MedRec: A Network for Personal Information Distribution

Nchinda Nchinda, Agnes Cameron, Kallirroi Retzepi, Andrew Lippman
MIT Media Lab
Massachusetts Institute of Technology
Cambridge, MA, 02139, USA
Email: nchinda2@mit.edu, agnescam@mit.edu, kalli@mit.edu, lip@media.mit.edu

Abstract—**MedRec is a simple, distributed system for personal control of identity and distribution of personal information. The work is done in the context of a medical information distribution system where patients retain control over who can access their data. We present a new architecture for the MedRec project, creating a network of trusted data repositories, the access to which are determined by a set of ”smart contracts”. These contracts are stored on a distributed ledger maintained by those who generate data. The distributed nature of the system allows unified access from diverse sources in a single application with no intermediary. This increases patient control while retaining a measure of privacy of both data content and source. MedRec is amenable to extensions for decentralized messaging and distribution of information to third parties such as medical researchers, healthcare proxies, and other institutions. The system is based on a blockchain that contains smart contracts defining user identity and distribution specifics.**

Index Terms—**blockchain, medical records, healthcare, distributed, ethereum**

I. Introduction

Personal information about all of us is generated constantly, deliberately, automatically, and autonomously through almost all of the interactions that we engage in. Most of that information is siloed and used for a variety of purposes, some of which add great convenience to our lives. On occasion, that data is also shared, aggregated, processed and re-distributed. Also, on occasion the results of this activity are available to the individuals to which that data pertains. Recent laws such as GDPR, passed in the EU but becoming an international model, facilitate this.

We elected to develop a simple and usable system to support an individual's control over the distribution of data about themselves that is not held by them. We chose the domain of medical information because in many ways, it exemplifies the situation noted above: Increasingly, people in the United States are required to manage their own healthcare and associated information. At the same time, healthcare providers have to make that data available. This circumstance opens the door for innovative approaches to patient management. One obvious solution is a ”Swiss bank” for healthcare records. This offloads engaged patient management to a cross-provider intermediary. In return for that convenience, we risk the addition of new data silos and commercial control points.

MedRec implements a non-commercial, distributed system that allows people to control who can access their records. MedRec is a *network* comprised of trusted medical providers who make data available on-demand at the behest of patients under the control of patient-created ”smart contracts”. These are stored on a blockchain and maintained at sites that originate records (and therefore hold data).

MedRec user interface is a single place where all data from all providers can be accessed; it provides a single point of entry to a diversity of healthcare providers. Current web-based solutions are unique for each provider – there is no simple way to merge information, notifications, or manage distribution that crosses provider boundaries. This is an important motivating factor in the design – Medrec is not a web app, instead it runs securely in a client that runs on a phone or a computer.

The architecture is not health specific. We envision that it can be a model for the management of individual identity and permissions in many circumstances where end-user control of identity and personal information across applications is important. This can be a basis for social networks and as a convenience for individuals who want to simplify who knows what about them. There is no coinage or transaction inherent in MedRec; it is designed to be free and open.

MedRec was inspired by original work by Ariel Ekblaw and Asaph Azaria [1]. The current version, which is a new architecture, is supported by a grant from the Robert Wood Johnson Foundation. Ekblaw and Azaria envisioned a blockchain mined by researchers who would be paid with access to data for research; the current version replaces mining with proof of authority granted to a network of trusted providers. This conserves energy and comports with the fact that records originators have an a priori commitment to managing them securely. There are no transactions and no coinage associated with the system, and therefore no economic attached to records distribution. In addition, we have a fully open source, open system where multiple users access data across providers. It is scalable and useful. The blockchain itself contains no medical information, only permission contracts that are translated into database queries on demand. The ensemble of a users contracts is available as a network connection diagram.

MedRec is managed using an Ethereum blockchain, and has been tested with databases provided by our research partner, the Massachusetts-based Beth Israel Deaconess Medical Center. Further development will be done by a new, non-profit research endeavor called the Health Technology Innovation

Center operated at BIDMC with continued participation by a team at the MIT Media Lab.

II. Overview and Background

A. System Overview

The architecture of MedRec is easily understood by analogy to the World Wide Web. The web consists of three elements: An HTTP server that provides access to local data, the HTML language by which web elements are defined, and a browser that forms the interface. Ideally, anyone and everyone could be a server and web browsers can draw from multiple ones to create a presentation. The World Wide Web is by design a network rather than a client-server architecture even though in practice there are dominant servers.

In MedRec, the language analogue is a set of smart contracts that define what entities or parties can access which records. These contracts are managed using a blockchain, which is analogous to the web server. The contracts are redundantly stored by each site that maintains the blockchain. Unlike the web, our servers, which provide access to and provide data based on the contracts, are linked and maintain a common blockchain. These servers are also the administrative members of the network. Each entity in the system has a unique ID that identifies them from each other. These terms and functions will be explained below. [Figure 1]

The system works by linking together four components: (1) A desktop application by which patients access their data and define who else can view it, (2) a daemon that interacts with provider databases and the underlying filesystem, (3) Ethereum clients which connect to each other to form the MedRec network and manage the blockchain, (4) an Ethereum blockchain which is used to manage access rights to medical records. In essence, patients create access contracts, which are stored on the blockchain. These contracts specify to whom the providers should release medical record information.

B. Blockchain

Bitcoin [2] is the original use of blockchain technology and it is still widely considered the most stable and secure implementation. Data stored on the Bitcoin blockchain can generally be assume to be indelible. The integrity of the blockchain is maintained by a costly ”proof of work” algorithm executed by all nodes.

Every transaction on the Bitcoin network costs money and takes time. One project using the Bitcoin blockchain for land title registry in the Republic of Georgia has an estimated 5-10 cents cost per registration [3]. This is feasible in a system where land ownership changes slowly, on the order of years, but not at the scale of MedRec.

The public Ethereum [4] network is an alternative and has some of these same limitations. Ethereum includes a Turing complete programming language that one can use to store executable agreements, called ”smart contracts” [5]. A full node[1] on the network must process all the transactions and blocks generated on the network. An Ethereum transaction on average takes 15 seconds to confirm and costs $.30.

Ethereum also introduced an alternative system of block validation called Proof of Authority (PoA) as an alternative to the more common Proof of Work. While the main Ethereum public chain runs on PoW, there are multiple test-nets and private chains running on PoA. PoA operates by a federated model, only certain nodes are allowed to make additions to the blockchain. This changes the security model: the system is more susceptible to fraudulent transactions by way of one of those authorities getting hacked. At the same time, it allows faster transaction confirmation times and the ability to remove the cryptocurrency aspect from blockchain technology.

Private blockchains bridge the gap between public blockchains, which are trustless and pseudonymous, and databases, which are centralized. In networks where certain entities are trusted but subject to attack it may be prudent to use a private blockchain instead of public with those entities as the custodians.

C. Privacy

All transactions on the public Ethereum network must be stored by all the full nodes on the network. As a result, everyone who has access to the blockchain can determine all relationships that any patient has. In a medical context, that metadata is sensitive. In addition, static analysis has been used to de-anonymize users of Bitcoin years after a hack [6]. Since data on the blockchain is persistent, it lacks retroactive identity confidentiality [7]. At some point in the future an algorithm may be developed to correlate and de-anonymize many users of a public blockchain.

A recent advancement from Narula et. al is zkLedger [8], which masks the sources and destinations of transactions by making it appear that one is making a transaction with everyone on the network for each exchange. The drawback is transaction creation and verification times are dependent upon the number of nodes in the network, scaling linearly with nodes.

In the more scalable space of off-chain solutions, Enigma [9] is a privacy protocol that conceals user data from nodes that execute computations, allowing sensitive information to be handled by ”secret” contracts. Enigma manages data off-chain, although computation on that data is managed differently. Secure multi-party computation based on Shamir's Secret Sharing allows ”secret” data to be operated on by external apps without revealing any information.

In Enigma, the blockchain serves as a point of external control and a guarantee of correctness, while Enigma itself ensures privacy for the user. Off-chain nodes construct a distributed database, distinct from (but analogous to) the consortium of provider-controlled databases in MedRec.

III. Architecture

A. The MedRec Blockchain

MedRec uses a Proof of Authority blockchain developed as part of the Go-Ethereum client. The data stored on the

[1] A node that keeps an entire record of the blockchain

blockchain can still be publicly read outside of the federated set of nodes maintaining the blockchain.

Medical providers are voted in as authorities by the existing set of full nodes. An authority is able to form blocks and add them to the blockchain. Since the identities of all providers are known, regulation can be external to the blockchain. The penalties for abusing their abilities as an authority would be loss of their trusted role.

Since each transaction is public, other providers are able to validate every transaction as well. A provider is incentivized not to generate fraudulent data because they will be easily identified. Furthermore the type of attacks on the system by providers is limited since their only duty is to confirm transactions. Providers cannot impersonate other actors in the system or generate false permissions for their patients as they do not have those private keys.

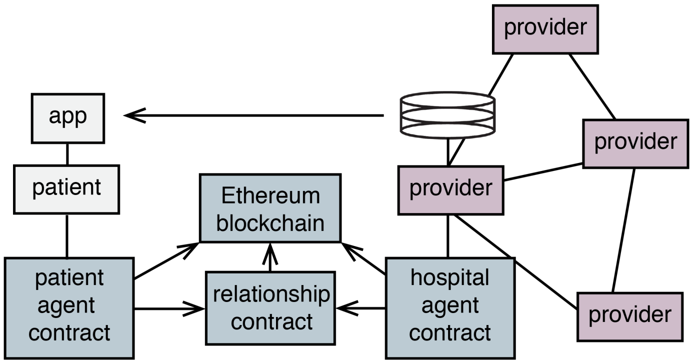


Fig. 1. The MedRec Network uses a series of smart contracts to interface between different agents. On the left, a patient is represented by an ”Agent Contract”, and navigates the system via an Electron app on their phone or computer. On the right, the patient’s data is stored by a hospital (represented by a second Agent Contract), which is in turn part of a consortium of trusted medical providers. All the smart contracts are recorded on the Ethereum blockchain. When a patient opens the app, a call is made to the Database Manager of providers that have a Relationship Contract with that patient. The patient’s unique Ethereum address is checked against that contract, and records associated with that request are displayed on the patient’s front end.

B. Storage Efficiency

Blockchains require all maintainers to store and reference the entire blockchain. We restrict the data on chain to a small set: the list of administrative authorities, and the smart contracts required to validate agents and define relationships. Every patient must store some data on-chain. They execute a 9727 byte transaction to create an Agent contract. Then each relationship between a patient and provider requires a 5007 byte transaction. Updates to contracts require 220 bytes.

We can estimate how many bytes need to be stored by every node. Assuming 350 million users (approx. US population) who each form relationships with 5 different providers, the blockchain would be 12 Terabytes in size. For a smaller case, such as Massachusetts, it can be as small as 290 gigabytes if every citizen has a relationship with two providers. Given the existing mandate [10] for hospitals to share medical records with their patients, and the comparative size of hospital information management systems [11], the size associated with managing the MedRec blockchain is a small price to pay.

C. Security and Convenience

MedRec runs as a desktop application, but we envision future implementations running on mobile devices. The blockchain permits application developers to make trade-offs between security and convenience. In the current version of MedRec every node verifies and propagates all transactions and blocks. Other types of nodes can adjust how much of the blockchain is stored and verified. *Pruned nodes* only keep a fixed number of the latest blocks. Other nodes, known as *light nodes*, only download the block headers, resulting in massive space savings but relying more on the existence of trusted nodes elsewhere to verify transactions. As MedRec has provider nodes with public identities maintaining the blockchain, future applications running on mobile devices can use a light nodes without a detrimental effect to the network’s security.

D. Bootstrapping

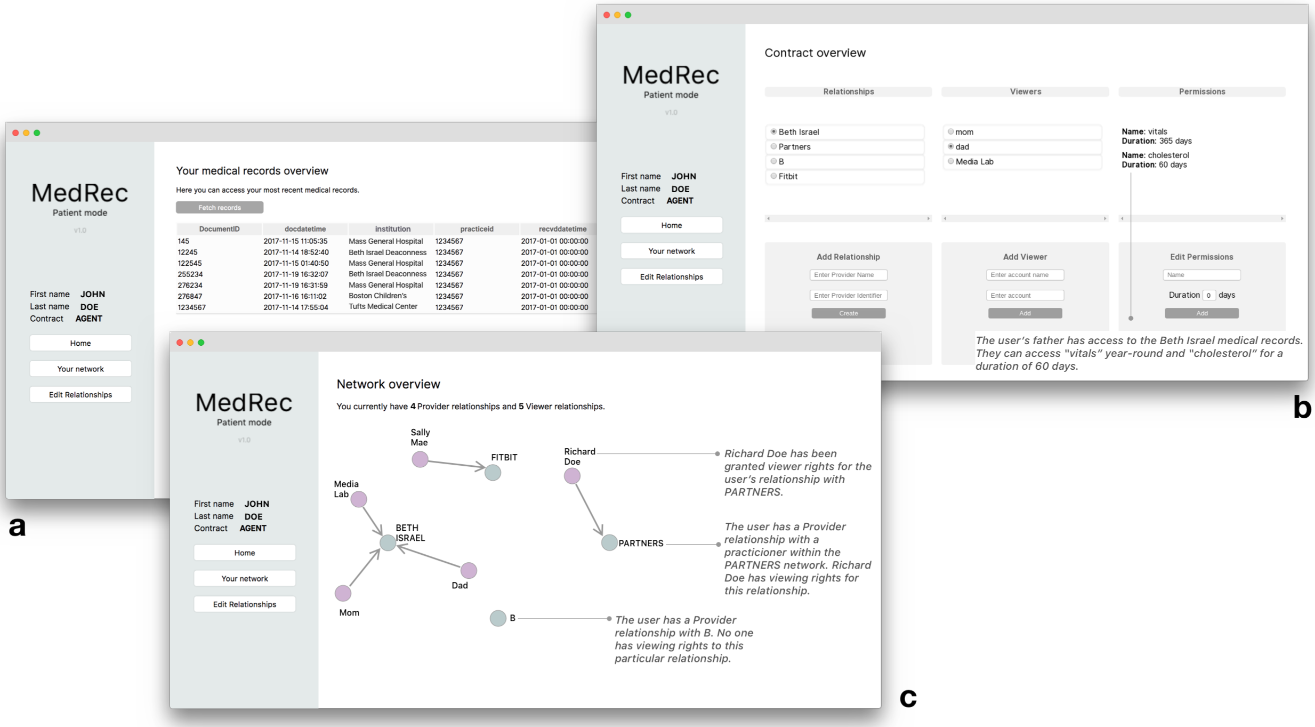
Even with a decentralized network, blockchains generally retain a centralized aspect in the form of bootnodes. A bootnode is a node on the network with the sole purpose of matchmaking new users with nodes already on the network. They are akin to DNS root authorities and BitTorrent trackers. These nodes are hard-coded into the MedRec client. When MedRec is operational they should at minimum be set to a few well known public healthcare providers on the network and at maximum all providers. Only one of these providers needs to be non-malicious in order for a user to safely join the network.

Past research on Bitcoin [12] and Ethereum [13] has analyzed the practicality of an attacker assuming control over a node’s connection to the rest of the network. These attacks are generally called eclipse attacks. While the victim believes they are well connected to the network, all peering connections from the victim’s node lead directly to an attacker who is able to manipulate their interaction with the blockchain. Using a PoA blockchain adds another mechanism for mitigation of these attacks. One way is by requiring all nodes to maintain a connection to at least one provider node at all times. The current set of provider nodes can be hard-coded into each version of the MedRec software as the nodes used for bootstrapping.

IV. Healthcare Specific Aspects

A. Anonymizing Metadata

An unintended side effect of using a single smart contract to represent a patient (the Agent contract) is that anyone who can link a patient’s real-world identity to their Ethereum address can then determine all healthcare providers with whom that patient has a relationship. A case where this would be harmful is when a patient shares portions of their medical records with both their employer and a gynecologist. The employer could detect this association and find a pretense to discharge



Your medical records overview

Here you can access your most recent medical records.

DocumentID	docdatetime	institution	practiceid	recvddatetime
145	2017-11-15 11:05:35	Mass General Hospital	1234567	2017-01-01 00:00:00
12245	2017-11-14 18:52:40	Beth Israel Deaconness	1234567	2017-01-01 00:00:00
122545	2017-11-15 01:40:50	Mass General Hospital	1234567	2017-01-01 00:00:00
255234	2017-11-19 16:32:07	Beth Israel Deaconness	1234567	2017-01-01 00:00:00
276234	2017-11-19 16:31:59	Mass General Hospital	1234567	2017-01-01 00:00:00
276847	2017-11-16 16:11:02	Boston Children's	1234567	2017-01-01 00:00:00
1234567	2017-11-14 17:55:04	Tufts Medical Center	1234567	2017-01-01 00:00:00

First name JOHN
Last name DOE
Contract AGENT

Contract overview

Relationships	Viewers	Permissions
● Beth Israel	○ mom	**Name:** vitals **Duration:** 365 days
○ Partners	● dad	**Name:** cholesterol **Duration:** 60 days
○ B	○ Media Lab	
○ Fitbit		

Add Relationship
Enter Provider Name
Enter Provider Identifier

Add Viewer
Enter account name
Enter account

Edit Permissions
Name
Duration 0 days

The user's father has access to the Beth Israel medical records. They can access "vitals" year-round and "cholesterol" for a duration of 60 days.

Network overview

You currently have **4** Provider relationships and **5** Viewer relationships.

Richard Doe has been granted viewer rights for the user's relationship with PARTNERS.

The user has a Provider relationship with a practicioner within the PARTNERS network. Richard Doe has viewing rights for this relationship.

The user has a Provider relationship with B. No one has viewing rights to this particular relationship.

Fig. 2. Three views of the MedRec interface. (a): Medical records from a diversity of providers may be viewed in the landing page of the MedRec interface. (b): The Contract Overview provides an interface where patients may view, create and edit relationship contracts. Patients can specify who can access specific data and define limitations, such as a time period. This page evolves as new contracts are designed. (c): Patients are presented with a visualization of their relationships as a network (trusted providers in grey, other patients or third parties with viewing rights in pink).

their employee to avoid paying health insurance. In addition, it might be possible to link a patient's identity and Ethereum account by examining the hospitals they have a relationship with [14].

We solve this in two steps. The first step is disassociating each patient identity from provider identities. Each provider makes a new Ethereum account for each patient provider relationship. We call these patient specific accounts *delegate accounts*. This method allows a patient to have public relationships without revealing the real world identities of the individual members of those relationships.

Even though a patient is not interacting directly with their provider's main account, that main account is still being used to provide the patient with ether for their transactions. Over time this correlation could be used to associate patients with providers. Thus, instead of fulfilling patient ether requests themselves, a provider asks another provider to do it for them, further masking the relationship.

B. Using MedRec as a Patient

The MedRec app is essentially a wallet that contains their Ethereum private key. This key is stored in a password-protected vault that is unlocked when the user logs in. As the app is designed to support multiple accounts, multiple users can use the same hardware without compromising their individual security. The patient has (currently) three views of their medical information status. The first (figure 2-a) shows records that are available and might have been changed since a previous view. This is a prototype for any records information system. Figure 2-b and 2-c show contract-specific presentations, the first being the details of specific contract, and the second being an overview of the current permission status.

We provide for recovery of a lost key using a standard first developed for Bitcoin that is in use in a number of blockchain wallets (such as Metamask and MyEtherWallet). A representation of a user's private key consisting of common English words [15] is generated when the account is created, along with instructions to the user to ensure that they note this information down. Much as people are expected to take care of important documents such as a social security number, the 12-word seed would be the single document a patient would need to keep safe (in contrast to the many currently required to interact with existing medical care systems). We note that even if all account information is lost, the patient can return to their medical provider, and create a new account.

The sample app is written using React JS, in an Electron wrapper, allowing it to be adapted readily to a range of devices, including mobile phones.

C. Data Retrieval

When the patient logs into the app, data is retrieved from each provider that they have a relationship with, and visualized in the interface. This polling takes the form of a Remote Procedure Call to the provider, digitally signed with the patient's Ethereum address. This signature is decoded by the Database Manager, and matched against a (hashed and salted [16]) key/value store (implemented using LevelDB), uniquely identifying the patient in that database from their Ethereum address. This key/value store also means that no extra information needs to be stored in the provider database the unique id representing the patient in their database is simply mapped to that representation in the Ethereum network.

D. Using MedRec as a Third Party

There are additional entities that are likely to join the MedRec network as partners in healthcare rather than providers or patients. Examples might include a patient's pharmacist or clinical research organizations. Given the trust model of MedRec where the network is administrated by the originators of medical records these entities should not control the blockchain directly. However, these entities should also have a different status in the network different to that of patients.

Third parties would be the subject of extensible and modifiable smart contracts. A patient, might, for example, give a pharmacist access to all prescription information, and in turn receive notification when their medication is available. A clinical trial or epidemiological study might want to access patient information from participants, and the patient would issue a contract permitting that. The patient may also permit access to information even when they are not at the moment engaged in a trial. Interactions between these third parties and patients are mediated by medical providers to ensure that the data will not be misused, and there is deliberately no coin or token associated with these interactions. The notion of a "viewing contract" is also applicable between patients in a network, where, for example, a couple might grant one another viewing rights to their records.

V. Conclusions and Future Work

There are several elements to adoption of a MedRec network that are subjects of further work and development.

Most important is the means by which providers adopt and interface to the system. A provider commits to run a program that grants access to their databases under the rules of MedRec contracts. This entails an interfacing investment that can be significant. For large providers who already use an existing patient management application, this need be done once for that system and others can then use it. For smaller providers such as group practices, one must build an interface for each system that is in use.

We argue that Ethereum-supported proof of authority mechanism is a robust solution. The overhead of running a full node is small both in terms of management and allocation of resources. Conversely, the advantages are large. The open-source model allows us to evolve with needs and community desires. These issues are assertions that will be tested at scale in real use. This system has yet to be fully tested in a clinical setting, and thus we cannot fully evaluate claims of scale.

We suspect that individual management of personal data is a chore akin to management of a retirement plan. They are similar in that when we are young and healthy, we likely dedicate little energy to either retirement or healthcare. It has been amply demonstrated that people devalue longterm or low probability events [17]. A good interface may ameliorate this. To date, the interface we have implemented is optimized to be simple, and encouraging. As we add features that are common in commercial healthcare interfaces, we have to ensure that the system does not become a chore to use.

Acknowledgment

This version of MedRec was funded by the Robert Wood Johnson Foundation, and is based on the original prototype by Ariel Ekblaw and Asaph Azaria.

References

[1] Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). MedRec: Using Blockchain for Medical Data Access and Permission Management. 2016 2nd International Conference on Open and Big Data (OBD). doi:10.1109/obd.2016.11

[2] Nakamoto, S. (2008, November 1). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved June 21, 2018, from *https : //bitcoin.org/bitcoin.pdf*

[3] Shin, L. (2017, July 17). The First Government To Secure Land Titles On The Bitcoin Blockchain Expands Project. Retrieved July 10, 2018, from *https : //www.forbes.com/sites/laurashin/*2017/02/07/*the − first − government − to − secure − land − titles − on − the − bitcoin − blockchain − expands − project/*#7*fd*9*c*7*d*94*dcd*

[4] E. (2018, July 1). Ethereum/wiki. Retrieved July 9, 2018, from https://github.com/ethereum/wiki/wiki/White-Paper

[5] Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. First Monday, 2(9). doi:10.5210/fm.v2i9.548

[6] Nilsson, K. (2017, July 27). Breaking open the MtGox case, part 1. Retrieved from *https : //blog.wizsec.jp/*2017/07/*breaking − open − mtgox −* 1*.html*

[7] Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2017). A survey on the security of blockchain systems,. Future Generation Computer Systems, 73. *doi* : *https* : *//doi.org/*10.1016/*j.future.*2017.08.020

[8] Narula, N. Vasquez, W., & Virza, M. (2018). zkLedger: Privacy-Preserving Auditing for Distributed Ledgers. USENIX Symposium on Networked Systems Design and Implementation, 80.

[9] Zyskind, G., Nathan, O., & Pentland, A. S. (2015). Decentralizing Privacy: Using Blockchain to Protect Personal Data. 2015 IEEE Security and Privacy Workshops. doi:10.1109/spw.2015.274

[10] HHS Office of the Secretary,Health Information Privacy Division. (2016, February 25). Individuals' Right under HIPAA to Access their Health Information. Retrieved July 20, 2018, from *https : //www.hhs.gov/hipaa/for − professionals/privacy/guidance/access/index.html*

[11] Halamka, J. (2011, April 6). The Cost of Storing Patient Records. Retrieved July 20, 2018 from *http : //geekdoctor.blogspot.com/*2011/04/*cost − of − storing − patient − records.html*

[12] Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015, October 04). Eclipse Attacks on Bitcoin's Peer-to-Peer Network MIT Security Seminar Medium. Retrieved July 13, 2018, from *https : //eprint.iacr.org*

[13] Marcus, Y., Heilman, E., & Goldberg, S. (2018, January 9). Low-Resource Eclipse Attacks on Ethereum's Peer-to-Peer ... Retrieved July 13, 2018, from *https : //eprint.iacr.org/*2018/236*.pdf*

[14] Sweeney, L. (2002). K-Anonymity: A Model For Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05), 557-570. doi:10.1142/s0218488502001648

[15] Palatinus, M., Rusnak, P., Voisine, A., & Bowe, S. (2013, September 13). Bitcoin/bips. Retrieved June 21, 2018, from *https : //github.com/bitcoin/bips/blob/master/bip −* 0039*.mediawiki*#*wordlist*

[16] Ducklin, P. (2017, August 07). Serious Security: How to store your users' passwords safely. Retrieved June 21, 2018, from *https : //nakedsecurity.sophos.com/*2013/11/20/*serious − security − how − to − store − your − users − passwords − safely/*

[17] Tversky A. & Kahneman D. (1974, Sep. 27), Judgment under Uncertainty: Heuristics and Biases, Science, New Series, Vol. 185, No. 4157., pp. 1124-1131