Blockchain-based self-sovereign identity applications in healthcare

bachelor thesis

Author
Sytse Oegema
s3173267

Supervisor
Fatih Turkmen

secundary supervisor
Mohammed Alghazwi

Faculty of Science and Engineering
University of Groningen
The Netherlands
July 8, 2020

Abstract

This thesis researches different requirements on electronic health records and how to meet these requirements with a self-sovereign identity via blockchain. Patients have difficulty sharing only parts of their electronic health records with medical experts and are unable to create personal attachments to electronic health records. Self-sovereign identity provides a user-controlled digital identity that facilitates the partial sharing of data and the creation of personal annotations. However, also a patient's consent is required to access identity-related data, obstructing emergency access. To solve these problems multiple protocols are introduced by means of literature research and a proof of concept that utilizes Hyperledger Indy's blockchain-based self-sovereign identity foundation. Partial sharing of data and the addition of personal annotations to the electronic health records are included by the intrinsic properties of self-sovereign identity. An offline protocol facilitates emergency access via the user's identity holding device and an online protocol facilities emergency access via Shamir's Secret Sharing with trusted parties. Both emergency access protocols are secured by validatable self-sovereign doctor identities. All protocols are designed according to self-sovereign properties and particularly focus on user control and consent. Hyperledger Indy in combination with the designed protocols enables self-sovereign handling of electronic health records facilitated by a blockchain.

Contents

1 Introduction **3**

2 Related Work **4**

3 Conceptual Design **7**

3.1 Hyperlegder Indy 7

3.2 EHR features 8

4 Implementation **11**

4.1 Government Instance 11

4.2 Doctor 12

4.3 Patient 13

5 Results **17**

5.1 Hyperledger Indy 17

5.2 EHR features 18

6 Conclusion **21**

7 Future Work **22**

References **23**

1. Introduction

Efficient and complete handling and sharing of electronic patient data between patients and medical professionals amplify the quality of healthcare. The sharing of private medical data has always been complicated due to privacy and security issues. Additionally, electronic medical data may only be shared with the consent of the patient, which raises another issue. Blockchain technology offers a promising new solution that enables the secure integration of patient health information between different institutions[1]. Self-sovereign identity, a new type of online identity, places patients in the center of control over their own private medical data. Research has already indicated the potential of blockchain-based self-sovereign identity in healthcare[2].

Traditionally each care provider would have its own system for electronic medical record administration. The inconvenience of sharing medical records with different care providers rests in the different types of storage systems and their incompatible data exchange. Electronic health record systems have been designed to facilitate the sharing of medical data, for example, using cloud computing. However, these systems still contain cost and security concerns, access and ownership problems, and liability issues[3].

Blockchain-based self-sovereign identity puts users in control of their digital identity and the data that is associated with it. In contrast to traditional identity mechanisms, this mechanism utilizes the distributed ledger foundation of blockchain technology thereby removing the need of a third party for authentication. The distributed ledger facilitates the immutable storage of system transactions and real-time recordings. Blockchain-based self-sovereign identity has potential for an application in healthcare where the user is in control of access and permission management of its personal medical data.

A range of studies has already been conducted on the topic of blockchain-based applications in healthcare, resulting in numerous schemes, protocols, and prototypes[1, 4, 5]. Besides these implementations, there also exists studies analyzing and comparing the different applications[6, 7]. These reviews indicate the technical challenges such as speed and scalability that blockchain applications currently encounter and review the best-proposed solutions to these problems; off-chain storage and a permissioned blockchain.

Nevertheless, research has thus far not introduced self-sovereign solutions that completely fulfill the demands of electronic health records. Most self-sovereign solutions provide the basic requirements as medical data sharing and patient-controlled data management[1, 4, 5]. Other solutions focus on specific problems in the total domain such as bidirectional data sharing[8] and emergency access[9, 10]. This thesis utilizes existing research to define and analyze protocols. These protocols extend the basic requirements of blockchain-based self-sovereign identity electronic health record solutions with the partial sharing of data, the creating of personal annotations in the electronic health records, and an emergency access protocol.

2. Related Work

The beneficial effect of health information technology on the productivity and quality of healthcare has been indicated in multiple studies [11][12]. Through the use of electronic health records(EHR), it is possible to digitally share patient information between different care providers. This easy and fast remote access can reduce paperwork and health errors. Treatment quality will improve significantly because medical professionals can review the complete medical history of a patient. Health information technology also attributes to the completeness of documentation as health care professionals contribute more detailed information and data entry is better structured[13]. Even though the system provides a great number of opportunities there are equally many challenges. The digital storing and sharing of private medical data raises security and privacy risks. Other challenges include the interoperability and transparency between different care providers.

Currently, care providers are moving from traditional client-server solutions to cloud-based solutions in order to increase worldwide accessibility and data sharing[14]. To protect the privacy of patient EHR stored in cloud-based services, patient-centric digital right management (DRM) approach has been proposed in [15]. In this approach, encryption is used by trusted DRM agents to control data and enforce policies after full or partial access is granted. In [16], a broker-based access control mechanism is used to circumvent the sharing issue of EHR in the healthcare computer-cloud environment. This mechanism facilitates an interface to connect different EMR systems(specific system of a care provider) and provide patients with local and global policies that respectively define constraints on EMR system sharing of EHR instances and centralized constraints on control over EHR records. Extra secure and robust interactions with the EHR systems can be achieved through a fail-proof two-level user authentication mechanism, which is based on biometrics and location to issue encrypted session tickets using the Kerberos protocol[17]. All these implementations do however require a trusted third party for access and permission management thereby still forming a privacy threat.

Self-sovereign identity offers a promising solution because it empowers identity owners with full control over their identities. It also improves privacy through pseudonyms and decoupling of sensitive identity records and utilizes distributed identity management therefore it is no longer depending on third parties[18]. There is no consensus on a definition of self-sovereign identity, however in an attempt to define it experts[2][19][20][21] refer to the following 10 principles from [22]:

1. Existence: *Users must have an independent existence.*
2. Control: *Users must control their identities.*
3. Access: *Users must have access to their own data.*
4. Transparency: *Systems and algorithms must be transparent.*
5. Persistence: *Identities must be long-lived.*
6. Portability: *Information and services about identity must be transportable.*
7. Interoperability *Identities should be as widely usable as possible.*
8. Consent: *User must agree tot the use of their identity.*
9. Minimalization: *Disclosure of claims must be minimized.*
10. Protection: *The rights of users must be protected.*

Blockchain intrinsically fulfills the properties, transparency, persistence, control, existence, access and consent by leveraging its underlying mechanism, decentralized ledger technology[23]. For this reason multiple self-sovereign identity frameworks based on blockchain have been developed over the last couple of years. Uport and Sovrin are two of the most complete examples of these frameworks[21]. Uport offers self-sovereign identity based on the public Ethereum blockchain and facilitates identity control via a mobile application that interacts with Ethereum's smart contracts[24]. Sorvin's self-sovereign identity is based on a public permissioned blockchain, which means that everyone can join the network, but only governed network nodes ensure consensus of the transactions on the ledger[25].

MedRec is one of the existing blockchain-based self-sovereign identity frameworks designed for healthcare[4]. MedRec logs patient-provider relationships and data access information to share access permission for health records by utilizing the public Ethereum blockchain and its decentralized computing via smart contracts. MedRec allows care providers to keep using their existing database systems to store encrypted health records off-chain. The off-chain storage access information is logged on-chain together with a hash value of the data to assure data originality. Two additional benefits of off-chain storage are scalability and deleting of records is possible, which is important in relation to the GDPR's "right to be forgotten"[6]. Different studies explore off-chain storing of medical data with public key encryption [1], Merkle Tree encryption[26] and ABE-encryption [27]. In comparison to traditional systems, blockchain-based self-sovereign identity enables distributed recording of EHR, which cannot be falsified and are unforgeable[1]. However using public blockchain as a foundation for a blockchain-based self-sovereign identity application in healthcare is criticized in [6], because of scalability issues. Permissioned blockchain is proposed as a better alternative because only authorized nodes can participate in the blockchain. A permissioned blockchain also further secures data and protects the patient's privacy better as well. EHR applications based on permissioned blockchain have been introduced that are more secure and scalable[28] and add close system monitoring[5].

While providing the important basis of patient-controlled secure storage and sharing of medical data, current studies do not completely comprise the total of electronic health record's functional requirements. In [29], functional requirements for shared electronic health records are defined based on multiple stakeholders in the system. The most important stakeholders being patients and medical professionals. The most vital functional requirement that remains untouched in the works described earlier is emergency access in situations where a patient is unable to provide consent. In [30], emergency doctors can acquire temporary access in case of emergency to medical records via smart contracts. This blockchain-based system provides patient the self-sovereign identity option of consent by defining which data is accessible and the time frame for which the data is available via the smart contracts. This study does not verify the authenticity of an emergency access request form doctor, thus exposing the security of private data. Different approaches that were investigated in non self-sovereign identity applications provide emergency access via a weighted voting among trusted parties[9] or via a smart card[10]. Also the self-sovereign identity principle of control has not fully been investigated yet, as the possibility for patients to make annotations or attribute findings to their EHR is not covered in existing studies. Bidirectional data sharing via EHRs is an example of this and posses a cost-effective and scalable solution to better health and better care in treating chronical-diseases[8].

Likewise, the self-sovereign identity property minimalization is not included in the covered studies. The minimalization property can be translated to the functional requirement of transparent data sharing, which enables patients to share parts of their medical records[29]. Minimalization can be achieved through multiple decentralized identifiers (DID), one for each relation the identity owner has[31]. Using a single DID for every relation makes it possible to minimize the data shared with that relation. The Hyperledger Indy blockchain framework, the open-source codebase of Sovrin, supports data minimalization as well. Data linked to a Sovrin identity can be shared without disclosing unnecessary information about the user[25]. Alternatively in [32], an identity system build on the Bitcoin blockchain efficiently reveals selective parts of an identity based on the Discrete Logarithm Representation (DLREP) proposed by [33]. Minimalization can also be achieved by utilizing a Merkle tree structure as the identity platform Civic does[34]. Instead of sharing the root hash of the tree and revealing all information, portions of the tree can selectively be revealed as well.

The main contribution of this study is to provide clearance in the extend to which blockchain-based self-sovereign identity can fulfill the requirements of EHRs. It will do so by extending existing blockchain-based self-sovereign identity sharing schemes with the functional features of emergency access, making of personal annotations and partial data sharing. Thereby improving on the self-sovereign identity properties control, consent and minimalization. Finally this new scheme will be compared with the properties of self-sovereign identity[22] to determine whether blockchain-based self-sovereign identity can fulfill EHR's requirements.

3. Conceptual Design

This project focuses on the application of blockchain-based self-sovereign identity in healthcare. Therefore this conceptual design extends the existing blockchain-based self-sovereign identity framework Hyperledger Indy, with EHR related functional features. Hyperlegder Indy is an open-source framework hosted by the Linux Foundation[35]. In order to understand the design of the EHR features a basic understanding of Hyperledger Indy is required. This section will first introduce the basic structure of Hyperledger Indy. Thereafter the design of the EHR features will be discussed.

3.1 Hyperlegder Indy

Hyperledger Indy hereinafter referred to as Indy, consist of three main pieces; Indy Node, Indy Plenum, and Indy SDK. Indy Node defines the available protocols for each node in the Indy blockchain network[36]. Multiple Indy nodes together form a single Indy blockchain network, where each node contains a copy of the distributed ledger. All transactions created on the network have to be authorized by each node in the network according to the Indy Plenum consensus algorithm. Indy Plenum is based on the Byzantine Fault Tolerant protocol and is designed for the special-purpose of an identity system[37]. A new transaction is added as a block to the blockchain if the majority of nodes consent to the transaction validated by the Plenum protocol. The Indy SDK(Software Development Kit) offers a C-callable library that provides an interaction interface for the Indy network[38]. Indy SDK supports wrappers in multiple programming languages(Phyton, Java, C#, etc.).

Indy is a public permissioned ledger, meaning that everyone can make use of the network, but only authorized nodes can participate in the consensus pool. Each node in the network is maintained by a Steward and only registered stewards can add new stewards(with nodes) to the network. In the Indy user hierarchy, only Trustees outrank Stewards. Trustees have the power to restart a pool and update the pool protocol. At the initial setup of the network, a Steward is predefined for each node that will host the new network. Additionally, there is the possibility to configure a Trustee for the new network as well.

Identities on the Indy network are represented by Decentralised Identifiers(DIDs). A DID contains a public key and a private verification key used to verify identities. Each DID is stored on the public ledger associated with a user-role. The Trustee and Steward role introduced before are examples of the role definitions on the ledger. Another important user-role is the Endorser role, which allows the identity owner to publish schemas and credential definitions to the ledger, and thus issue credentials. It is notable that the Trustee and Steward that can add new nodes and have other privileges have a similar DID as an Endorser or a regular identity owner, and that their DIDs are handled in the same fashion as well. While the Indy blockchain network is public to join, a DID can only be published by a trusted Steward or Trustee. This procedure makes sure that a DID owner is actually the identity owner. Typically DIDs are created and stored in wallets, which is only accessible by the identity owner. Someone is called an identity owner if he has access to the wallet containing the DID associated with that identity.

Indy uses credentials to share verifiable data between parties. A credential can only be issued by an Endorser, Steward, or Trustee and is signed with the public key of the issuer. In order to issue a credential, the issuer first generates and provides a credential offer, which contains information about the schema, the issuer, and its revocation mechanism. In order for an identity owner to obtain a credential, the identity owner has to create a credential request based on the credential offer, containing a linked secret, and a proving secret. Finally, the credential issuer combines the credential with the provided secrets and creates the credential. This credential can only be used by the identity owner that holds the linked and proving secret. This identity owner can store the credential in its wallet so that it can be used in a proof later on.

Credentials can only be issued after a credential definition has been defined. The main content of the credential definition is the schema, which defines the data format for the credential. The credential definition also holds the public key of the credential issuer. Both a schema and a credential definition can only be published to the ledger by a Trustee, Steward or Endorser. Because a credential definition is required to issue credentials, a credential can only be issued by either a Trustee, Steward or Endorser. Multiple credential definitions can be constructed from a single schema. This is very convenient in the sense that one party can define a schema and that schema can be used by different parties in their own credential definition. By means of this mechanism, different parties are no longer required to adhere to a public data format because they can use the predefined schema instead.

Credentials can eventually be used to generate proofs. Indy provides a mechanism that enables credential holders to prove certain things about their identity based on the information inside the credentials they own. Basically, a proof is a request for data. This data may be self-provided (e.g. self-signed) but it might also require the signature of the data provider. Proofs that specifically require signed data can be verified because each credential contains the public key of the credential issuer. Additionally there exist requirements for data originating from a credential based on a defined schema and for data-fields that necessarily require a specifical value or value range.

3.2 EHR features

A blockchain-based self-sovereign identity application for healthcare is constructed by utilizing multiple concepts and structures. Indy offers the basic building blocks for the blockchain-based self-sovereign identity and has been introduced in the previous subsection. Additional structures and techniques will be used to extend Indy's self-sovereign identity suitability with the needs of electronic health records. These additions enable the application to not only safely control and share data in a user-centric way but also provide doctors with safe and user-permissioned access to electronic health records in case of emergency.

The design defines four types of users: government(or other reliable) instances, doctors, patients and trusted parties. Each user is the owner of its own wallet together with its identity. A government instance user is considered an unquestionable reliable party. As a reliable party, a government instance defines the data schemas that are used by both doctors and patients to securely control and share healthcare-related data. The data schemas only specify a uniform data format that has to be maintained and therefore the government instance does not conflict with control, transparency or any of the other self-sovereign identity properties.

A doctor user could be any sort of care provider there is, for example, general practitioners, surgeons, nurses, and dentists. This group of care providers is extended even further, in this design someone is considered a doctor user if that someone creates health records for patients. Doctors using the same data schemas for health record sharing is important for both doctors and patients because it eases doctors understanding of other doctor's records, which eventually increases the quality of care as well [39]. Using the same standard is even more vital in relation to emergency access to electronic health records because records adhering to a standard are identified easier. The emergency access protocol requires the splitting of the doctors to limit the risk of malicious users to obtain access to a patients emergency health records. On top of the standard data structures and privileges, a regular doctor has, an emergency doctor holds a verifiable emergency doctor credential issued by a trustworthy issuer like a government instance.

For the scope of this design, a patient has been defined as anyone that is or might ever be in need of medical care, which can be simplified to everyone. This means that a doctor can also be a patient at times when he or she is in need of medical attention. Indy provides the basic features of user- or patient-centric control over electronic health records. Patients store the electronic health records created and shared by the care provider in their own wallet, which empowers the patients to control the (partial) sharing of their own records. On top of that patients can add self-created records to their wallet as well. These records will also be included in the patient's medical dossier as long as these records stick to the publicly published medical standard data formats. This offers patients the opportunity to contribute their findings and related data to their own medical records. In regular situations, the patient itself controls which documents are (partially) shared with care providers. However, in emergency situations where the patient is in need of care and yet is unable to approve the sharing of its data, a different protocol comes into play.

A patient should at all times be in control of the emergency access protocol in order to adhere to the self-sovereign identity properties. This can be realised by creating transparent protocols that a patient may use. The patient is able to select the electronic health records that are made available via the emergency access protocols. The selected data that can be accessed via the protocol has to be securely stored outside of the wallet since the wallet must always be kept private as it contains the user's identity. Firstly, this application defines an offline emergency access protocol that allows emergency doctors to obtain the patient's electronic health records via the patient's device holding the wallet. The second protocol allows for online retrieval of the patient's medical records via the retrieval of partial secrets from trusted parties. A patient can encrypt his selected medical records and split the decryption key into partial secrets, which the patient shares with trusted parties(family, friends, or special institutions). Both the offline and the online protocol are secured as they only allow proof validated emergency doctors to obtain the information. Neither protocol is flawless as the offline protocol requires the wallet device to be in possession of the doctor and the online protocol requires a quick response of the trusted parties. The offline and online protocols are both used to maximize the possibilities for emergency doctors the gain access to important medical records without compromising the self-sovereign identity properties.

The trusted party user defined by the application has to be regarded as an instance with the sole purpose of guarding partial secrets of patients. This trusted party is only required to do two actions. Firstly, accept partial secrets from patients and store them in their wallet. Secondly, trusted parties verify doctors with an emergency doctor proof based on a verifiable proof provided by a doctor. A legitimate verifiable proof can only be generated with a valid emergency doctor credential. After the proof has been positively validated the trusted party can share the partial secret of the patient in emergency medical need.

4. Implementation

The health record managing application is based on the foundation of Hyperledger Indy as has been indicated in the previous section. It uses the C# Indy SDK in combination with the dotnet framework to facilitate self-sovereign handling of electronic health records via a command-line interface. Furthermore, the IPFS blockchain file storage platform is used for the emergency access protocol and requires an IPFS client. This section explains the different data scheme formats, data flows and protocols used in the application based on the different types of users the application defines. The application sets up the environment as described in the rest of this section by running the `EHR environment setup` command in the CLI. The application has been implemented and tested on Ubuntu 16.04.

4.1 Government Instance

The application initializes the identity of a single government instance. This government instance defines four data schemas and thus data standards, which are publicly published as they are used throughout the entire application. As can be seen in table 4.1, a schema is published with a name, version and its attributes that define the data standard to be used. The schemas are publicly published to ensure data standardization between different parties. Each schema is published to the Indy ledger and is assigned a public identifier. Indy provides schema versioning that makes it possible to modify schemas to unpredictable future changes without majorly impacting the application as the identifier is only slightly changed based on the schemas version. Versioning is a benefit especially for the Electronic Health Record schema because of the large variety in types of care providers that create health records. Of the four defined schemas, the government instance only interacts actively with the Doctor Certificate and passively with the Emergency Trusted Parties.

The legitimacy of doctors is important to ensure the protection of the patient's medical dossier. A doctor can authorize itself as a doctor via credentials based on the Doctor Certificate schema. While the Doctor Certificate is a security-sensitive schema it is publicly published, because a schema's attributes are important for constructing a proof. The attributes of the Doctor Certificate are necessary to construct a proof that verifies whether

Name	Version	Attributes
Electronic Health Records	1.0.0	[importance_level, issuer, data_value, data_type]
Shared Secrets	1.0.0	[secret_owner, secret_issuer, secret]
Emergency Trusted Parties	1.0.0	[secret_owner, secret_issuer, min, total]
Doctor Certificate	1.0.0	[name, is_emergency_doctor, school]

Table 4.1: Data schemes defined by Gov-Health-Department

someone is actually an emergency doctor. The Doctor Certificates is designed to distinguish the emergency doctors from the regular doctor by means of the `is_emergency_doctor` attribute, which is used in the emergency doctor proof as well. Publicly publishing the schema, unfortunately, enables everyone to issue a doctor credential. Fortunately, Indy facilitates proofs that provide checks on the issuer of the credential to circumvent the disadvantage of counterfeit doctor credentials. The government instance is initialised as the only legitimate identity holder to issue doctor credentials in the application's setup environment.

The government instance uses the Emergency Trusted Parties schema in a passive sense as it does not issue credentials based on this schema. The government instance actively stores credentials based on this schema in its wallet to maintain an administration of shared secrets. This administration registry is intended as a registry that emergency doctors can consult to learn who the trusted parties of a patient are. This administration registry can be extended to multiple government instances to prevent downtime. The credential contains the name of the shared secret holder together with the number of partial secrets that exist and the number of partial secrets necessary to reconstruct the secret. Only emergency doctors can require the information as a government instance always validates a requested emergency doctor proof before sharing any patients' credentials.

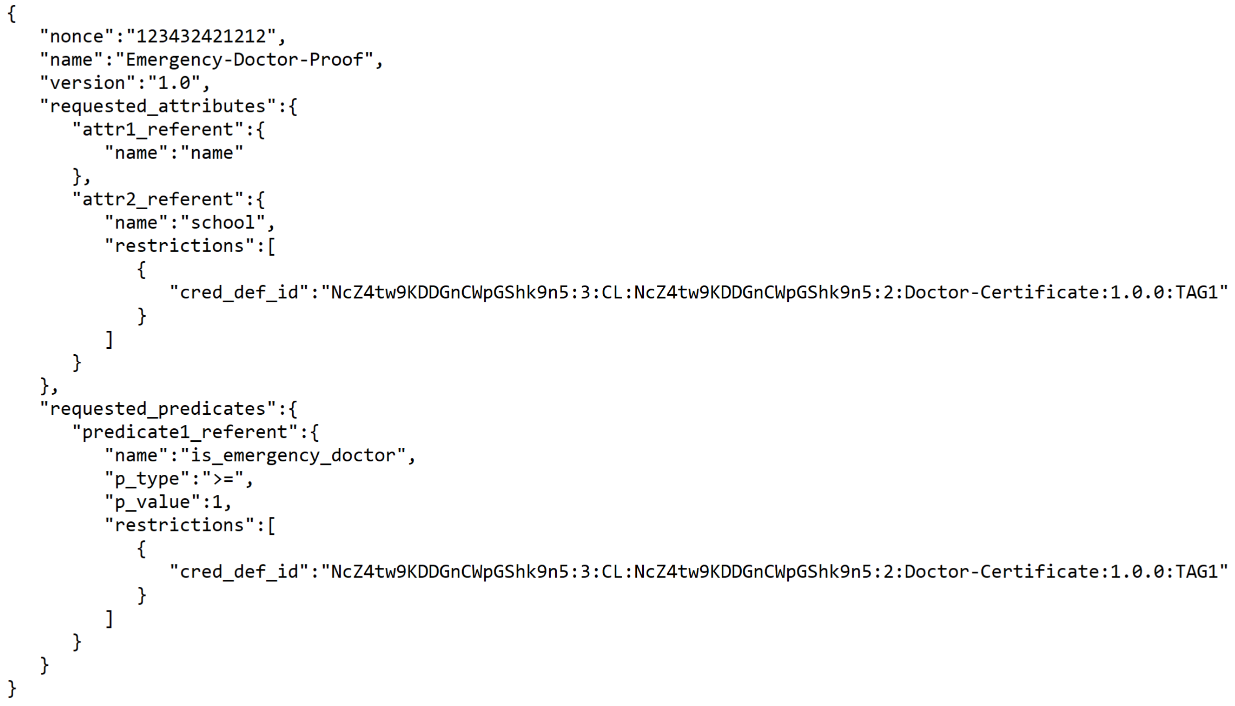
4.2 Doctor

Three complete doctor identities are created via the setup command of the application. Doctors are separated from regular users via doctor credentials that are issued only by the government instance. Doctors are in their turn split again in the regular doctors and the emergency doctors based on the `is_emergency_doctor` attribute within the doctor credential. The main purpose of the doctor user is to create an electronic health record after providing care, doing research, having finished a consultation or completing any other task that produces an electronic health record. These records are shared with patients in the form of an issued credential that is based on the Electronic Health Records schema as defined by the government instance.

Every doctor possesses a credential definition allowing the doctor to issue an electronic health record credential. After treatment, the doctor generates a credential offer for the patient which the patient uses to create a credential request. The doctor then generates a credential using the patient's credential request, thereby integrating the identity of both the doctor and the patient in the credential signature. Finally, the patient can store the issued credential in its wallet. Since the stored credential is based on the Electronic Health Records schema of the government department the credential is automatically affiliated with the medical dossier.

Doctor Proof

Indy credentials contain data signed by the issuer of the credential. This allows the credential owner to prove the validity of the data and of the owner itself. The doctor proof is founded on this mechanism. A doctor credential's validity is guaranteed via the signed



```
{
   "nonce":"123432421212",
   "name":"Emergency-Doctor-Proof",
   "version":"1.0",
   "requested_attributes":{
      "attr1_referent":{
         "name":"name"
      },
      "attr2_referent":{
         "name":"school",
         "restrictions":[
            {
               "cred_def_id":"NcZ4tw9KDDGnCWpGShk9n5:3:CL:NcZ4tw9KDDGnCWpGShk9n5:2:Doctor-Certificate:1.0.0:TAG1"
            }
         ]
      }
   },
   "requested_predicates":{
      "predicate1_referent":{
         "name":"is_emergency_doctor",
         "p_type":">=",
         "p_value":1,
         "restrictions":[
            {
               "cred_def_id":"NcZ4tw9KDDGnCWpGShk9n5:3:CL:NcZ4tw9KDDGnCWpGShk9n5:2:Doctor-Certificate:1.0.0:TAG1"
            }
         ]
      }
   }
}
```

Figure 4.1: Indy proof definition of the emergency doctor proof

data inside the credential. The doctor proof requires one self-provided unsigned attribute that specifies the name of the doctor, one signed attribute specifying the university the doctor graduated from and one signed predicate stating whether the doctor is an emergency doctor. This emergency predicate must equal the value true(1) in order for the prover to be verified legitimate. A comparison between the prover's data and the trusted issuer's credential definition identifier validates the proof. For this reason, only the indisputably trustable government instance is initialized as a valid doctor credential supplier.

4.3 Patient

A patient is a user that uses all the publicly available government schemas because of the variety of functionality at the patient's disposal. The functionalities include sharing of personal data from the wallet, collecting the electronic health records into one medical dossier, creating personal annotations, exporting the medical dossier for emergency access purposes, and providing two secure medical dossier access mechanisms for doctors in emergency situations. This section describes the implementation of these functionalities.

Sharing Data

Indy offers two ways of sharing data, credential issuing, and proof generation. A third manner for sharing data would be sharing of plain self-selected data without signing that data in the form of an issued credential. Credentials can be issued by the patient according to any defined schema to share any self-selected data that fits the schema's definition. Patients sign the credential's data with one of their (public) identifiers that makes the signer the provable issuer of that credential. Sharing data via a proof is different as the proof's definition specifies the data, has to be provided and the requirements it has to meet. Indy facilitates methods to automatically search the wallet's credentials for the required data fields. The right field can be selected or filled in by the patient in the situation there are no requirements defined for the data field. The patient can always define the credentials that are partially or fully shared.

Medical Dossier

The medical dossier consists of credentials in the patient's wallet that are based on the Electronic Health Records schema. The credentials for the medical dossier are retrieved by querying the wallet for the credentials containing the schema identifier of the Electronic Health Records schema. Indy facilitates the wallet query language, which is specifically designed for this purpose. All electronic health record credentials issued by doctors need to be based on the Electronic Health Records schema because the credential is otherwise not recognized as part of the medical dossier. Personal annotations have to be created with credentials based on the Electronic Health Records schema as well, as the medical dossier only includes these credentials. Therefore personal annotations are self-issued credentials stored in the patient's wallet. Since a patient is not registered as a doctor, a patient signed electronic health record credential does not hold any medical substantiated data. Seeing that a patient is likewise a doctor possible to create the electronic health record credentials and a doctor can become the patient in specific situations. A doctor could be constructed as a patient that among the standard patient credentials also contains a certified doctor credential.

Normally the medical dossier can only be extracted from the password-protected wallet by the wallet owner. This guarantees that medical records and other private records are controlled and shared by the patient itself conform to the self-sovereign identity property consent. However, situations exist where the patient is in desperate need of medical attention but unable to provide access to its medical dossier. Both patients and doctors can benefit from the emergency access protocol that allows doctors to access the patient's emergency medical dossier.

The emergency medical dossier is a copy of the medical dossier in the wallet exported to a file in the cloud. The export file is publicly available so that it can be accessed at any point in time by anyone. To maintain the application's independence of third parties the InterPlanetary File System(IPFS) is used as a cloud storage provider. IPFS is a peer-to-peer network that facilitates among other features a peer-to-peer cloud storage network[40]. Because the emergency medical dossier is publicly available, it is encrypted with symmetric key encryption to secure the exported data. Van Rijndael encryption is the symmetric key encryption algorithm used to encrypt the medical dossier[41]. The encryption uses a 256-byte key and initialization vector to encrypt the medical dossier. The emergency medical dossier is retrievable only when the public IPFS file path to the encrypted file is known together with the encryption cypher key and initialization vector. Doctors can take advantage of two related protocols to acquire access to this information, an offline and an online protocol.

Offline Emergency Access

The offline emergency access protocol is based on the idea of emergency access via a smart card that contains a hash to access the medical dossier[10]. Instead of a smart card, the wallet device holds an offline emergency protocol that doctors can use to obtain the emergency access information of the patient. The emergency access information used for the offline emergency protocol is stored on the device outside the wallet because the wallet is password protected and should only be accessible by the patient. The offline emergency protocol protects the emergency access information from malicious intruders by a doctor proof so that only verified emergency doctors can acquire the emergency access information. The offline emergency protocol is a totally standalone process because the Indy SDK can process the proof without an active wallet or an active connection to a pool.

Online Emergency Access

The online emergency access protocol obtains the patient's emergency access information from the patient's network of trusted parties. The emergency access information is encrypted with Shamir's Secret Sharing which splits the encrypted information into multiple keys[42]. Shamir's secret sharing is based on a polynomial interpolation (k,n) threshold scheme that encrypts a piece of data into n different keys. The encrypted data can be recovered by providing at least k keys, where $k \geq 3$ and $k \leq n$. A patient can use Shamir's secret sharing to encrypt the emergency access information into secret keys and can define k and n itself as long as the requirements on k and n are met.

The created keys are stored in wallet records inside the patient's wallet. Each record contains a key and a specifier marking the key as shared or not yet shared. The records feature an administrative view for the patient regarding its shared and unshared keys. The patient controls the distribution of the keys via credential issuing and the information in the administrative records. The shared secret schema forms the foundation of the issued credential to maintain a standard for emergency keys throughout the system. It is important that at least k keys are distributed right after the creation of the keys to guarantee a reconstruction possibility of the emergency access information via the online protocol. Patients are encouraged to use a value for n that is large than k as emergency access is time-sensitive and more keys increase the chance on a quick reconstruction. Of course, larger n increases the chance of a conspiracy among trusted parties to obtain the reconstruction information as well.

Emergency Doctors face the challenge of collecting k patient emergency access keys in order to reconstruct the emergency access information. To ease their search an additional administration is created based on the Emergency Trusted Parties schema of the government instance. The credentials based on the Emergency Trusted Parties schema specify a trusted party for a patient and are created and issued by the patient itself. These credentials are only shared with reliable instances to maintain privacy and prevent malicious use.

Preferably patients share the emergency trusted party credentials with multiple reliable instances to instantiate a backup lookup via multiple administrations. Reliable instances storing these credentials have to be designated so that emergency doctors know where to obtain information on a patient's trusted parties. In an emergency situation a list of trusted parties can be obtained at the designated administrative instances by a valid emergency doctor only. The emergency doctor can request the patient's key from all trusted parties and use the first k responses to reconstruct the emergency access information. Each request to a trusted party is accompanied by a doctor proof for the trusted party to verify the validity of the request.

5. Results

This section judges the discussed protocols and structures on their self-sovereign identity properties. It will start by evaluating Indy's self-sovereign identity fitness because all protocols are based on the foundation of Indy. Thereafter the implemented protocols will be considered for each of Allen's ten self-sovereign identity principles. Both results are combined in the end to state the overall self-sovereignty of the protocols and the application.

5.1 Hyperledger Indy

Hyperledger Indy is an open-source blockchain network specifically designed for the purpose of digital identity. Indy provides the code base for the public self-sovereign identity blockchain Sovrin[25]. In this subsection, the self-sovereign identity of Indy will be investigated via Sovrin because multiple studies independently analyse Sovrin's self-sovereign identity features[21, 43]. Moreover, Sovrin itself provides extensive documentation and analysis on their own identity framework as well[44].

Sovrin's foundation rests on the blockchain network of Hyperledger Indy. Blockchain technology intrinsically guarantees the self-sovereign identity properties existence, control, access, transparency, persistence, and consent. Therefore Indy too fulfils these properties. Existence is guaranteed through a unique DID for each user. The user controls the identity as the user is the only person with access to the private key associated with the DID. The data stored on the blockchain is publicly available and private data is stored in the user's wallet. The blockchain foundation guarantees transparent data storage and handling using Indy's Plenum protocol. The persistence property is achieved through the immutability of the blockchain. Consent is closely related to control and the consent property is achieved seeing that Sovrin identities are maintainable only by the user, thus providing the user's consent. Table 5.1 indicates the unanimous agreement of the separate studies on these properties. The protection property is also unanimously agreed on because Sovrin puts the user in control of protecting its own rights. Sovrin is designed according to the privacy rights of the user and merely provides a platform to exchange information with the consent of the user only.

The self-sovereign identity properties which are not unanimously agreed on include minimalization, interoperability, and portability. The disagreement can be explained by the lack of information and the continuous improvement of the relatively new Hyperledger Indy framework. The credential and proof structure designed within Indy enables users to select those attributes of the credentials they own that they wish to share with a different party. These mechanisms facilitate minimalization on Indy-based applications. A Sovrin can be used in various credential interchanges to facilitate interoperability. On top of that Sovrin uses the open standards of Indy that uses Decentralised Identifiers as defined to guarantee portability as well[45]. Concluding the above Sovrin and its foundation Indy can certainly be classified as self-sovereign identity frameworks.

5.2 EHR features

The list of self-sovereign identity properties as defined by C. Allen is used to analyse the self-sovereignty of the designed features[22]. Though not all properties are related to the designed features they are included to complete the analysis. The analysis mainly focuses on the emergency access protocol as the creation of personal annotations and particularly partial sharing of data are intrinsic parts of Indy.

1. Existence: ✔ *Users must have an independent existence.* Indy facilitates a single wallet per user that holds the DID associated with someone's identity. The procedures utilize this principle but do not alter it thereby satisfying the existence property.

2. Control: ✔ *Users must control their identities.* Users control all data exchanges made on behalf of their identity. On top of that, they control all data created with their identity. This translates into the three procedures as well. The user is in control of sharing data with different parties and determines which and how much data is shared. The user can independently create annotations that are included in the medical dossier. In the emergency access process, the user is solely responsible for activating the procedures. Furthermore, in the online emergency access protocol, the user determines the number of secret keys necessary to reconstruct the emergency access information and selects the trusted parties the keys are shared with.

3. Access: ✔ *Users must have access to their own data.* The procedures store user-related data always in the corresponding user's wallet. Indy's credential structure is used to create and issue data to the related user. The user stores these credentials in its wallet thereby making the data always accessible to the user. Credentials are created for

Document	[21]	[43]	[44]
Existence	✔	✔	✔
Control	✔	✔	✔
Access	✔	✔	✔
Transparency	✔	✔	✔
Persistence	✔	✔	✔
Portability	?	?	✔
Interoperability	?	✔	✔
Consent	✔	✔	✔
Minimalization	?	✔	✔
Protection	✔	✔	✔

Table 5.1: Self-sovereignity of Sovrin. A ✔ denotes the presence of an property. A **?** indicates a lack of information.

data sharing, the creation personal annotations, and the emergency access protocol. Though a subtle difference exists. While with data sharing and the emergency access protocol the user issues credentials to a different related-user, with creating personal annotations a credential is self-issued and stored in the own wallet.

4. Transparency: ✔ *Systems and algorithms must be transparent.* The procedures are well documented and publicly published.

5. Persistence: ✔ *Identities must be long-lived.* Indy's identities are represented by DIDs that are published to the blockchain. By design, the blockchain is immutable, which results in long-lived DIDs thus identities.

6. Portability: ✔ *Information and services about identity must be transportable.* Indy's identities are not reliant on a single party, because they are identified by the pool of Indy nodes. The protocols build on this foundation and avoid mechanisms that would prevent identity owners from portability.

7. Interoperability ✔ *Identities should be as widely usable as possible.* The protocols are specifically designed for medical purposes, but they do not oppose Indy's overall identity. The setup identities can be expanded to include passports, driving licenses, and all kinds of other contracts. Alternatively, the protocols could be implemented on a different self-sovereign identity system.

8. Consent: ✔ *User must agree to the use of their identity.* Consent is closely related to the control property. As the protocols are designed according to Indy's foundation, they impose user-centered control. Users control a major part of their identity and control the use of their identity thus agreeing to the use of their identity. The emergency protocol is invoked to grant access to an identity without the direct consent of the user. This protocol may only be used in a situation where the user is unable to provide consent and access to the user's identity is of vital interest. The emergency access protocol grants passive consent to the use of an identity and always has to be initialized by the user. The emergency access protocol can be set completely to the user's preference. The protocols are only usable with the user's consent and only able to use an identity with the user's passive or active consent.

9. Minimalization: ✔ *Disclosure of claims must be minimized.* Indy's proof design facilitates minimalization by constructing proofs using only the required data elements of available credentials. The doctor proof is an example of such a proof. Most of the data handled by the procedures is shared through the issuing of credentials. In that situation, the user defines the data shared and is responsible for minimalization. Of course, the data formats are standardized and designed according to the minimalization property.

10. Protection: ✔ *The rights of users must be protected.* Indy's structure of identity roles guarantees the standards and rights in the system. The rights for the protocols are founded on this structure and utilize Indy's credentials and proofs to authorize

some users with more rights. The authorizers of these credentials are indisputable trusted parties, which guarantees the protection of users' rights. Due to the blockchain foundation of Indy, there is not a single user capable of disrupting the protocols.

All in all the designed application adheres to the self-sovereign identity properties and is based on blockchain technology. The designed protocols adhere to the 10 properties of self-sovereign identity. The protocols are strongly connected to the Indy foundation they were constructed upon. Indy is a complete self-sovereign identity framework and therefore offers a solid foundation for the protocols. While the protocols can not be regarded without the design of the self-sovereign identity foundation, any self-sovereign identity framework could be used as the foundation for the protocols.

6. Conclusion

The goal of this bachelor thesis is to research the potential of blockchain-based self-sovereign identity in electronic health records. Existing research into this topic is extended with protocols that allow partial data sharing, creating personal annotations and emergency access. The emergency access protocol is especially important as access to medical data in emergency situations is of vital importance but does not cope well with privacy and the patient's consent.

To achieve this goal the blockchain-based self-sovereign identity framework Hyperledger Indy has been used as the foundation of the protocols. Indy's identity management handles data interactions using a wallet. A medical dossier containing only electronic health records is defined inside this wallet as a basis for the protocols. The medical dossier definition inside the wallet enables the use of Indy's basic features of data creation and sharing for the protocols. The partial sharing of medical data is handled with Indy's capable data sharing facilities. Also, the creation of personal annotations is founded on Indy's verifiable credential schema in combination with the medical dossier definitions. However, the emergency access protocol required extra capabilities on top of Indy.

The patient's consent is the main difficulty for the emergency access protocol as in an emergency situation patients are often unable to provide consent. The emergency protocol is designed to circumvent this issue by providing passive consent in the form of predefined emergency access information. This emergency access information can either be set in a local file on the patient's wallet device or be encrypted into multiple keys using Shamir's Secret Sharing and are held by trusted parties. Both access protocols have to be set up by the patient so that the patient actively consents to both passive consent protocols.

The C# command line interface is a proof of concept implementation and focuses on protocol instead of user experience. To successfully integrate the protocols in real-life electronic health record applications requires adaptation to a user-friendly environment for reliable parties, doctors, and especially patients. Examples already exist where smartphones are used as edge devices and a mobile application is used to interact with blockchain-based self-sovereign identity service providers.

All in all the protocols researched in this thesis offer suitable solutions to the electronic health record problems of partial sharing, creating personal annotations and emergency access. Especially the emergency access protocol can easily be adapted by different electronic health record frameworks to ensure patient consent in emergency situations.

7. Future Work

- **Temporary Emergency Access**: At this point, the emergency access protocol provides information to access the emergency medical dossier and to decypher the encrypted contents. The process is designed so that doctors can use a patient's medical dossier while the patient remains in a critical condition. The patient can replace the emergency medical dossier with a newly encrypted one so that the access information obtained by the doctor is invalidated. However, this does not affect the medical dossier the doctor downloaded in the emergency situation. Time-based re-encryption can be used to resolve this issue[46]. Another solution includes revocable access rights to the emergency medical dossier in the cloud[47]. It is important that doctors may never be interrupted by any of these security measures regardless of the implemented solution.

- **Mobile Application**: A mobile application that implements the researched protocols can improve the user experience and decrease human errors. Indy's SDK supports iOS development for Apple devices and multiple other languages that can be used to develop applications for Android devices[38].

  On top of the user experience, a mobile application can also majorly improve the offline emergency access protocol. Doctors can only use the emergency access protocol if they have physical access to the patient's wallet holding device. The chance of acquiring the patient's smartphone is a lot bigger than the patient's laptop or PC. Furthermore, both Apple and Android devices come with a medical ID. A medical ID is available without unlocking a smartphone and contains important medical information. The medical ID is an ideal location for the offline emergency information, which should of course only be accessible for emergency doctors.

- **Multiple Cloud Providers**: IPFS is the only available cloud storage provider for the emergency medical dossier at the moment. While IPFS is perfectly safe and sound, including multiple cloud providers improves the interoperability of the online emergency access protocol's implementation. Moreover, the application would also lose its dependency on an IPFS client.

References

[1] Yi Chen et al. "Blockchain-Based Medical Records Secure Storage and Medical Service Framework". In: *Journal of Medical Systems* (2019).

[2] Jeroen Schouten. "OPPORTUNITIES FOR BLOCKCHAIN-BASED IDENTITY IN HEALTHCARE". In: (2017). URL: https://www.ru.nl/publish/pages/813276/schouten_jeroen_-1a.pdf (visited on 09/04/2020).

[3] Leslie Beard et al. "The challenges in making electronic health recordsaccessible to patients". In: *Journal of the American Medical Informatics Association, Volume 19, Issue 1, January 2012, Pages 116–120* (2012).

[4] Asaph Azaria et al. "MedRec: Using Blockchain for Medical Data Access and Permission Management". In: *2016 2nd International Conference on Open and Big Data (OBD)* (2016).

[5] Qi Xia et al. "BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments". In: *Information v8 n2 (20170417): 44* (2017).

[6] Cornelius C. Agbo, Qusay H. Mahmoud, and J. Mikael Eklund. "Blockchain Technology in Healthcare: A Systematic Review". In: *Healthcare v7 n2 (2019)* (2019).

[7] Suveen Angraal, Harlan M. Krumholz, and Wade L. Schulz. "Blockchain Technology Applications in Health Care". In: *Circ. Cardiovasc. Qual. Outcomes 2017, 10, e003800* (2017).

[8] Malinda M. Peeples, Anand K. Iyer, and Joshua L. Cohen. "Integration of a Mobile-Integrated Therapy with Electronic Health Records: Lessons Learned". In: (2013). URL: https://journals.sagepub.com/doi/pdf/10.1177/193229681300700304 (visited on 05/03/2020).

[9] Tingting Chen and Sheng Zhong. "Emergency Access Authorization for Personally Controlled Online Health Care Data". In: *Journal of Medical Systems 36, 291-300 (2012)* (2012).

[10] Yu-Yi Chen, Chuan-Chiang Huang, and Jinn-Ke Jan. "The Design of AATIS Emergency Access Authorization for Personally Controlled Online Health Records". In: *Journal of Medical and Biological Engineering v35 n6 (201512): 765-774* (2015).

[11] M. Z. Hydari, R. Telang, and W. M. Marella. "Saving Patient Ryan - Can Advanced Electronic Medical Records Make Patient Care Safer?" In: *Manage Sci Articles in Advance:1–19, 2018* (2018).

[12] Jaymeen R. Shah, Mirza B. Murtaza, and Emmanuel Opara. “Electronic Health Records: Challenges and Opportunities”. In: (). URL: https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1082&context=jitim (visited on 04/24/2020).

[13] K Häyrinen. “Definition, structure, content, use and impacts of electronic health records: a review of the research literature”. In: *International Journal of Medical Informatics, Volume:77, Issue:5, Page(s):291* (2008).

[14] R. S. Evans. “Electronic Health Records: Then, Now, and in the Future”. In: *Yearb Med Inform 2016; 25(S 01): S48-S61* (2016).

[15] M. Jafari, R. Safavi-Naini, and N. Sheppard. “A rights management approach to protection of privacy in a cloud of electronic health records”. In: *Proceedings of the 11th annual ACM workshop on Digital rights management, pages 23–30. ACM* (2011).

[16] Ruoyu Wu, Gail-Joon Ahn, and Hongxin Hu. “Secure Sharing of Electronic Health Records in Clouds”. In: *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)* (2012).

[17] Uthpala Subodhani Premarathne et al. “Hybrid Cryptographic Access Control for Cloudbased Electronic Health Records Systems”. In: *IEEE CLOUD COMPUTING* (2017). URL: https://pdfs.semanticscholar.org/cef3/13ff1fd55fb0ffc6e5e838e7392153495fcb.pdf (visited on 04/24/2020).

[18] Geovane Fedrecheski et al. “Self-Sovereign Identity for IoT environments:A Perspective”. In: *arXiv:2003.05106* (Mar. 2020). URL: https://arxiv.org/pdf/2003.05106.pdf (visited on 04/30/2020).

[19] Toth KC. and Anderson-Priddy A. “Self-Sovereign Digital Identity: A Paradigm Shift for Identity”. In: *IEEE Security and Privacy v17 n3 (2019 05 01): 17-27* (2019).

[20] van Bokkem D. et al. “Self-Sovereign Identity Solutions: The Necessity of Blockchain Technology”. In: *arXiv preprint arXiv:1904.12816. 2019 Apr 29.* ().

[21] Farida Chowdhury MD Sadek Ferdous and Madini O. Alassafi. “In Search of Self-Sovereign Identity Leveraging Blockchain Technology”. In: *IEEE Access, Jaargang:7, Pagina(s):103059-103079* (2019).

[22] Allen C. “The Path to Self-Sovereign Identity”. In: *Life With Alacrity* (Apr. 2016).

[23] Quinten Stokkink and Johan Pouwelse. “Deployment of a Blockchain-Based Self-Sovereign Identity”. In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (2018).

[24] C. Lundkvist et al. “UPORT: A Platform For Self-Sovereign Identity”. In: (Oct. 2016). URL: https://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf (visited on 04/15/2020).

[25] Phillip J. Windley. “How Sovrin Works”. In: *White Paper Sovrin Foundation* (Oct. 2016). URL: `https://sovrin.org/wp-content/uploads/2018/03/How-Sovrin-Works.pdf` (visited on 04/16/2020).

[26] Ashutosh Dhar Dwivedi et al. “A Decentralized Privacy-Preserving Healthcare Blockchain for IoT”. In: *Sensors 19(2):326* (Jan. 2019).

[27] Hao Guo et al. “Attribute-based Multi-Signature and Encryption for EHR Management: A Blockchain-based Solution”. In: *arXiv:2002.11078* (Feb. 2020). URL: `https://arxiv.org/pdf/2002.11078.pdf` (visited on 04/29/2020).

[28] Zhe Xiao et al. “EMRShare: A Cross-Organizational Medical Data Sharing and Management Framework Using Permissioned Blockchain”. In: *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)* (2019).

[29] Thomas Schabetsberger et al. “What are Functional Requirements of Future Shared Electronic Health Records?” In: *ReserachGate* (2005).

[30] Ahmed Raze Rajput et al. “EACMS: Emergency Access Control Management System for Personal Health Record Based on Blockchain”. In: *IEEE Access (Volume: 7)* (2019).

[31] Mehmet Aydar and Serkan Ayvaz. “Towards a blockchain based digital identity verification,record attestation and record sharing system”. In: (June 2019). URL: `https://arxiv.org/pdf/1906.09791.pdf` (visited on 05/06/2020).

[32] Daniel Augot et al. “A User-centric System for Verified Identities onthe Bitcoin Blockchain”. In: *Springer International Publishing, Cham, 2017.* (2017). URL: `https://doi.org/10.1007/978-3-319-67816-0_22` (visited on 05/06/2020).

[33] S.A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building inPrivacy.* 2000.

[34] “Civic”. In: *Whitepaper Civic Technologies, Inc.* (2017).

[35] Linux Foundation. *Hyperledger Indy.* URL: `https://www.hyperledger.org/use/hyperledger-indy` (visited on 06/15/2020).

[36] *Indy Node.* URL: `https://github.com/hyperledger/indy-node` (visited on 06/15/2020).

[37] *Indy Plenum.* URL: `https://github.com/hyperledger/indy-plenum` (visited on 06/15/2020).

[38] *Indy SDK.* URL: `https://github.com/hyperledger/indy-sdk` (visited on 06/15/2020).

[39] PhD; Walter Sujansky MD and MPH Sophia Chang MD. “The California Clinical DataProject: A Case Study in the Adoption of Clinical Data Standards for Quality Improvement”. In: *Journal of Healthcare Information Management — Vol. 20, No. 3* (2006). URL: `https://www.researchgate.net/profile/Nir_Menachemi/publication/6883096_EHR_and_other_IT_adoption_among_physicians_results_of_a_large-scale_statewide_analysis/links/02bfe5107eec9400cf000000/EHR-and-other-IT-adoption-among-physicians-results-of-a-large-scale-statewide-analysis.pdf#page=73` (visited on 06/22/2020).

[40] Sebastian Henningsen et al. "Mapping the Interplanetary Filesystem". In: (Feb. 2020). URL: `https://arxiv.org/pdf/2002.07747.pdf` (visited on 07/04/2020).

[41] Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Springer, Berlin, Heidelberg, 2002.

[42] Adi Shamir. "How to Share a Secret". In: *Communications of the ACM Volume 22 Number 11* (1979). URL: `https://cs.jhu.edu/~sdoshi/crypto/papers/shamirturing.pdf` (visited on 06/26/2020).

[43] Paul Dunphy and Fabien A. P. Petitcolas. "A First Look at Identity Management Schemes on the Blockchain". In: *IEEE Security and Privacy Magazine special issue on 'Blockchain Security and Privacy' 2018* (2018).

[44] Sovrin Foundation. "Sovrin Governance Framework V2 Master Document V2". In: (Dec. 2019). URL: `https://sovrin.org/wp-content/uploads/Sovrin-Governance-Framework-V2-Master-Document-V2.pdf`.

[45] Drummond Reed et al. "Decentralized Identifiers (DIDs) v1.0". In: (July 2020). URL: `https://w3c.github.io/did-core/` (visited on 07/08/2020).

[46] Qin Liu, Guojun Wang, and Jie Wu. "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment". In: *Information Sciences Volume 258* (Feb. 2014).

[47] Kaitai Liang et al. "An Efficient Cloud-Based Revocable Identity-Based Proxy Re-encryption Scheme for Public Clouds Data Sharing". In: *Computer Security - ESORICS 2014* (2014).