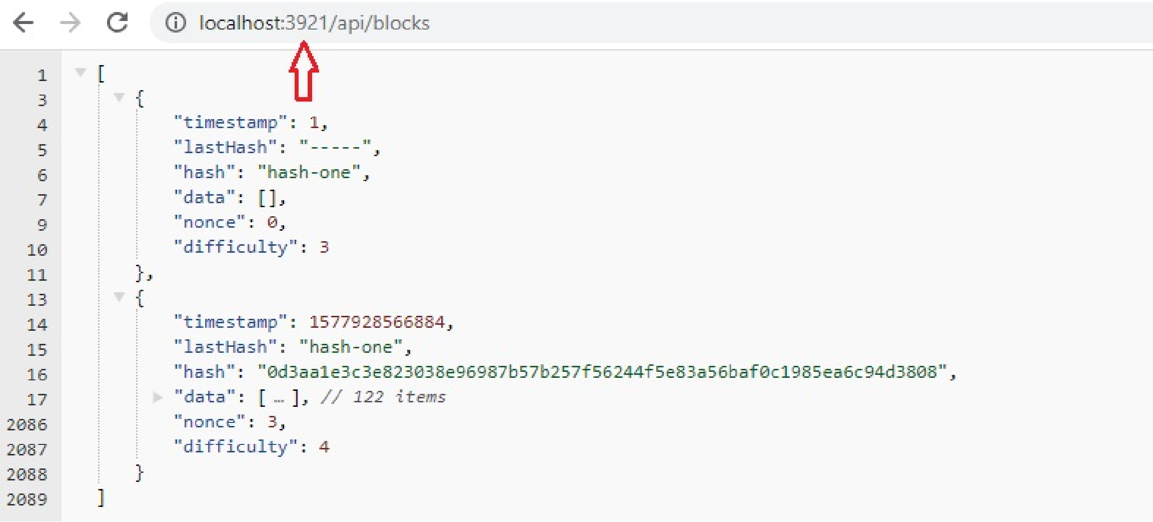
```
},
"e07aad80-2cf4-11ea-a230-f3b5e231a221": {
    "id": "e07aad80-2cf4-11ea-a230-f3b5e231a221",
    "outputMap": {
        "040f77ef67cfd6ea55c1a7ee8a19e469862d7859bcc7a044e04f16256523f2a9ddfe8fae2560ab5bc9279c3fcae758f83f9484db77804f7c915de5d9dff0149c8e": "This is a test message",
        "044f70afeae55ede940caf72f59d4d47583eb7b2ebf66a9e3662f52b0c47e3e41ddbcb56e8754a1b83e81d1dbc4e9fc4057e5e0d323d34f9ac84656a6d84460460": 999
    },
    "input": {
        "timestamp": 1577924083800,
        "address": "044f70afeae55ede940caf72f59d4d47583eb7b2ebf66a9e3662f52b0c47e3e41ddbcb56e8754a1b83e81d1dbc4e9fc4057e5e0d323d34f9ac84656a6d84460460",
        "signature": {
            "r": "a90afeab9c2e901cf99d63976cea2546a1f552dd68d0b078afbf38d0dc48bf1a",
            "s": "145fd02c1c1e691da7fca55c51c55ddbbce296ec117e4d8cd721bad341c83de",
            "recoveryParam": 0
        }
    }
},
"248cd340-2cf5-11ea-aea1-fba188de00fc": {
    "id": "248cd340-2cf5-11ea-aea1-fba188de00fc",
    "outputMap": {
        "040f77ef67cfd6ea55c1a7ee8a19e469862d7859bcc7a044e04f16256523f2a9ddfe8fae2560ab5bc9279c3fcae758f83f9484db77804f7c915de5d9dff0149c8e": "This is a test message",
        "0470b8cfe2df1614ce3a1196c21d570046eefb5230fde16d56887fc4bf95ef48a5fc8dcc9a78a2ed455318b97438bdf753d66da15934fe68041d49e90466af0f5e": 999
    },
    "input": {
        "timestamp": 1577924198005,
        "address": "0470b8cfe2df1614ce3a1196c21d570046eefb5230fde16d56887fc4bf95ef48a5fc8dcc9a78a2ed455318b97438bdf753d66da15934fe68041d49e90466af0f5e",
        "signature": {
            "r": "6f9816fbae669feda39fd3c45e8a596416c5d00691f1c4b85f916062dae57405",
            "s": "6c3fec8cf7eb333ef651b48fcc48bd3849155c89cb6e13488e30e2354a056ddd",
            "recoveryParam": 1
        }
    }
},
"4afb94c0-2cf6-11ea-904a-85e2eaab0d22": {
    "id": "4afb94c0-2cf6-11ea-904a-85e2eaab0d22",
    "outputMap": {
        "04f587a439a62f72826a9d320f0ede51fd1c904456f6fe703a471d380b491b32acc342c5c1d10ea4ba2e9bb78323df887f7f751f5be69e3f2b6901e03906a9925a": "This is a test message",
        "04b6a6ab2d13d9fb572ae65af1901eb92cb3eadd2e201138baf6fd57b6b74c1e7ef1ea178ddd3373d26465716b790b53e843d2edfa5f33ee66c85e1f894a4147e4": 999
    },
    "input": {
        "timestamp": 1577924691980,
        "address": "04b6a6ab2d13d9fb572ae65af1901eb92cb3eadd2e201138baf6fd57b6b74c1e7ef1ea178ddd3373d26465716b790b53e843d2edfa5f33ee66c85e1f894a4147e4",
        "signature": {
            "r": "2ccdf1a8f9874d97a6cf18ffc763e6c1babb8b3e92c36f101c7719c88b0822d5",
            "s": "21c484523a800d699175decd81819f8411c917efae793a8331e00dbd1d6988bf",
            "recoveryParam": 0
        }
    }
},
"4edffae0-2cf6-11ea-964e-91160c399589": {
    "id": "4edffae0-2cf6-11ea-964e-91160c399589",
    "outputMap": {
        "04f587a439a62f72826a9d320f0ede51fd1c904456f6fe703a471d380b491b32acc342c5c1d10ea4ba2e9bb78323df887f7f751f5be69e3f2b6901e03906a9925a": "This is a test message",
        "043fffbd62776581137b1d3947c4777dbc1e04aa0af7f2c69d4a8c0ffb5123555faa36ddf3474c21f8dcf280035476dd506949d1520ae3525ed0b29dd4e2a2354c": 999
    },
    "input": {
        "timestamp": 1577924698510,
```

Fig. 6.3.1: Snapshot of a node, running at port 3000, calling the "transaction-pool-map" API and getting the list of unmined transactions.



```
    },
    "e07aad80-2cf4-11ea-a230-f3b5e231a221": {
        "id": "e07aad80-2cf4-11ea-a230-f3b5e231a221",
        "outputMap": {
            "040f77ef67cfd6ea55c1a7ee8a19e469862d7859bcc7a044e04f16256523f2a9ddfe8fae2560ab5bc9279c3fcae758f83f9484db77804f7c915de5d9dff0149c8e": "This is a test message",
            "044f70afeae55ede940caf72f59d4d47583eb7b2ebf66a9e3662f52b0c47e3e41ddbcb56e8754a1b83e81d1dbc4e9fc4057e5e0d323d34f9ac84656a6d84460460": 999
        },
        "input": {
            "timestamp": 1577924083800,
            "address": "044f70afeae55ede940caf72f59d4d47583eb7b2ebf66a9e3662f52b0c47e3e41ddbcb56e8754a1b83e81d1dbc4e9fc4057e5e0d323d34f9ac84656a6d84460460",
            "signature": {
                "r": "a90afeab9c2e901cf99d63976cea2546a1f552dd68d0b078afbf38d0dc48bf1a",
                "s": "145fd02c1c1e691da7fca55c51c55ddbbce296ec117e4d8cd721bad341c83de",
                "recoveryParam": 0
            }
        }
    },
    "248cd340-2cf5-11ea-aea1-fba188de00fc": {
        "id": "248cd340-2cf5-11ea-aea1-fba188de00fc",
        "outputMap": {
            "040f77ef67cfd6ea55c1a7ee8a19e469862d7859bcc7a044e04f16256523f2a9ddfe8fae2560ab5bc9279c3fcae758f83f9484db77804f7c915de5d9dff0149c8e": "This is a test message",
            "0470b8cfe2df1614ce3a1196c21d570046eefb5230fde16d56887fc4bf95ef48a5fc8dcc9a78a2ed455318b97438bdf753d66da15934fe68041d49e90466af0f5e": 999
        },
        "input": {
            "timestamp": 1577924198005,
            "address": "0470b8cfe2df1614ce3a1196c21d570046eefb5230fde16d56887fc4bf95ef48a5fc8dcc9a78a2ed455318b97438bdf753d66da15934fe68041d49e90466af0f5e",
            "signature": {
                "r": "6f9816fbae669feda39fd3c45e8a596416c5d00691f1c4b85f916062dae57405",
                "s": "6c3fec8cf7eb333ef651b48fcc48bd3849155c89cb6e13488e30e2354a056ddd",
                "recoveryParam": 1
            }
        }
    },
    "4afb94c0-2cf6-11ea-904a-85e2eaab0d22": {
        "id": "4afb94c0-2cf6-11ea-904a-85e2eaab0d22",
        "outputMap": {
            "04f587a439a62f72826a9d320f0ede51fd1c904456f6fe703a471d380b491b32acc342c5c1d10ea4ba2e9bb78323df887f7f751f5be69e3f2b6901e03906a9925a": "This is a test message",
            "04b6a6ab2d13d9fb572ae65af1901eb92cb3eadd2e201138baf6fd57b6b74c1e7ef1ea178ddd3373d26465716b790b53e843d2edfa5f33ee66c85e1f894a4147e4": 999
        },
        "input": {
            "timestamp": 1577924691980,
            "address": "04b6a6ab2d13d9fb572ae65af1901eb92cb3eadd2e201138baf6fd57b6b74c1e7ef1ea178ddd3373d26465716b790b53e843d2edfa5f33ee66c85e1f894a4147e4",
            "signature": {
                "r": "2ccdf1a8f9874d97a6cf18ffc763e6c1babb8b3e92c36f101c7719c88b0822d5",
                "s": "21c484523a800d699175decd81819f8411c917efae793a8331e00dbd1d6988bf",
                "recoveryParam": 0
            }
        }
    },
    "4edffae0-2cf6-11ea-964e-91160c399589": {
        "id": "4edffae0-2cf6-11ea-964e-91160c399589",
        "outputMap": {
            "04f587a439a62f72826a9d320f0ede51fd1c904456f6fe703a471d380b491b32acc342c5c1d10ea4ba2e9bb78323df887f7f751f5be69e3f2b6901e03906a9925a": "This is a test message",
            "043fffbd62776581137b1d3947c4777dbc1e04aa0af7f2c69d4a8c0ffb5123555faa36ddf3474c21f8dcf280035476dd506949d1520ae3525ed0b29dd4e2a2354c": 999
        },
        "input": {
            "timestamp": 1577924698510,
```

Fig. 6.3.2: Snapshot of a node, running at port 3759, calling the "transaction-pool-map" API and getting the list of unmined transactions.

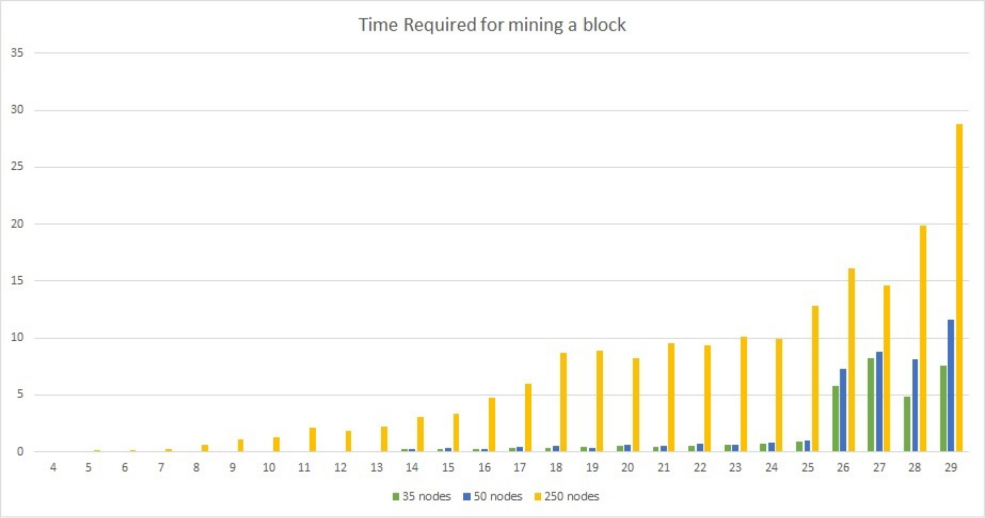


```
[
    {
        "timestamp": 1,
        "lastHash": "-----",
        "hash": "hash-one",
        "data": [],
        "nonce": 0,
        "difficulty": 3
    },
    {
        "timestamp": 1577928566884,
        "lastHash": "hash-one",
        "hash": "0d3aa1e3c3e823038e96987b57b257f56244f5e83a56baf0c1985ea6c94d3808",
        "data": [ … ], // 122 items
        "nonce": 3,
        "difficulty": 4
    }
]
```

Fig. 6.3.3: Snapshot of a node, running at port 3921, mined a block of 120 transactions on it (in the "data" block).

Also, we monitored and found out that our private Blockchain network, in each case, was maintaining all the policies we mentioned in the subsection 5.2.1.

Finally, we measured the time taken to mine a block varying the difficulty level which is showing in the figure 6.3.4.



X axis = Difficulty Level
Y axis = Time (in min)

Fig. 6.3.4: Time requirement for mining a block

The average time to mine a transaction in Bitcoin is approximately 10 minute. If we consider the Bitcoin Blockchain as a standard, in our case, the optimal difficulty level could be from 21 to 28. Unfortunately, in case of a network with 250 nodes, the mining is taking excessive amount of time in compare to others. The reason is, since, in our experiment, we have created the network within the same work station, the time requirement depends on the number of the nodes in the network. If a network is created with different work stations, the time requirement for mining a block would have been different as in that case, every node will have an individual machine to compute and find the nonce.

The complete source code, for both off-chain and on-chain exchange, along with the dataset are available in Appendix section of this report.

CHAPTER 7

Conclusion and Future Work

Had there is no sharing of data, patients have to undergo the same procedure of storing their medical records every time they go to a new healthcare provider. The process is not only tedious but also error-prone, which leads to a loss of potential information. Sharing of personal health records can quickly resolve the case considering the non-repudiation documents are exchanged fairly. With non-repudiation documents in hand, the parties involved in the exchanged can be reluctant knowing that they have strong evidence that will opposed to a misrepresented case. In this thesis, a new approach to achieving true fairness while exchanging personal health records from a patient to a healthcare provider is introduced. The approach is optimistic in the sense that the original health records are exchanged in a P2P manner and thus does not require any third party to interfere. The token of exchange of original records is then posted on the Blockchain network, which ensures that the non-repudiation documents, proof of origin, and proof of receipt, of the record are exchanged fairly. The scheme guarantees non-repudiation and fair exchange by utilizing the immutability property of the shared distributed ledger. To our knowledge, this is the first approach to achieve fairness in digital exchange with the help of Blockchain technology, considering the exchange of large-sized sensitive medical records. The future work of this thesis are discussed below.

7.1 Future Works

We want to extend our works on the following points in the future:

- Extend the scheme to enable exchanging personal health records between healthcare providers on behalf of the patients while allowing the patients to selectively decide how and when they wish to share their data and with whom.

- Test the effect of using other consensus algorithms as a replacement of the default PoW on the scalability of the Blockchain network.

- Investigate the behaviors of the network in case of massive node failures, where a node suddenly disconnects from the network.

REFERENCES

[1] Abbas, A. and Khan, S. U. (2014). A review on the state-of-the-art privacy-preserving approaches in the e-health clouds. *IEEE Journal of Biomedical and Health Informatics*, 18(4):1431–1441.

[2] Alabbasi, S., Ahmed, A., Kaneko, K., Rebeiro-Hagrave, A., and Fukuda, A. (2014). Data types managed database design for dynamic content: A database design for personal health book system. In *TENCON 2014 - 2014 IEEE Region 10 Conference*, pages 1–5.

[3] Alex Roehrs, Cristiano André da Costa, R. d. R. R. and de Oliveira, K. S. F. (2017). Personal health records: A systematic literature review. *Journal of medical Internet research*, 19.

[4] Asokan, N., Shoup, V., and Waidner, M. (2000). Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):593–610.

[5] Aste, T., Tasca, P., and Di Matteo, T. (2017). Blockchain technologies: The foreseeable impact on society and industry. *Computer*, 50(9):18–28.

[6] Ateniese, G. (1999). Efficient verifiable encryption (and fair exchange) of digital signatures. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, CCS '99, pages 138–146, New York, NY, USA. ACM.

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

[8] Barakat, S. (2013). Design and implementation of restful non-repudiation services.

[9] Bauerle, N. (2019). Blockchain 101. https://www.coindesk.com/information/what-is-a-distributed-ledger.

[10] Beck, R. (2018). Beyond bitcoin: The rise of blockchain world. *Computer*, 51(2):54–58.

[11] Bellare, M. and Rogaway, P. (1994). Entity authentication and key distribution. In Stinson, D. R., editor, *Advances in Cryptology — CRYPTO' 93*, pages 232–249, Berlin, Heidelberg. Springer Berlin Heidelberg.

[12] Ben-Or, M., Goldreich, O., Micali, S., and Rivest, R. L. (1990). A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, 36(1):40–46.

[13] Bentov, I., Lee, C., Mizrahi, A., and Rosenfeld, M. (2014). Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]y. *SIGMETRICS Perform. Eval. Rev.*, 42(3):34–37.

[14] Bitcoin (accessed December 19, 2019). Bitcoin. https://www.bitcoin.com/.

[15] BitShares (accessed December 19, 2019). Open-source business development and financial management platform. https://bitshares.org/.

[16] Blanchet, B. (2008). A computationally sound mechanized prover for security protocols. *IEEE Transactions on Dependable and Secure Computing*, 5(4):193–207.

[17] Blog, E. G. S. (2019). What is staking? ethereum's replacement for mining. https://ethgasstation.info/blog/what-is-staking/.

[18] Boyd, C. and Foo, E. (1998). Off-line fair payment protocols using convertible signatures. In Ohta, K. and Pei, D., editors, *Advances in Cryptology — ASIACRYPT'98*, pages 271–285, Berlin, Heidelberg. Springer Berlin Heidelberg.

[19] BRAVO (accessed December 19, 2019). Send and receive secure payments with the power of blockchain. https://bvo.trybravo.com/.

[20] Caligtan, C. A. and Dykes, P. C. (2011). Electronic health records and personal health records. *Seminars in Oncology Nursing*, 27(3):218 – 228. Patient-Centered Technologies: Enhancing Communication and Self-Care for Patients and Caregivers.

[21] Canetti, R. and Krawczyk, H. (2001). Analysis of key-exchange protocols and their use for building secure channels. In Pfitzmann, B., editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 453–474, Berlin, Heidelberg. Springer Berlin Heidelberg.

[22] Cheong, H. J., Shin, N. Y., and Joeng, Y. B. (2009). Improving korean service delivery system in health care: Focusing on national e-health system. In *2009 International Conference on eHealth, Telemedicine, and Social Medicine*, pages 263–268.

[23] Coffey, T. and Saidha, P. (1996). Non-repudiation with mandatory proof of receipt. *SIGCOMM Comput. Commun. Rev.*, 26(1):6–17.

[24] Comben, C. (2019). Delegated byzantine fault tolerance (dbft) explained. https://coinrivet.com/delegated-byzantine-fault-tolerance-dbft-explained/.

[25] Crown, S. . (2019). Proof of importance (poi). https://sci.smithandcrown.com/glossary/proof-of-importance.

[26] CureMD (accessed December 3, 2019). Curemd: Practice without boundaries. https://www.curemd.com/.

[27] Curran, B. (2018a). What is practical byzantine fault tolerance? complete beginner's guide. https://blockonomi.com/practical-byzantine-fault-tolerance/.

[28] Curran, B. (2018b). What is proof of authority consensus? staking your identity on the blockchain. https://blockonomi.com/proof-of-authority/.

[29] Damgård, I. B. (1995). Practical and provably secure release of a secret and exchange of signatures. *Journal of Cryptology*, 8(4):201–222.

[30] Davidson, E. and Heslinga, D. (2007). Bridging the it adoption gap for small physician practices: An action research study on electronic health records. *IS Management*, 24:15–28.

[31] Davis, J. (2019). 32m patient records breached in first half of 2019, 88% caused by hacking. https://healthitsecurity.com/news/32m-patient-records-breached-in-first-half-of-2019-88-caused-by-hacking.

[32] Decred (accessed December 19, 2019). Decred is an autonomous digital currency. https://decred.org/.

[33] Deng, R. H., Gong, L., Lazar, A. A., and Wang, W. (1996). Practical protocols for certified electronic mail. *Journal of Network and Systems Management*, 4(3):279–297.

[34] Deskera (accessed December 3, 2019). Track. reconcile. optimize. with deskera erp software. https://www.deskera.com/erp/.

[35] Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., and Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 1085–1100, New York, NY, USA. ACM.

[36] docengage (accessed December 3, 2019). Software for hospitals. https://www.docengage.in/.

[37] docusign (2003). Understanding digital signatures. https://www.docusign.ca/how-it-works/electronic-signature/digital-signature/digital-signature-faq.

[38] Dong, N., Jonker, H., and Pang, J. (2012). Challenges in ehealth: From enabling to enforcing privacy. In Liu, Z. and Wassyng, A., editors, *Foundations of Health Informatics Engineering and Systems*, pages 195–206, Berlin, Heidelberg. Springer Berlin Heidelberg.

[39] drchrono (accessed December 3, 2019). Practice medicine, not administration. streamline your entire workflow—and get back to what matters most. https://www.drchrono.com/.

[40] drfirst (accessed December 3, 2019). Practical. powerful. innovations. https://www.drfirst.com/.

[41] eClinicalWorks (accessed December 3, 2019). eclinicalworks: Improving healthcare together. https://www.eclinicalworks.com/.

[42] Emily MacIntosh, N. R. and Salah, H. (2014, (accessed December 19, 2019)). Transforming health: Towards decentralized and connected care. https://www.marsdd.com/news/transforming-health-decentralized-connected-care/.

[43] EMRBOTS.ORG (accessed December 3, 2019). Experiment with artificial large medical data-sets without worrying about privacy. http://www.emrbots.org/.

[44] Ethereum (accessed December 19, 2019a). Ethereum is a global, open-source platform for decentralized applications. https://ethereum.org/.

[45] Ethereum (accessed December 19, 2019b). Kovan testnet explorer. https://kovan.etherscan.io/.

[46] Even, S., Goldreich, O., and Lempel, A. (1983). A randomized protocol for signing contracts. In Chaum, D., Rivest, R. L., and Sherman, A. T., editors, *Advances in Cryptology*, pages 205–210, Boston, MA. Springer US.

[47] Fan, L., Buchanan, W., Thummler, C., Lo, O., Khedim, A., Uthmani, O., Lawson, A., and Bell, D. (2011). Dacar platform for ehealth services cloud. In *2011 IEEE 4th International Conference on Cloud Computing*, pages 219–226.

[48] Feng Bao, Deng, R. H., and Wenbo Mao (1998). Efficient and practical fair exchange protocols with off-line ttp. In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No.98CB36186)*, pages 77–85.

[49] Ferrer-Gomila, J.-L., Hinarejos, M. F., and Isern-Deyà, A.-P. (2019). A fair contract signing protocol with blockchain support. *Electronic Commerce Research and Applications*, 36:100869.

[50] Fu, X., Wang, H., Shi, P., and Mi, H. (2018). Popf: A consensus algorithm for jcledger. In *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pages 204–209.

[51] Gai, F., Wang, B., Deng, W., and Peng, W. (2018). Proof of reputation: A reputation-based consensus protocol for peer-to-peer network. In *DASFAA*.

[52] Gaži, P. and Maurer, U. (2009). Cascade encryption revisited. In *Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '09, pages 37–51, Berlin, Heidelberg. Springer-Verlag.

[53] GoChain (accessed December 19, 2019). The blockchain company. https://gochain.io/.

[54] Goldreich, O. and Chaum, D. (1984). *A Simple Protocol for Signing Contracts*, pages 133–136. Springer US, Boston, MA.

[55] Goldschmidt, P. (2005). Hit and mis: Implications of health information technology and medical information systems. *Commun. ACM*, 48:68–74.

[56] Grewal, S. (2018). Komodo's delayed proof of work (dpow) security, explained. https://blog.komodoplatform.com/delayed-proof-of-work-explained-9a74250dbb86.

[57] Group, B. (2015a). Proof of stake versus proof of work. https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf.

[58] Group, B. (2015b). Public versus private blockchains. https://bitfury.com/content/downloads/public-vs-private-pt2-1.pdf.

[59] Health, G. (accessed December 3, 2019). Prime suite: A cloud-based, clinically-driven electronic health record and practice management system that can be customized to align with the unique documenting, billing, and reporting needs of your practice. https://www.greenwayhealth.com/prime-suite.

[60] Hinarejos, M. F., Ferrer-Gomila, J., and Huguet-Rotger, L. (2019). A solution for secure certified electronic mail using blockchain as a secure message board. *IEEE Access*, 7:31330–31341.

[61] Huang, L.-C., Chu, H.-C., Lien, C.-Y., Hsiao, C.-H., and Kao, T. (2009). Privacy preservation and information security protection for patients' portable electronic health records. *Computers in biology and medicine*, 39:743–50.

[62] Huang, Q., Yang, G., Wong, D. S., and Susilo, W. (2015). Ambiguous optimistic fair exchange: Definition and constructions. *Theoretical Computer Science*, 562:177 – 193.

[63] Hyperledger (accessed December 19, 2019a). Distributed ledger software. https://www.hyperledger.org/projects/fabric.

[64] Hyperledger (accessed December 19, 2019b). Distributed ledger software. https://www.hyperledger.org/projects/sawtooth.

[65] Imamoto, K. and Sakurai, K. (2002). A certified e-mail system with receiver's selective usage of delivery authority. In Menezes, A. and Sarkar, P., editors, *Progress in Cryptology — INDOCRYPT 2002*, pages 326–338, Berlin, Heidelberg. Springer Berlin Heidelberg.

[66] Information, S. (accessed December 19, 2019). Symmetric vs. asymmetric encryption – what are differences? https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences.

[67] Insta (accessed December 3, 2019). Get the most comprehensive clinic management system. https://www.practo.com/providers/clinics/insta.

[68] Jianying Zhou and Gollman, D. (1996). A fair non-repudiation protocol. In *Proceedings 1996 IEEE Symposium on Security and Privacy*, pages 55–61.

[69] Kaelber, D., Jha, A., Johnston, D., Middleton, B., and Bates, D. (2008). A research agenda for personal health records (phrs). *Journal of the American Medical Informatics Association : JAMIA*, 15:729–36.

[70] Komodo (accessed December 19, 2019). The industry's only truly decentralized exchange: Atomicdex. https://komodoplatform.com/.

[71] Kremer, S., Markowitch, O., and Zhou, J. (2002). An intensive survey of fair non-repudiation protocols. *Computer Communications*, 25(17):1606 – 1621.

[72] Kumar, T., Ramani, V., Ahmad, I., Braeken, A., Harjula, E., and Ylianttila, M. (2018). Blockchain utilization in healthcare: Key requirements and challenges. In *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–7.

[73] Leaders, E. C. (accessed December 3, 2019). Intuitive solutions that grow your practice and free you to focus on your patients. https://eyecareleaders.com/io-practiceware/.

[74] Lowe, G. (1998). Towards a completeness result for model checking of security protocols. In *Proceedings. 11th IEEE Computer Security Foundations Workshop (Cat. No.98TB100238)*, pages 96–105.

[75] Luo, X., Qin, Z., Geng, J., and Wu, C. (2006). P2pfair: Fair exchange in p2p sharing system without dedicated ttp. In *2006 First International Conference on Communications and Networking in China*, pages 1–5.

[76] Markowitch, O. and Roggeman, Y. (2001). Probabilistic non-repudiation without trusted third party.

[77] Martin, T. (2017). Blockchain, the technology behind bitcoin, is quickly gaining traction as a secure way to record all kinds of transactions. what is it and why should you care? slalom.com.

[78] Maruyama, H., Nakamura, T., and Hsieh, T. (2003). Optimistic fair contract signing for web services. In *Proceedings of the 2003 ACM Workshop on XML Security*, XMLSEC '03, pages 79–85, New York, NY, USA. ACM.

[79] Meng, H., Bian, E., and Tang, C. (2019). Themis: Towards decentralized escrow of cryptocurrencies without trusted third parties. In *2019 Sixth International Conference on Software Defined Systems (SDS)*, pages 266–271.

[80] Metri, P. (2011). Privacy issues and challenges in cloud computing.

[81] Mettler, M. (2016). Blockchain technology in healthcare: The revolution starts here. In *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–3.

[82] Mezher, M. and Ibrahim, A. (2019). Introducing practical sha-1 collisions to the classroom. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, pages 879–884, New York, NY, USA. ACM.

[83] Mikula, T. and Jacobsen, R. H. (2018). Identity and access management with blockchain in electronic healthcare records. In *2018 21st Euromicro Conference on Digital System Design (DSD)*, pages 699–706.

[84] Nakamoto, S. et al. (2008). Bitcoin: A peer-to-peer electronic cash system.

[85] NEM (accessed December 19, 2019). The smart asset blockchain. https://nem.io/.

[86] Neo (accessed December 19, 2019). An open network for the smart economy. https://neo.org/about.

[87] Oncescu, O. (2019). Presenting purple's consensus algorithm — sspow. https://medium.com/purple-protocol/presenting-the-consensus-algorithm-behind-purple-sspow-784ebadb983d.

[88] OPC (2019, (accessed December 19, 2019)). Pipeda if brief. https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/pipeda˙brief/.

[89] Organization, W. H. (2005). Connecting for health : global vision, local insight : report for the world summit on the information society.

[90] Park, J. M., Chong, E. K. P., and Siegel, H. J. (2003). Constructing fair-exchange protocols for e-commerce via distributed computation of rsa signatures. In *Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, PODC '03, pages 172–181, New York, NY, USA. ACM.

[91] Pay, B. (2018). Federated byzantine agreement. https://medium.com/@BRAVOPay/federated-byzantine-agreement-bfa1585a5b41.

[92] Protocol, P. (accessed December 19, 2019). The global decentralized ledger infrastructure. https://purpleprotocol.org/.

[93] Ray, I. and Ray, I. (2002). Fair exchange in e-commerce. *SIGecom Exch.*, 3(2):9–17.

[94] Reddcoin (accessed December 19, 2019). Reddcoin is the first social cryptocurrency. https://www.reddcoin.com/.

[95] Rilee, K. (2018). Understanding hyperledger sawtooth — proof of elapsed time. https://medium.com/kokster/understanding-hyperledger-sawtooth-proof-of-elapsed-time-e0c303577ec1.

[96] rondx, J. (2018). Proof-of-stake-time. https://wiki.vericoin.info/index.php?title=Proof-of-Stake-Time.

[97] rondx, J. (2019). Proof of stake velocity. https://www.reddcoin.com/reddpaper/.

[98] S. Even and Y. Yacobi (1980). Relations among public key signature scheme. Technical Report 175, Computer Science Dept. Israel.

[99] Sascha (2019). Proof of meaningful work (pomw) in the blockchain consensus encyclopedia. https://www.vrenelium.com/blog/.

[100] Shamir, A. (1979). How to share a secret. *Commun. ACM*, 22(11):612–613.

[101] Shen, B., Guo, J., and Yang, Y. (2019). Medchain: Efficient healthcare data sharing via blockchain. *Applied Sciences*, 9(6).

[102] Slamanig, D. and Stingl, C. (2008). Privacy aspects of ehealth. In *2008 Third International Conference on Availability, Reliability and Security*, pages 1226–1233.

[103] Sleiman, M. D., Lauf, A. P., and Yampolskiy, R. (2015). Bitcoin message: Data insertion on a proof-of-work cryptocurrency system. In *2015 International Conference on Cyberworlds (CW)*, pages 332–336.

[104] Slimcoin (2014). Slimcoin a peer-to-peer crypto-currency with proof-of-burn. https://github.com/slimcoin-project/slimcoin-project.github.io/blob/master/whitepaperSLM.pdf.

[105] SoftClinic (accessed December 3, 2019). Clinic and hospital management software. https://www.softclinicsoftware.com/.

[106] Spitzer, J. (2017). 11 of the biggest healthcare cyberattacks of 2017. https://www.beckershospitalreview.com/cybersecurity/11-of-the-biggest-healthcare-cyberattacks-of-2017.html.

[107] Stephens, L. S. (2017). Explain delegated proof of stake like i'm 5. https://hackernoon.com/explain-delegated-proof-of-stake-like-im-5-888b2a74897d.

[108] Tang, P. C., Ash, J. S., Bates, D. W., Overhage, J. M., and Sands, D. Z. (2006). Personal Health Records: Definitions, Benefits, and Strategies for Overcoming Barriers to Adoption. *Journal of the American Medical Informatics Association*, 13(2):121–126.

[109] VeriCoin (accessed December 19, 2019). The only integrated digital currency + digital reserve ecosystem based on the binary-chain protocol. https://vericoin.info/.

[110] Viriyasitavat, W. and Hoonsopon, D. (2019). Blockchain characteristics and consensus in modern business processes. *Journal of Industrial Information Integration*, 13:32 – 39.

[111] Vrenelium (accessed December 19, 2019). Vrenelium: The 3rd generation blockchain ecosystem from switzerland. https://www.vrenelium.com/.

[112] Wan, Z., Deng, R. H., and Lee, D. (2015). Electronic contract signing without using trusted third party. In Qiu, M., Xu, S., Yung, M., and Zhang, H., editors, *Network and System Security*, pages 386–394, Cham. Springer International Publishing.

[113] Wang, G. (2010). An abuse-free fair contract signing protocol based on the rsa signature. *IEEE Transactions on Information Forensics and Security*, 5:158–168.

[114] Wu, R., Ahn, G., and Hu, H. (2012). Secure sharing of electronic health records in clouds. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 711–718.

[115] Yap, H. K. (2018). Unblocking the blockchain: Cryptography and digital signatures. toughnickel.com.

[116] Zhang, R. and Liu, L. (2010). Security models and requirements for healthcare application clouds. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 268–275.

[117] Zhou, J. (2001). *Non-repudiation in Electronic Commerce*. Artech House, Inc., Norwood, MA, USA.

[118] Zhou, J. and Gollmann, D. (1996). Certified electronic mail. In Bertino, E., Kurth, H., Martella, G., and Montolivo, E., editors, *Computer Security — ESORICS 96*, pages 160–171, Berlin, Heidelberg. Springer Berlin Heidelberg.

APPENDIX

LIST OF ABBREVIATIONS

TTP Trusted Third Party

P2P Peer-to-Peer

EHRs Electronic Health Records

PHRs Personal Health Records

PHI Protected Health Information

HIPAA Health Insurance Portability and Accountability Act

PIPEDA Personal Information Protection and Electronic Documents Act

PKE Public Key Encryption

SKE Symmetric Key Encryption

POO Proof of Origin

POR Proof of Receipt

SHA1 Secure Hashing Algorithm Version 1.0

SHA-256 Secure Hashing Algorithm, 256 bits

MD5 Message-Digest algorithm 5

AES Advanced Encryption Standard

DES Data Encryption Standard

RSA Rivest, Shamir, and Adelman

ECC Elliptic-curve cryptography

NIWI Non-Interactive Witness Indistinguishable

NIZK Non-Interactive Zero-Knowledge

IoTs Internet of Things

DLT Distributed Ledger Technology

BFT Byzantine Fault Tolerance

ISO International Organization for Standardization

PoC Proof of Concept

PoW Proof of Work

OFE Optimistic Fair Exchange

API Application Programming Interface

KB Kilobytes

SOURCE CODE

The source code of this thesis is available in the following repository:

`https://github.com/nasim-shourav/P2P-OFE.git`

VITA AUCTORIS

NAME:	Nasim Al Goni
PLACE OF BIRTH:	Dhaka, Bangladesh
YEAR OF BIRTH:	1991
EDUCATION:	Chittagong University of Engineering and Technology, B.Sc in Computer Science, Chittagong, Bangladesh, 2014
	University of Windsor, M.Sc in Computer Science, Windsor, Ontario, 2020