Secure Medical Record Sharing using Blockchain and Elliptic Curve Cryptography

by

Emrul Kais
13101018
Fazle Rabbi
14301097
Sad Murshid Khan Adon
14301101
Mohibul Hassan Chowdhury
13101219

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
August 2019

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

EMRUL KAIS
13101018

FAZLE RABBI
14301097

SAD MURSHID KHAN ADON
14301101

MOHIBUL HASSAN CHOWDHURY
13101219

Approval

The thesis "Secure Medical Record Sharing using Blockchain and Elliptic Curve Cryptography" submitted by

1. Emrul Kais (ID: 13101018)
2. Fazle Rabbi (ID: 14301097)
3. Sad Murshid Khan Adon (ID: 14301101)
4. Mohibul Hassan Chowdhury (ID: 13101219)

Of Summer, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on August 23, 2019.

Examining Committee:

Supervisor:
(Member)

Prof. Mahbub Alam Majumder
Chairman, CSE
Brac University

Program Coordinator:
(Member)

Dr. Jia Uddin
Associate Professor, CSE
CSE
Brac University

Head of Department:
(Chair)

Prof. Mahbub Alam Majumder
Chairman, CSE
Department of Computer Science and Engineering
Brac University

Abstract

Medical records contain sensitive data of a person. In our country, different organizations manage medical records in different ways. Since there is no standard procedure for maintaining medical records, the overall treatment procedure is delayed. In this paper, we are going to introduce a unified medical record blockchain system for maintaining patient records using Ethereum so that the amount of their hassle can be minimized and their treatment procedure can run smoothly. This system will not only be beneficial for our country, but also for other countries that have not adopted a standard for sharing medical records. The principal aim of our proposed system is to ensure authorized access of patient records as well as sharing them securely and cost effectively when needed. In our proposed model, software design patterns are also introduced which will help us achieve scalability, authorization of patient records, and security of data.

Keywords: Ethereum; patient; record; solidity; contract; access control; hash; encryption; decryption; ECC; file; IPFS; EMR; EHR

Acknowledgement

At first, we would like to thank Almighty Allah, to enable us to study in Computer Science and Engineering as well as to submit this thesis paper for the degree of Bachelor of Computer Science and Engineering. Foremost, we wish to express our deepest gratitude to our project supervisor Prof. Mahbub Alam Majumder, Chairperson, Department of Computer Science and Engineering, BRAC University for his guidance, encouragement, helpful directions and commitment in monitoring us throughout this research. We would also like to express our deepest gratitude to Md. Golam Rabiul Alam, PhD Assistant Professor of Computer Science and Engineering Department of BRAC University for his patience and effort in improving our model.

We express our earnest admiration to the Department of Computer Science and Engineering for providing us all the support and giving us the honor to perform the research in partial fulfilment of the requirement for the degree of Bachelor of Computer Science and Engineering.

We are grateful to our family and friends for their immense mental support throughout our research. Lastly, we offer our regards and blessings to all of those who supported us in any respect during the completion of the research.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Trust is the basis of all economic and other transactions. But with the exponentially growing population, it has become impossible to know and trust everyone. So, privacy of sensitive information has become a major issue all over the world. Especially, when it comes to dealing with important information like intellectual properties and healthcare records, trust between the parties involved in exchanging these informations is a must. But as the number of people involved in the exchange of information grows, many times it becomes infeasible to verify everyone. That is when a middleman like bank, government, or someone whom both the parties trust needs to be involved in the process of sharing and verifying information. But then questions of authenticity of the record and security of data uses arises in our mind. Moreover, in this era of internet, people are becoming more interested in financial transactions and exchanging information via internet. So, privacy and security of data is also a major concern in this field. For these reasons, we need some trustless protocol to be notified about the changes of our stored data. A trustless protocol has no central authority and the users themselves are the guardian of their data so there is a less chance of data misuse. That is when blockchain came into the picture. Though the initial use of blockchain was only for money transaction, blockchain has a vast number of applications. With the concept of Decentralized Applications (Dapps) using smart contract, blockchain is gaining more popularity for solving issues related to exchanging information online while preserving the rights of the owners to have the right to know the current condition of their data [15],[26].

1.2 Motivation

Like many other countries, In Bangladesh, traditional and online healthcare record management system lacks interoperability and accountability. Patients leave their data to many hospitals and providers in a scattered way. Sometimes, the patients need to consult multiple doctors for the same or similar type of health problems. But Because of lack of interoperability, they have to start from the beginning of a treatment process. So the overall treatment cost increases. Again, patients don't have control over their data being shared with unauthorized parties. So, security of patient's healthcare records is also a major concern. These problems hampers the overall treatment process. These types of issues motivated us to plan and implement a trustless protocol which will ensure accountability and interoperability between the privacy of healthcare organizations, doctors and providers and also be beneficial for patients to preserve the privacy and security of their healthcare records, Because of smart contract feature and option to integrate with other networks out of blockchain through interfaces, we have found ethereum the best choice to design and implement a system that will facilitate the patients with many benefits.

1.3 Contribution Summary

The purpose of our paper is to design and implement a decentralized medical record system using Ethereum smart contracts and IPFS that will help the patients to control the access of authorities who can view or use their healthcare records. We also implemented a system to store all the files with almost no cost and also increase the security of the files with the help of encryption and decryption with Ed25519 [1]

1.4 Thesis Orientation

The rest of the thesis is organized is as follows:

- Chapter 2, Background information related to our research work.

- Chapter 3, the literature review, discusses the previous works related to electronic medical records (EMRs) such as MedRec, Medchain, Cypherium.

- Chapter 4 discuss our proposed model and methodology. It also shows a summary of the work done.

- Chapter 5 shows the experiment done to prove the proposed model and analyzes the results and the features of the proposed model.

- Chapter 6 of the thesis concludes and summarizes our work, and discuss the scope for future research.

Chapter 2

Background Information

Blockchain: When we say the words “block” and “chain” in this setting, we are really talking around binary data (the “block”) put away in an open database (the “chain”). “Blocks” on the blockchain are made up of digitized pieces of data. Particularly, they have three parts: data of exchanges like the date, the time, and the dollar sum of your most recent buys; data on who is taking part in exchanges; data that separates them from other pieces [5].

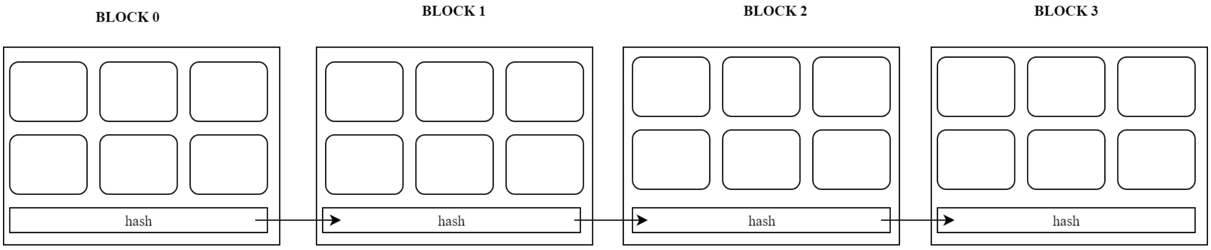


Figure 2.1: A simple diagram of blockchain.

Ethereum test net: When composing programs for the EVM (Ethereum Virtual Machine), for example the Ethereum blockchain, we have to pay for their launch and utilization in gas. This expense can be restrictive in the midst of system abuse and it can likewise be monetarily risky – a bug conveyed on the live system is a bug perpetually open to manhandle. Any change on the Ethereum blockchain is lasting and can’t be fixed. Testnets are duplicates of the Ethereum blockchain practically indistinguishable inside and out to the Mainnet aside from in the way that their Ether is useless (and, obviously, the product that has been sent on these testnets) [2].

TestRPC : Ethereum TestRPC is a quick and customizable blockchain emulator. It permits creating calls to the blockchain while not the overheads of running associate actual Ethereum node. In TestRPC , accounts is recycled, reset and instantiated with a set quantity of Ether (faucets or mining not necessary) also gas value and mining speed is changed. An outstanding example of TestRPC is ganache [3].

Ethereum wallet: In Ethereum just like others in the cryptocurrency world a wallet represents the storage of the currency however in contrast to the physical world wallet; in cryptocurrency a wallet holds the private key of the public-private key pair that enables the wallet holder to access the cryptocurrency [4].

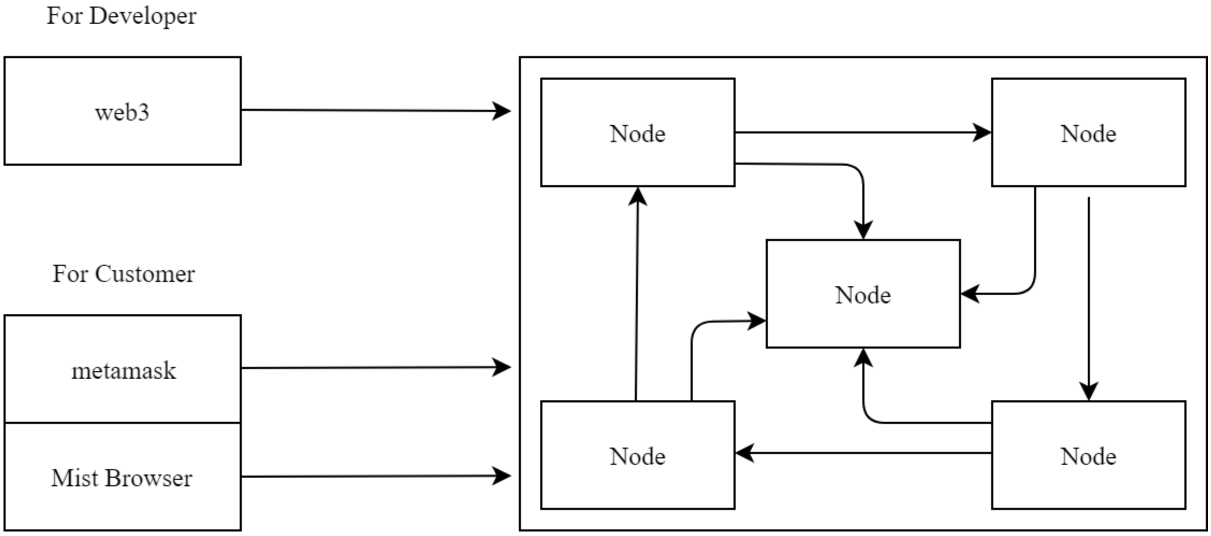


Figure 2.2: A block diagram of how Ethereum works

Cryptocurrency transaction: In cryptocurrency or any decentralized app that usages blockchain dose dealings from one account to a different or perform an action. This dealings may be of 2 types: one that moves funds from one account to a different or will deploy or interacts with a sensible contact like add, update or retrieve data from that smart contact [6].

Wei: The Ethereum cryptocurrency network has a unit named Wei which the smallest unit of ether. One(1) Ether = 1,000,000,000,000,000,000 Wei (1018).

nonce	Amount of transaction done by a user.
to	Destination account address
value	Number of 'Wei' to be sent to destination address
gas prize	Amount of Wei the sender is willing to pay per unit gas to get his transaction processed
startGas/ gasLimit	Units of gas that this transaction can consumed
v r s	Cryptographic pieces of information that can be utilized to produce the sender's account address. Produced from the sender's account address

Table 2.1: Overview of a transaction.

DApps: DApps or Decentralized Apps are software systems running on Peer-2-Peer decentralized computer networks and using a blockchain as a cryptographically stored ledger, a scarce asset model system compared to a traditional centralized database system. [7].

Smart contract: Smart contracts are self-executing contracts with the particulars of the understanding among purchaser and vendor being straightforwardly composed into lines of code. The code and the understandings contained in that exist over an appropriated, decentralized blockchain network. Smart contracts grant trusted exchanges and agreements to be done among unique, unknown gatherings without the requirement for a focal authority, lawful framework, or outer authorization system. They render exchanges discernible, straightforward, and irreversible. [8].

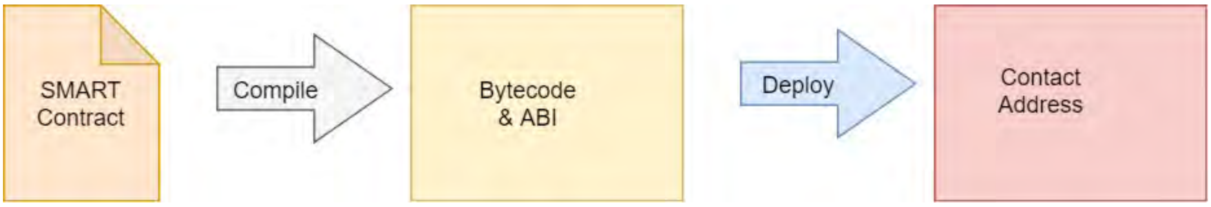


Figure 2.3: Flow diagram for compile and deployment of Smart Contract.

Ethereum: Ethereum was a blockchain protocol that had a programming language that permits for applications, referred to as contracts, to run in the blockchain network. At first described in a white paper by its creator, Vitalik Buterin in late 2013, Ethereum was created as a platform for developing decentralized applications that may do quite build easy coin transfers [9]. But later it becomes an open software platform which is based on blockchain technology. For development purposes, developers can build and deploy different kinds of decentralized software or applications. Using local test network or main network of public nodes, Ethereum now can be used as a decentralized virtual machine.

Ethereum testnet vs mainnet pros and cons: The testnet is used to test and deploy smart contract locally. The main advantages of testnets are : they are completely free of cost and it is easy to install the necessary dependencies and run the blockchain applications locally. Moreover, since there are a few participants for local testnet, the whole deployment procedure was very fast. On the other hand, Though it costs real currency to deploy the blockchain based app in mainnet, to get the benefits of decentralized trustless network, mainnet is the only option .

	Mainnet	Testnet
Purpose	Functional blockchain	Testing environment
Transaction	Real transaction stored on actual blockchain	Fake transaction
Coin	Possess real value	No monetary value
Transaction cost	Paid using native coins	Very low cost
Transaction frequency	High	Low

Table 2.2: Comparison of Ethereum Mainnet and Testnet [10].

Solidity(programming Language): Solidity object-oriented, high-level, application-oriented programming language that is used to create a smart contract. Smart contracts are applications that administer the conduct of accounts among the Ethereum state [11].

Hashing: Hashing is generating a value from a string of text using a mathematical function. Hashing is one way to enable security during the process of message transmission when the message is intended for a particular recipient only. It is a one way cryptographic function that cannot be decrypted to original text. The hash function produces a value that is of fixed size. (i.e. in SHA512, the hash value is a 512 bit long string). A formula generates the hash, which helps to protect the security of the transmission against tampering. Hashing is also a method of sorting key values in a database table in an efficient manner [12].

Public and private key cryptography: Public key cryptography (PKC) is an encryption technique that uses a paired public and private key (or asymmetric key) algorithm for secure data communication. A message sender uses a recipient's public key to encrypt a message. To decrypt the sender's message, only the recipient's private key may be used. PKC is also known as public key encryption, asymmetric encryption, asymmetric cryptography, asymmetric cipher, asymmetric key encryption and Diffie-Hellman encryption [13].

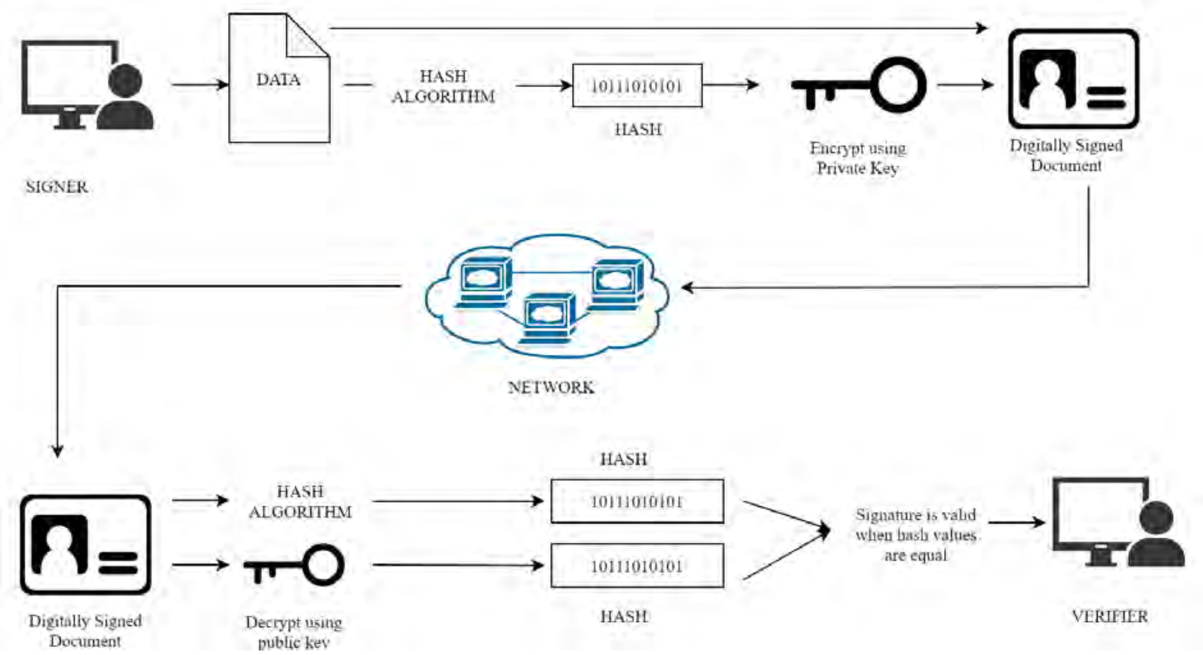


Figure 2.4: Digital signature generation and verification with public key and private key.

Elliptic Curve Cryptography: Elliptic Curve Cryptography (ECC) is a technique to encrypt and decrypt data using the algebraic structure of elliptic curve over a finite field. ECC needs much smaller keys (usually 128or 256bit keys) to encrypt and decrypt data. Where, Elliptic Curve is a curve that is defined over a field of k, the form $y^2 = x^3 + ax + b$, where a and b are co-efficients belongs to K such that, $4a^3 + 27b^2 \neq 0$, which determines what the shape of the curve will be[14].

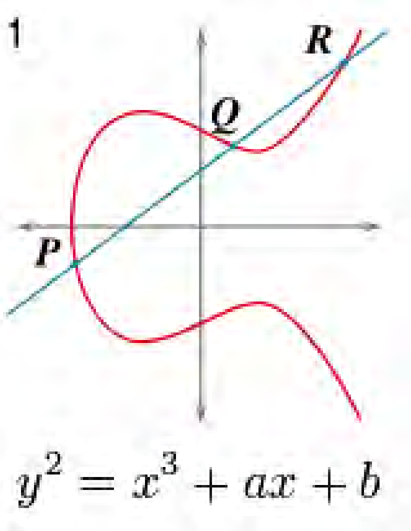


Figure 2.5: Example of Elliptic Curve.

Edward curve: Edward curves are a special type of elliptic curve, more precisely hyper elliptic curve which is of the form $a^2 + b^2 = c^2\left(1 + da^2 2b^2\right)$. Where $cd\left(1 - dc^4\right) \neq 0$. For example, Curve25519 is a hyper-elliptic curve [15].

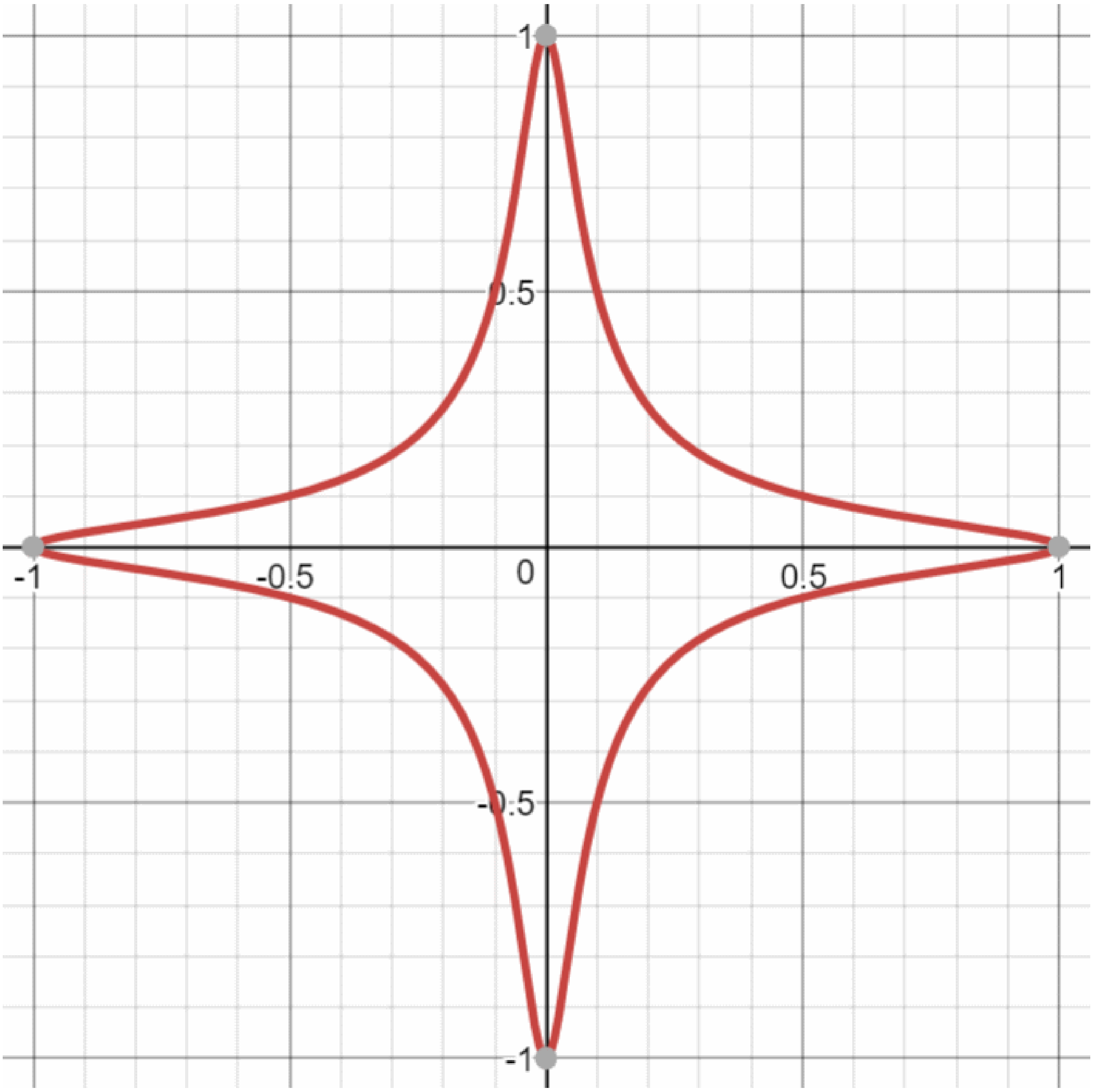


Figure 2.6: Edwards curves [40]

Ed25519 is a particular example of the signature scheme family of EdDSA. Ed25519 is defined in RFC8032 and is commonly used[1]. EdDSA's only other example that anyone cares about is Ed448, which is slower, not commonly used, and also indicated in RFC 8032. Keys and signatures in one example of EdDSA are not relevant in another example of EdDSA: Ed25519 and Ed448 are distinct signatures EdDSA has following parameters:

- an integer power p congruent to 1 (mod 4) , p is prime

- an integer value c, where $c \geq 10$

- a cryptography based hash function H that produces output of $2c$ bit

- encoding of elements of length $(c - 1)$ bits under a field F_p, where F_p is a finite

- an element d of the field F_p, where d is non-square

- a prime number l which is in the range of $2c-1$ and $2c-3$ which satisfies the following constraint:
- a component $B \neq (0,1)$ of the set:

$$E = \{(a,b) \in F_q \times F_q : -a^2 + b^2 = 1 + da^2b^2\}$$

The condition that d is certainly not a square suggests that $d \in \{0, -1\}$, so this a set E makes a group with impartial component $0 = (0,1)$ under the twisted Edwards addition law

$$((a_1, b_1) + (a_2, b_2) = \left(\frac{a_1b_2 + a_2b_1}{1 + da_1a_2b_1b_2}, \frac{b_1b_2 + a_1a_2}{1 - da_1a_2b_1b_2} \right)$$

introduced by Bernstein, Birkner, Joye, Lange, and Peters in [17].
EdDSA key generation : Ed25519 use private keys of size 32 bytes, public keys of size 32 bytes, signature of 64 bytes and provides high security level (128 bit).

The elliptic curve for the EdDSA algorithm starts with a generator point G and a subgroup of order q for the elliptic curve points which are generated from G. r
The EdDSA Key-pair consists of :
Private Key (Integer) : $\mathrm{K_{private}}$
Public Key (Elliptic Curve Point) : $\mathrm{K_{public}} = \mathrm{K_{private}} * \mathrm{G}$
The private key , $\mathrm{K_{private}}$ is generated from a random integer called *seed* which has the same bit length as the curve order.

IPFS: IPFS is a distributed system for storing and accessing files, websites, applications, and data [18].

GPG: As defined by [16] the OpenPGP standard has a free and complete implementation done by the GNU Foundation known as GPG [19]. By using GnuPG, user's digital data and communications can be cryptographically encrypted and signed; it features a all-around key management system. GunPG also comes with different modules to access different kind of public key lists. GnuPG, also known as GPG, comes with the ability to integrate with a variety of tools and is a very intuitive command line tool. An abundance of frontend applications and libraries are handy that can be used with GPG. GnuPG can also be used with S/MIME and Secure Shell (SSH) [19],[20].

Metamask: MetaMask is a browser extension that acts as a bridge between internet browsers, Ethereum, and decentralized applications built on the Ethereum network [21].

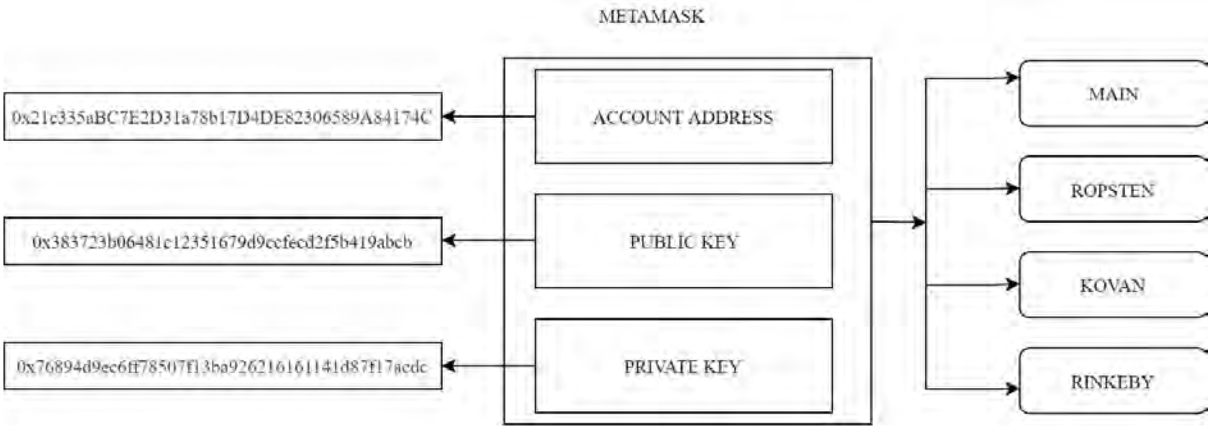

METAMASK
0x21c335aBC7E2D31a78b17D4DE82306589A84174C
ACCOUNT ADDRESS
0x383723b06481c12351679d9ccfecd2f5b419abcb
PUBLIC KEY
0x76894d9ec6ff78507f13ba926216161141d87f17acdc
PRIVATE KEY
MAIN
ROPSTEN
KOVAN
RINKEBY

Figure 2.7: Metamask interfacing with different Ethereum network

Bio-sample: Samples of body tissue for diagnostic purposes to determine the cause of disease.

Chapter 3

Literature Review

Over the years people have tried to come up with solutions to store the electronic medical record in a secure and privacy centric manner. Some have suggested vertical data partition and search across plain text and cryptext (Yang, Li Niu 2015) [22].Furthemore, EMR(Electronic Medical Record) are very private data which is owned by the patient and without their consent this data cant be used anywhere, even if we want to use for research purposes. EMR data is needed to be shared amongst healthcare providers and other doctors along with insurance and drug providers who will provide insurance and proper medication for the patient. Within the US, they have specific law that gives control over a citizens health data, and they might set laws and controls on who can see and get his health information[23][24]. If someone wants to share his clinical information for academic purposes, or exchange them from one clinic to another, he may be required to sign a legal contract that indicates what sort of information will be shared, who will receive the data, and how long the recipient will be able to access the data. This may be greatly troublesome to facilitate, particularly when someone is moving to another city, locale, or country and may not know the healthcare facility he/she needs to go [25][26]. With the invention of technology like blockchain and IPFS we can observe a trend in the shift towards the cryptography based, decentralized approach for saving electronic medical record. We tried to study as much as paper we found related to electronic medical record storage and retrieval.

Blockchain technology has developed as a reliable and transparent mechanism to store and circulate data that is capable of addressing various privacy and security-related issues in healthcare system (Khezr, Moniruzzaman, Yassine, amp; Benlamri, 2019). As blockchain technology uses consensus mechanisms and cryptography to verify transactions in untrustworthy environments, the need for third-party intermediaries is eliminated in blockchain technology-based healthcare systems (Moniruzzaman et al. 2019) [27].

Timothy Nugent, David Upton and Mihai Cimpoesu (2016) highlight that since Satoshi Nakamoto's paper on bitcoins, blockchains just entered a new era with the release of 2.0 with the introduction of smart contracts, which can address various data manipulation issues that appear in clinical settings. In smart contracts, the program and information used are located at a distinct address in a blockchain and their execution is cryptographically approved by the user network (Nugent et al.

2016), which allows for secure interactions among the parties. Rather than only endorsing balances and exchange of digital tokens, they show how smart contracts permit the state about an arbitrary information and logic “to be agreed on by the network using the same cryptographic principles” (Nugent et al. 2016). In other words, as confidentiality is one of the significant issues in healthcare, smart contracts allow all parties involved interact with each other in decentralized environments[28].

Laure A. Linn Martha B. Koo, M.D. describes an access control manager for health records based on blockchain to promote the interoperability difficulties of the industry articulated in the Office of the National Coordinator for Health Information Technology (ONC) Shared National Interoperability Roadmap According to their proposal, a public blockchain network is used to manage access control of the medical records kept outside of the blockchain network (Laure Koo, 2016). The research published on using a public blockchain to manage and control access to personal data done by the Massachusetts Institute of Technology’s is referenced in the paper.Laure A. Linn Martha B. Koo, M.D. describes system that can facilitate and manage who has permission for healthcare document based on blockchain to promote the interoperability difficulties of the industry articulated in the Office of the National Coordinator for Health Information Technology (ONC) Shared National Interoperability Roadmap According to their proposal, a public blockchain network is used to manage access control of the medical records kept outside of the blockchain network (Laure Koo, 2016). The research published on using a public blockchain to manage and control access to personal data done by the Massachusetts Institute of Technology’s is referenced in the paper.[29].

In their paper, they also mentioned that the information used in our proposed health blockchain system would be acting like an index, inventorying the healthcare document and information of the entire user population. The index is like a library for medical info with a catalog. The catalog has metadata about the medical records and says where can we get the data from. The medical blockchain would work like this as well. Exchange in the blocks of blockchain would must have a user’s unique id, a cryptographic link to the medical record and a timestamp for when the exchange was done in the system. To significantly improve data access and efficiency, the exchange would contain the what type of information contained in the medical record and any other metadata that would help regularly used queries (the metadata can be integrated into the system just like tags). The medical blockchain must have full indexed history of all the healthcare information, with past healthcare information and other health data that can help to get a good information structure for the patient from handheld and wearable devices, and a user can make a lifelong use of it [29].

All medical information would be stored off blockchain in a data repository called a data lake. Data lakes are scalable and can persist a wide range and type of data, from pictures to document files to key value stores. Data lakes would be a very helpful tool for health research and would be used for a variety of analysis including mining for factors and variables that impact outcomes, choosing optimal treatment options based on genetic markers and features and resolving elements that influence preventative medicine for the patients. Data lakes support robust queries, text mining, text analytics and ML. All data stored in the data lake would be encrypted and digitally signed to make sure of the privacy and authenticity of the information. Actualizing the proposed medical blockchain framework portrayed in this paper has the imminent to draw in a large number of people, medical service providers, health care entities and health researchers will be able to share large amounts of medical data with proven and supported security and privacy protection of the patients. The attainment of,storage and sharing of this information would lay a scientific foundation for the progress of medical research and test whether or not mobile devices can engage individuals in their health care for good health. [29].

Furthermore 'MedRec: Using Blockchain for Medical Data Access and Permission Management' is considered to be the pionear on the field of decentralizing electronic medical record and using cryptography. In their paper they proposed 3 different contract to achieve the intended goal. The Register contract is used for the identity information to a Ethereum address and it also allows the creation of new identity and the change of existing one. The Patient-Provider Relationship Contract is used to communicate between two nodes where one node used for data upload for other. It also defines the data points and access control for those data. The Summary Contract is used to keep track of history. The system stores the patient info in the blockchain but the report files are stored in a SQL database. The best feature of this paper is that it shows the way to decentralization of the electronic medical record by introducing smart contact and blockchain but the weak point is saving the data to the SQL database which even encrypted and distributed needs a central system to work properly [30].

As a key feature of Medchain whitepaper v1.0 to healthcare, is that MedChain patient decentralized app can be downloaded from app store for free and can also be used as a web app on the MedChain website so that service provides gets encouraged to use the service.[31].

Ethereum is the "implementation of a permissionless programmable blockchain that allows any user to create and execute the code of arbitrary algorithmic complexity on the Ethereum platform," which is also known as the Ethereum Virtual Machine (EVM) (Dubovitskaya, Xu, Schumacher, amp; Wang, 2018). Two types of Accounts can be created on EVM: Externally owned account (EOA) and contract account (Dubovitskaya et al. 2018). Externally owned account (EOA) is controlled by a user's private key whereas contract account can be seen as "an autonomous agent that lives in the Ethereum execution environment and is controlled by its contract code: smart contract" (Dubovitskaya et al. 2018). The primary use of smart contract is the encoding of "arbitrary state transition functions," which allows users to create systems with various functionalities by "transforming the logic of the system into the code" (Dubovitskaya et al. 2018) [32].

In their paper in section 4.2.3 they have discussed about how to implement the Proxy pattern to introduce interoperability through the blockchain system. Keeping patients medical information from getting directly encoded in the blockchain. They also have mentioned about remote proxies that can be helpful in an Ethereum based healthcare application as they give us a local representation for an object that is in a different address space and server to create a wrapper function for the contract object that has relevant data if it is used as a storage system. A defensive proxy can also be introduced to control access of the original sensitive object. For illustration, to intercept unauthorized users on the blockchain to change the patient data object, it can be set up as a proxy to do some permission checking prior to forwarding the change request [33]. Cypherium Uses IPFS for data storage and ed25519 for address generation [34].

"Cypherium: A Scalable and Permissionless Smart Contract Platform" presents a model called Cypherium, which is based on blockchain network that is capable of maintaining both Bitcoin's decentralizing aspect and its processing of innumerable transactions-per-second. This "hybrid consensus mechanism" or "dynamic node election mechanism" of Cypherium is aimed at ensuring the replacement of all nodes after a certain period, which can significantly reduce "the possibility of single point of failure." These features of Cypherium make it a reliable mechanism to store high-value data. One crucial property of Cypherium is that it can be utilized as a registry of hashes that can direct to files on external storages such as IPFS, which enhances its applicability in secure data storage.[34]

Furthermore, to securely storing the file and accessing it with proper authorization many system uses IPFS(InterPlanetary File System). Juan Banet mentioned about IPFS in his white paper draft. In his paper he said that IPFS can be expressed as a single BitTorrent swarm which exchange objects inside a single Git repository. Using IPFS we can get a high through-put content-addressed block storage model, with content-addressed hyperlinks. It will give a simplified Merkle DAG. It's a data structure which can be used to build a versioned file systems, blockchains and even a Permanent Web. IPFS combines a distributed hash table, an incentive block exchange, and a self-certifying namespace. As it's a distributed file system that's why a single node getting error or problem does not cause the whole file system to have an error. IPFS is trustless that means the nodes that are in the file system doesn't need to trust each other[35],[36].

Chapter 4

Proposed Model

In our proposed model we have 3 types of user: “patients”, “doctors” and “providers”. The patients are general users with personal data, medical records and information; the doctors are verified experts on subject matter; and the Providers are organizations, companies or health institutes that provide health care services. The user creates an account on the platform and puts their basic information, such as name, date of birth, height, weight, contact and address. The users also go through a set of questionnaires to determine their smoking habits, allergies, previous medical procedure(s), and family history of illness, if any. A public key of the user which is generated using GPG is also kept as a record, as it will be used to encrypt the data. The doctors are registered with their validation of expertise, ie. certificates, name, place of visit, visiting hours, and a public key. The providers will register with their valid license and the list of services the provide with their price. All the info is saved in the Ethereum blockchain.

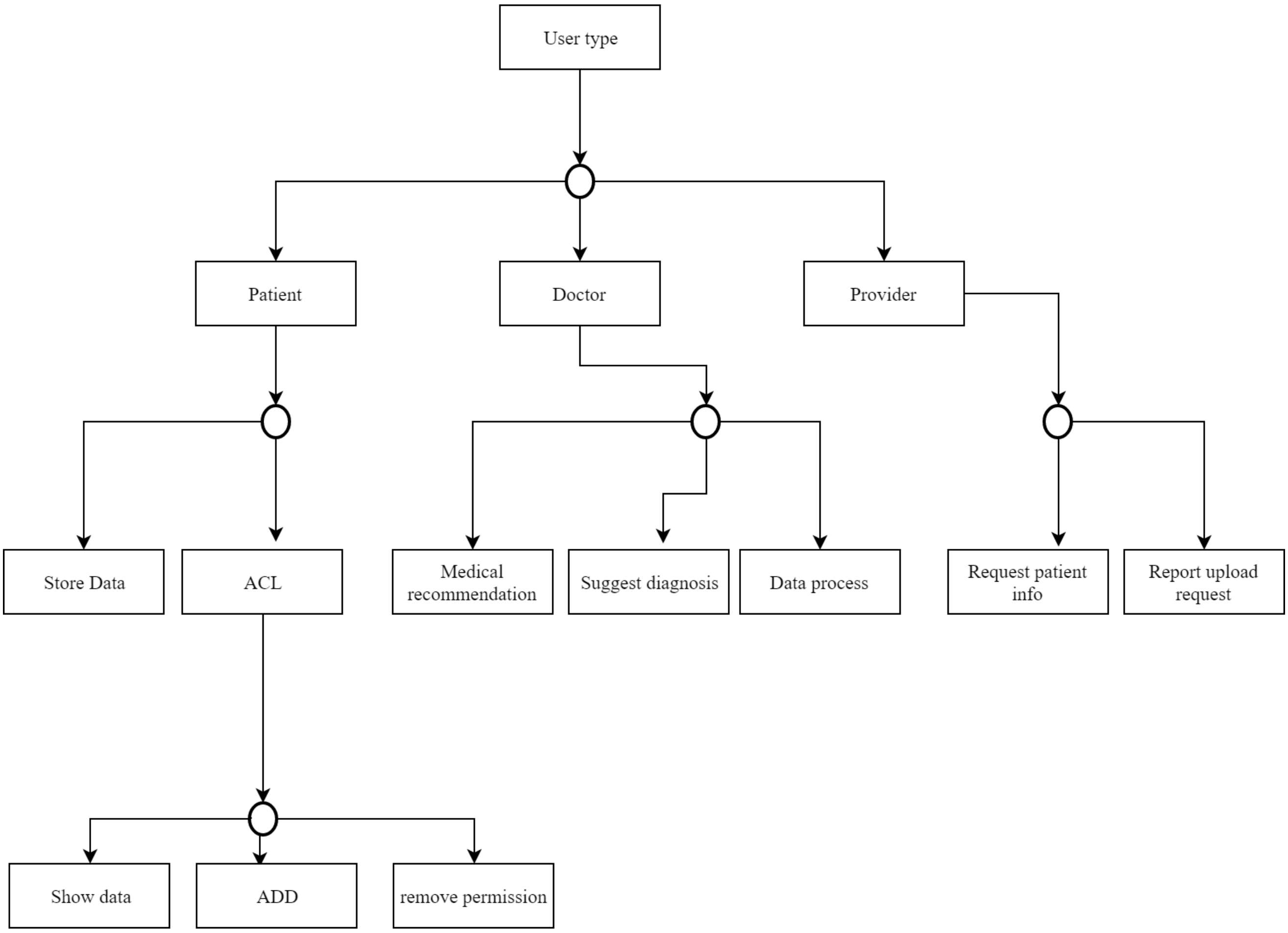


Figure 4.1: Overview of user roles in the proposed system

The patients, if needed, can see a doctor and allow the doctor to see his/her medical information and reports only (ie. name, age, height, weight, smoking habits, allergies, previous medical procedure and family history of illness), but not the date of birth, contact or home address info. After the doctor does his primary diagnosis and observation, a more specific diagnosis or test may require so the doctor would recommend that test to the patient. The patient then can choose a provider for that test.

If a patient chooses a provider, his/her name, age, contact info and address becomes visible to the provider. After payment to the provider is done, the patient will then provide the bio-sample for pathology if needed or be present at the lab to do the test. Once the test is done, the provider will upload the report to the IPFS system using the patient's public key so only the patient can see the report file for now. Additionally, a notification is generated on the patient's end saying that the report has been uploaded, and when the patient confirms the notification, the hash generated after uploading the encrypted report to the IPFS is added to his/her record. Now, the patient can both allow or deny access to the doctor. Allowing the access would mean decrypting the report using the user's public key and re-encrypting it with the doctor's public key. The user can put a time limit where he/she can set for how long the doctor will be able to see the report. In all circumstances, no party will be allowed to download or store the report.

4.1 Storing Data

We have designed an Ethereum blockchain-based system to store the reference of the patient's record. Patient record can include multiple types of files, such as patient's prescription as document, images of test reports, and some others formats. The actual record of a patient will be stored in the InterPlanetary File System (IPFS). Every record is identified by a unique hash by which one can access the record. This hash will be stored as the reference of the patient's record in the solidity contract of the Ethereum blockchain system. Here, we used the concept of permissioned blockchain. This is achieved by storing the patient record data out of Ethereum blockchain network and enforcing an access control policy to give permission to specific organization and authority to access patient's record.

4.2 Access Control

We are going to introduce a permission protocol to give permission for the authorized access of patient's data by using the concept of owner and inheritance in solidity smart contract. The solidity contract InitialContract is implemented as an interface where only the patient will be the owner of the record. In that interface, an access control list is provided which will contain the addresses of the providers and organization account which are allowed to access patient records. User can also modify the access of other accounts (i.e allow or block any account to access the records) [37].

4.3 Data Security

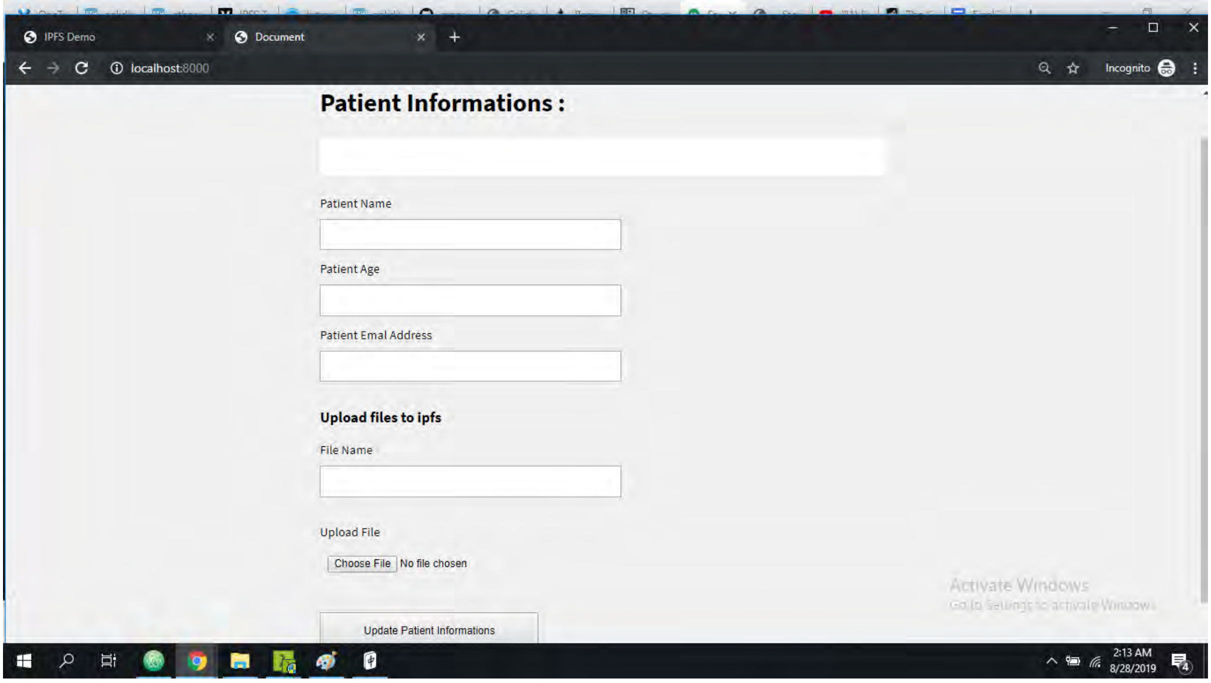
Since the files will be stored in IPFS, anyone can access the files if they somehow manage to get or generate the hash. But we do not want everyone to access our files since those files contain personal and sensitive information about an individual. Considering the security, the files will be encrypted using the user's public key. Only the authorized parties who are included in the access control list in [3.2] can access the files and decrypt it with the private key.

Chapter 5

Experimental Setup and analysis

5.1 Introduction to Experimental Setup

To test out our proposed model we created a prototype one page site. The proof of concept site has two input fields to put the name and the age of the patient and a file uploader using which a user can upload report files to the system. After the initial deployment of the Smart Contract the users can put the data on the input fields and select the file to be uploaded; the input fields data is put on a block. A public key of the user which is generated using a PGP software in this case GPG is also uploaded to the block. The file to be uploaded is first encrypted with the public key of the user and then upload to the IPFS system and the hash of that file is saved to the block containing users data. When the user asked for the data it can be retrieved using the same system.



Patient Informations :

Patient Name

Patient Age

Patient Emal Address

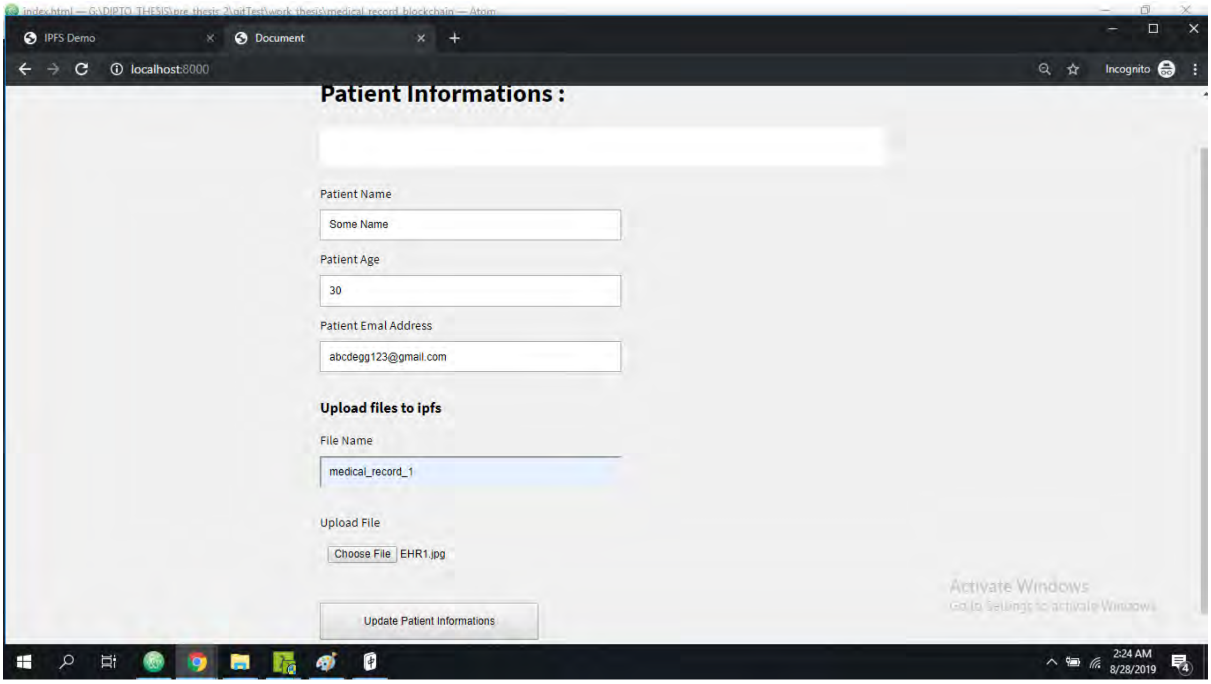
Upload files to ipfs

File Name

Upload File

No file chosen

Figure 5.1: User Interface - In the beginning



Patient Informations :

Patient Name

Some Name

Patient Age

30

Patient Emal Address

abcdegg123@gmail.com

Upload files to ipfs

File Name

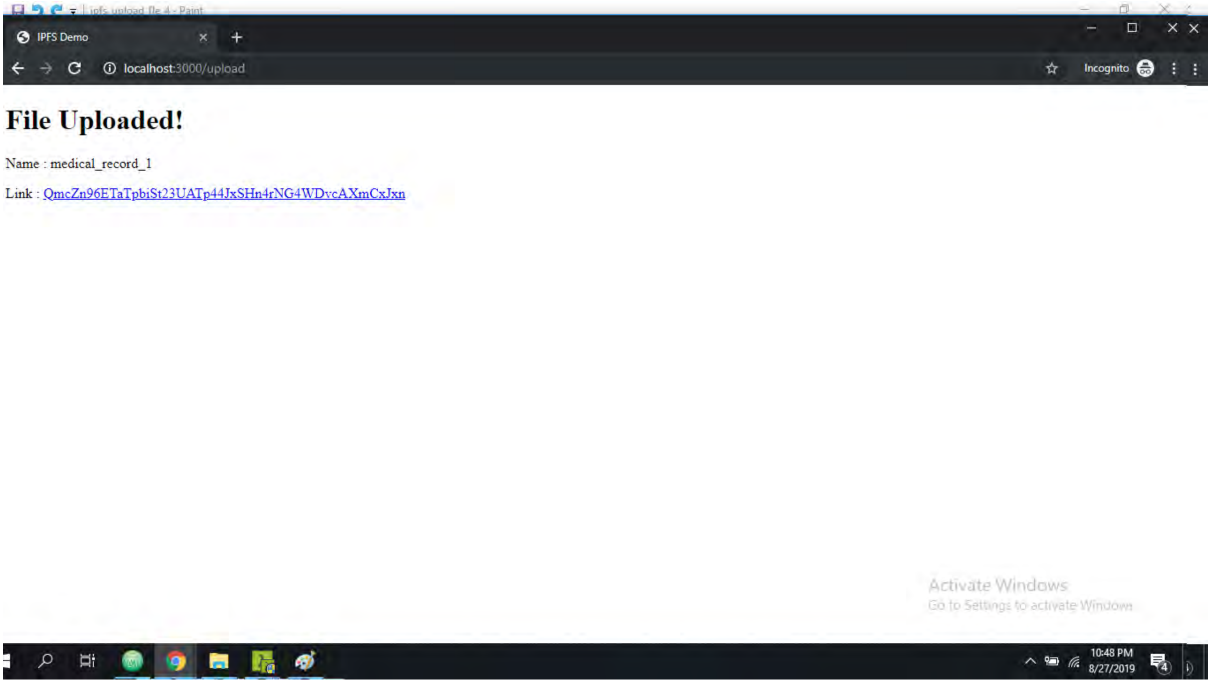
medical_record_1

Upload File

Choose File EHR1.jpg

Update Patient Informations

Figure 5.2: User Interface - After filling up info and file



File Uploaded!

Name : medical_record_1

Link : QmcZn96ETaTpbiSt23UATp44JxSHn4rNG4WDvcAXmCxJxn

Figure 5.3: User Interface - After file upload

5.2 Setting Up Necessary Software

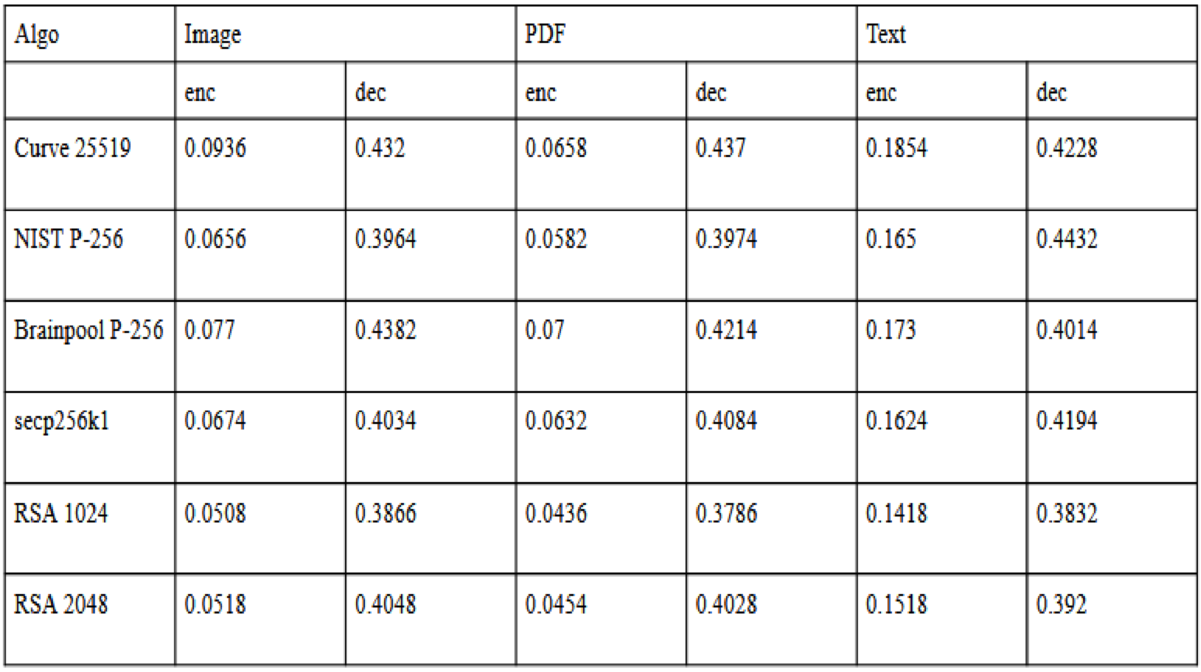
To achieve this experimental setup we used the following software stack:

- Ganache: the local blockchain testing platform
- Web3.js: the interface between the and user interface

- Remix IDE: the IDE where the smart contract is developed and tested
- Metamask: the browser bridge for browsers, Ethereum and DApps
- IPFS: the file storage system for the reports
- GPG: the PGP implementation which is used for file encryption
- Cocalc (Sage Cloud): the cloud platform where we experiment with the elliptic curve properties
- Nodejs: the back-end technology
- HTML and CSS: the front-end technology

5.3 Result and Analysis

To provide lower cost we had to choose IPFS file system as storing data in ethereum blockchain is costly. For instance, 1 MB is 1024 × 1024 (2^{20}) bytes. A word in ethereum is 256 bits, or 32 (2^5) bytes. This means that in order to store 1 MB of data, you must store $32,768\,(2^{15})$ words' worth of data. 1 MB is 1024 × 1024 (2^{20}) bytes. A word in Ethereum is 256 bits, or 32 (2^5) bytes. Using current median gas price of 5 gwei, this equates to $\frac{\sim 5\times(20{,}000\times 32{,}768)}{1{,}000{,}000}$ ETH, or ∼ 3.7628 ETH. At the current GDAX price of ∼ \$570 /ETH, that approximately equals ∼ \$1867.78 [33]. That is significantly costly so we opted for IPFS.
Average time required to encrypt and decrypt different file using different algorithm:



Algo	Image		PDF		Text	
	enc	dec	enc	dec	enc	dec
Curve 25519	0.0936	0.432	0.0658	0.437	0.1854	0.4228
NIST P-256	0.0656	0.3964	0.0582	0.3974	0.165	0.4432
Brainpool P-256	0.077	0.4382	0.07	0.4214	0.173	0.4014
secp256k1	0.0674	0.4034	0.0632	0.4084	0.1624	0.4194
RSA 1024	0.0508	0.3866	0.0436	0.3786	0.1418	0.3832
RSA 2048	0.0518	0.4048	0.0454	0.4028	0.1518	0.392

Figure 5.4: Comparison of encryption and decryption with algorithm and file types(1)

In this data table the result of the average time in seconds to encrypt and decrypt different types of files in linux command line. We have tested with image, PDF and text files. Each type of file was encrypted and then decrypted with Elliptic curves such as Curve25519, NIST P-256, Brainpool P-256, secp256k1 and RSA algorithms of key size 1024 and 2048 bits respectively.

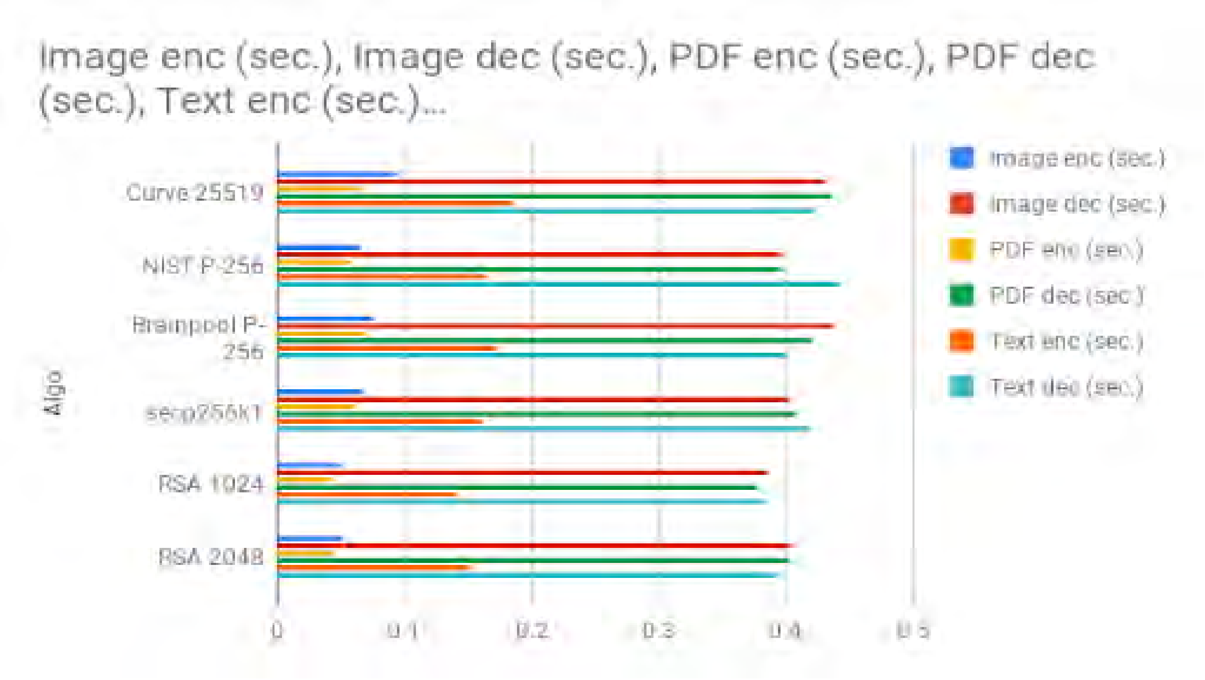
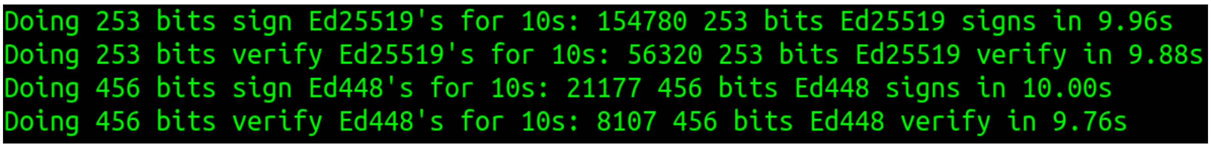
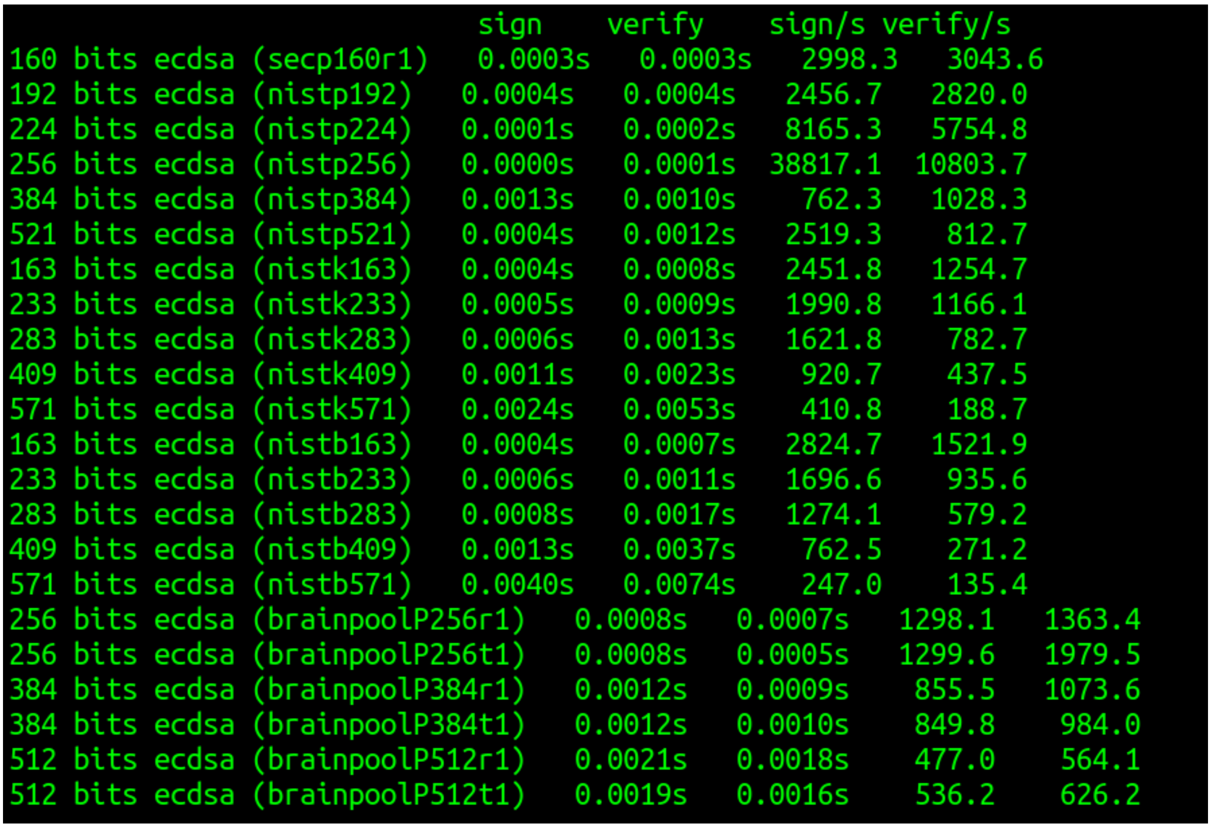


Figure 5.5: Comparison of encryption and decryption with algorithm and file types(2)



```
Doing 253 bits sign Ed25519's for 10s: 154780 253 bits Ed25519 signs in 9.96s
Doing 253 bits verify Ed25519's for 10s: 56320 253 bits Ed25519 verify in 9.88s
Doing 456 bits sign Ed448's for 10s: 21177 456 bits Ed448 signs in 10.00s
Doing 456 bits verify Ed448's for 10s: 8107 456 bits Ed448 verify in 9.76s
```

Figure 5.6: Time for EDDSA sign and verify



```
                              sign    verify    sign/s verify/s
160 bits ecdsa (secp160r1)   0.0003s   0.0003s   2998.3   3043.6
192 bits ecdsa (nistp192)   0.0004s   0.0004s   2456.7   2820.0
224 bits ecdsa (nistp224)   0.0001s   0.0002s   8165.3   5754.8
256 bits ecdsa (nistp256)   0.0000s   0.0001s  38817.1  10803.7
384 bits ecdsa (nistp384)   0.0013s   0.0010s    762.3   1028.3
521 bits ecdsa (nistp521)   0.0004s   0.0012s   2519.3    812.7
163 bits ecdsa (nistk163)   0.0004s   0.0008s   2451.8   1254.7
233 bits ecdsa (nistk233)   0.0005s   0.0009s   1990.8   1166.1
283 bits ecdsa (nistk283)   0.0006s   0.0013s   1621.8    782.7
409 bits ecdsa (nistk409)   0.0011s   0.0023s    920.7    437.5
571 bits ecdsa (nistk571)   0.0024s   0.0053s    410.8    188.7
163 bits ecdsa (nistb163)   0.0004s   0.0007s   2824.7   1521.9
233 bits ecdsa (nistb233)   0.0006s   0.0011s   1696.6    935.6
283 bits ecdsa (nistb283)   0.0008s   0.0017s   1274.1    579.2
409 bits ecdsa (nistb409)   0.0013s   0.0037s    762.5    271.2
571 bits ecdsa (nistb571)   0.0040s   0.0074s    247.0    135.4
256 bits ecdsa (brainpoolP256r1)   0.0008s   0.0007s   1298.1   1363.4
256 bits ecdsa (brainpoolP256t1)   0.0008s   0.0005s   1299.6   1979.5
384 bits ecdsa (brainpoolP384r1)   0.0012s   0.0009s    855.5   1073.6
384 bits ecdsa (brainpoolP384t1)   0.0012s   0.0010s    849.8    984.0
512 bits ecdsa (brainpoolP512r1)   0.0021s   0.0018s    477.0    564.1
512 bits ecdsa (brainpoolP512t1)   0.0019s   0.0016s    536.2    626.2
```

Figure 5.7: Time for ecdsa sign and verify

We run each of the above eddsa and ecdsa algorithms of Fig 13 and Fig 14 respectively to generate digital signature for 10 seconds in linux command line. We found that, Ed25519 generates 154780 signs in 9.96 seconds. So, to generate a single sign takes 9.96/154780= 0.000065 seconds, where the other eddsa algorithm , Ed448 takes 10/21177 = 0.0002 seconds (approximately) which is much less than the time to generate a sign using Ed25519. Also, all the ecdsa algorithms above takes a longer time for a single sign. This proves that the selected algorithm Ed25519 used for digital signature in our system produces faster signature and hence gives us better performance in signature generation than the other Digital.Signature Algorithms (DSA).

Chapter 6

Conclusion and Future Research

6.1 Conclusions

In this research work , we designed and implemented a prototype of decentralized blockchain system to preserve the interoperability of patient medical data and authorizing access to their healthcare data . It can also be used to share patient's medical records to authorized parties securely with less cost. The "Upload File" option in our system implemented using InterPlanetary File System (IPFS) [34] can store the encrypted files containing patient healthcare record at a minimal cost. After the file is uploaded to IPFS, a unique hash generated by IPFS is stored in the smart contract of our Ethereum network and this hash is used to access the file . We achieved the concept of permissioned Blockchain by proposing and implementing our own Access Control List (ACL) to to control who can access patients healthcare related files stored in IPFS will facilitate the patients with data sharing option with trustworthy authorities and thus provides authorization and accountability to patients. Since anyone can access files stored in IPFS, we stored encrypted files which can only be decrypted by authorities who got access from the access control list in the blockchain network. To have better security, we used Curve25519 for encryption, decryption and Ed25519 for faster digital signature generation and verification[35] which protect the privacy of patient's sensitive data .

6.2 Future Plan

- Introducing robust and easier user interface that will be beneficial for users specially who are technologically not so advanced.

- We implemented a secure file sharing system to share patients healthcare related files with less cost. We need to implement the access control procedure more rigorously, i.e role based access control to ensure better interoperability and better performance in terms of authorized access control. Also, we will need to follow better software design patterns and procedures to make our system scalable.

- We are also interested to work on Cryptography that will be secure from quantum attack .

For this research, we used IPFS to store and share files at low cost, In future, we will be looking forward to design and implement our own decentralized and scalable file sharing system.

6.3 Limitations

Since mainnet is very costly and requires high performance computer to run our desired application containing smart contracts in the Ethereum network, we had to run and test our blockchain application locally using and testnet .Although GPG has the option for creating new keys using Curve25519 for encryption and digital signature, sometimes it shows a warning that Curve25519 is not a part of the OpenPGP standard. For this reason we had to go through a couple of extra setup steps to be able to use Curve25519 for encryption , decryption and digital signature using Ed25519. Though we tried to make the system more user friendly, upto this point , the patients who are technologically less advanced may face difficulties operating the system since there are many options involved related to permissions . The system is not very scalable as it needs IPFS server to run and servers comes with additional cost. Moreover, to gain maximum benefit from the proposed system , an improved definition of transactions between different types of users is needed.

- This is a system designed for expert user; average user will not be able to use it fluently.
- The user have not sign in process as of yet
- The system it still vulnerable to quantum attacks
- The system is not very scalable as it needs IPFS server to run and servers comes with additional cost.

Chapter 7

References

[1] RFC 8032 - Edwards-Curve Digital Signature Algorithm (EdDSA). (2019). Retrieved 27 August 2019, from https://tools.ietf.org/html/rfc8032.

[2] “What Is an Ethereum Testnet and How Is It Used?,” Bitfalls, 31-May-2018. [Online]. Available: https://bitfalls.com/2018/05/31/what-is-an-ethereum-testnet-and-how-is-it-used.

[3] N. community, “TestRPC,” TestRPC - Nethereum Documentation. [Online]. Available: https://nethereum.readthedocs.io/en/latest/ethereum-and-clients/test-rpc.

[4] “What Is A Cryptocurrency Wallet?,” ETHNews.com, 29-Mar-2017. [Online]. Available: https://www.ethnews.com/what-is-a-cryptocurrency-wallet.

[5] L. Fortney, “Blockchain Explained,” Investopedia, 25-Jun-2019. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp.

[6] ”Smart Contracts—A Simple yet Comprehensive Explanation in Pictures”, Hackernoon.com, 2019. [Online]. Available: https://hackernoon.com/smart-contracts-a-simple-yet-comprehensive-explanation-in-pictures-bc21c7ab89b6. [Accessed: 02-Aug- 2019].

[7] “What’s a DApp?,” State of the DApps - What’s a DApp. [Online]. Available: https://www.stateofthedapps.com/whats-a-dapp. [Accessed: 02-Aug-2019].

[8] J. Frankenfield, “Smart Contracts: What You Need to Know,” Investopedia, 04-Aug-2019. [Online]. Available: https://www.investopedia.com/terms/s/smart-contracts.asp. [Accessed: 05-Aug-2019].

[9] L. Gaines, “What is Ethereum?,” Medium, 19-Feb-2018. [Online]. Available: https://medium.com/@eightlimbed/what-is-ethereum-e6c37bc30f1a. [Accessed: 02-Aug-2019].

[10] Aziz, “Crypto Mainnet vs Testnet: What is the Difference?,” Master The Crypto, 11-Mar-2019. [Online]. Available: https://masterthecrypto.com/mainnet-vs-testnet-whats-the-difference. [Accessed: 27-Jul-2019].

[11] “Solidity,” Solidity. [Online]. Available: https://solidity.readthedocs.io/en/v0.5.10. [Accessed: 27-Jun-2019].

[12] “What is Hashing? - Definition from Techopedia,” Techopedia.com. [Online]. Available: https://www.techopedia.com/definition/14316/hashing. [Accessed: 25-Jul-2019].

[13] “What is Public Key Cryptography (PKC)? - Definition from Techopedia,” Techopedia.com. [Online]. Available: rhttps://www.techopedia.com/definition/9021/public-key-cryptography-pkc. [Accessed: 26-Jul-2019].

[14] “Elementary number theory: primes, congruences, and secrets: a computational approach,” Choice Reviews Online, vol. 47, no. 02, Jan. 2009.

[15] “Edwards curves,” Explicit-Formulas Database / Edwards curves. [Online]. Available: https://hyperelliptic.org/EFD/oldefd/edwards.html. [Accessed: 05-Aug-2019].

[16] “Edwards-Curve Digital Signature Algorithm (EdDSA),” IETF Tools. [Online]. Available: https://tools.ietf.org/html/rfc8032. [Accessed: 27-Jul-2019].

[17] D. J. Bernstein, “Curve25519: New Diffie-Hellman Speed Records,” Public Key Cryptography - PKC 2006 Lecture Notes in Computer Science, pp. 207–228, 2006.

[18] “IPFS Documentation,” What is IPFS? – IPFS Documentation. [Online]. Available: https://docs.ipfs.io/introduction/overview. [Accessed: 03-Aug-2019].

[19] IETF. [Online]. Available: https://www.ietf.org/rfc/rfc4880.txt. [Accessed: 05-Aug-2019].

[20] “The GNU Privacy Guard,” The GNU Privacy Guard. [Online]. Available: https://gnupg.org/.

[21] D. Sewell, M. Naval, A. Dean, C. Faraguna, J. Gomila, and M. Manipula, “Metamask - Wiki,” Golden. [Online]. Available: https://golden.com/wiki/Metamask.

[22] J.-J. Yang, J.-Q. Li, and Y. Niu, “A hybrid solution for privacy preserving medical data sharing in the cloud environment,” Future Generation Computer Systems, pp. 43–44, 2015.

[23] “Health Information Privacy,” HHS.gov, 04-Jan-2019. [Online]. Available: http://www.hhs.gov/hipaa. [Accessed: 20-Jul-2019].

[24] “Electronic Code of Federal Regulations,” Electronic Code of Federal Regulations. [Online]. Available: https://www.ecfr.gov/. [Accessed: 20-Jul-2019].

[25] Office for Human Research Protections and Ohrp, “45 CFR 46,” HHS.gov, 16-Feb-2016. [Online]. Available:
https://www.hhs.gov/ohrp/regulations-and-policy/regulations/45-cfr-46/. [Accessed: 20-Jul-2019].

[26] “Lex Access to European Union law,” EUR. [Online]. Available: https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:31995L0046. [Accessed: 20-Jul-2019].

[27] S. Khezr, M. Moniruzzaman, A. Yassine, and R. Benlamri, “Blockchain Technology in Healthcare: A Comprehensive Review and Directions for Future Research,” Applied Sciences, vol. 9, no. 9, p. 1736, 2019.

[28] T. Nugent, D. Upton, and M. Cimpoesu, “Improving data transparency in clinical trials using blockchain smart contracts,” F1000Research, vol. 5, p. 2541, 2016.

[29] L. A. Koo and M. B., Use of Blockchain for Healthcare and Research Workshop. Maryland, United States: Gaithersburg, 2016.

[30] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “MedRec: Using Blockchain for Medical Data Access and Permission Management,” 2016 2nd International Conference on Open and Big Data (OBD), 2016.

[31] J. Sandgaard and S. Wishstar, “MedChain White Paper,” v. 1.

[32] P. Zhang, J. White, D. C. Schmidt, and G. Lenz, “Applying Software Patterns to Address Interoperability in Blockchain-based Healthcare Apps,” 2017.

[33] S. Guo, “Cypherium: A Scalable and Permissionless Smart Contract Platform,” whitepaper.

[34] J. Benet, “Ipfs-content addressed, versioned, p2p file system,” arXiv preprint arXiv:1407.3561, 2014.

[35] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, “High-speed high-security signatures,” Journal of Cryptographic Engineering, vol. 2, no. 2, pp. 77–89, 2012.

[36] M. Steichen, B. Fiz, R. Norvill, W. Shbair and R. State, ”Blockchain-Based, Decentralized Access Control for IPFS”, IEEE Xplore, 2018. Available:
https://ieeexplore.ieee.org/abstract/document/8726493. [Accessed 28 August 2019].

[37] D. J. Bernstein, “Curve25519: New Diffie-Hellman Speed Records,” Public Key Cryptography - PKC 2006 Lecture Notes in Computer Science, pp. 207–228, 2006.

[38] “GNU wannabee and Happy Hacking,” GNU wannabee and Happy Hacking RSS. [Online]. Available: https://www.gniibe.org/memo/software/gpg/keygen-25519.html.

[39] P. Schwabe, D. Bernstein, N. Duif, and T. Lange, “CARAMEL seminar,” in CARAMEL seminar, 20-Mar-2012.

[40] S. Nakov, Elliptic curves in the elliptic curve cryptography (ECC) may be presented in several forms (representations), which are proven to be birationally equivalent (isomorphic). 2019.