Applying Blockchain Technology to Security-Related Aspects of Electronic Healthcare Record Infrastructure

Ryno Adlam
Master of Technology student, School of Information Technology, Nelson Mandela University, Port Elizabeth, South Africa
https://orcid.org/0000-0002-6514-3673

Bertram Haskins
Associate Professor, School of Information Technology, Nelson Mandela University, Port Elizabeth, South Africa
https://orcid.org/0000-0002-9762-7381

Abstract
The centralised architecture employed by electronic health records (EHRs) may constitute a single point of failure. From the perspective of availability, an alternative cloud-based EHR infrastructure is effective and efficient. However, this increased availability has created challenges related to the security and privacy of patients' medical records. The sensitive nature of EHRs attracts the attention of cyber-criminals. There has been a rise in the number of data breaches related to EHRs. The infrastructure used by EHRs does not assure the privacy and security of patients' medical records. Features of blockchain platforms, such as decentralisation, immutability, auditability, and transparency, may provide a viable means of augmenting or improving services related to the security of EHRs. This study presents a series of experimental data flow configurations to test the application of blockchain technology to aspects of EHRs. The insights gained from these experiments are founded on a theoretical base to provide recommendations for applying blockchain technology to services related to the security of EHR infrastructure. These recommendations may be employed by developers when redesigning existing EHR systems or deploying new EHR systems.

Keywords
healthcare, electronic health records (EHRs), blockchain, information security

DOI: https://doi.org/10.23962/10539/32211

Recommended citation
Adlam, R., & Haskins, B. (2021). Applying blockchain technology to key aspects of electronic healthcare record infrastructure. *The African Journal of Information and Communication (AJIC), 28*, 1-28. https://doi.org/10.23962/10539/32211

1. Introduction

An electronic health record (EHR) is the electronic equivalent of the medical history of a specific patient. It presents potential benefits, such as a reduction of errors, an increase in the availability of medical records, and, as a knock-on effect, an improvement in the quality of patient care (Thakkar & Davis, 2006). However, EHR systems may encounter several challenges in the form of data breaches, privacy compromises, interoperability, auditability, and fraud. EHR systems currently utilise a centralised architecture that requires a centralised authority of trust and leaves medical records vulnerable due to a single point of failure (Liang et al., 2017). Highly sensitive patient-related information is associated with an EHR, including information such as patient demographic details, medical history, and data points related to patient vital signs (Menachemi & Collum, 2011). This wealth of information makes EHRs lucrative targets for cybercriminals and, as a result, the number and severity of successful cyberattacks on EHRs are increasing (Ronquillo et al., 2018). The conventional model employed by EHR systems can no longer ensure the security and privacy of patient health records (Kshetri & Carolina, 2018). The privacy and security of EHRs may be improved by the desirable features of blockchain technology such as decentralisation, immutability, auditability, and transparency (Emmadi et al., 2019).

To improve or augment the services related to the security of electronic healthcare infrastructure, developers may turn to blockchain technologies, but may be unfamiliar with how or where to apply them to the EHR infrastructure. Therefore, the objective of this study is to present recommendations for applying blockchain technology to services related to the security of the electronic healthcare record infrastructure. The remainder of this article is structured as follows: section 2 presents an overview of key background concepts; section 3 discusses how aspects of blockchain technology may be applied to EHRs; and section 4 provides an overview of the workflows generated from experimenting with blockchain technologies. Insights gained from the experiments and theory are presented as a set of recommendations in section 5, and the study is concluded in section 6.

2. Background

EHRs are widely used to maintain patient data in an online format. Blockchain technology is a means of storing information in a distributed fashion. To understand how these concepts could intersect, this section provides an overview of the respective technologies, their component aspects, and examples of their use.

Electronic health records (EHRs)

The healthcare industry is continually evolving. The evolution towards EHRs from a paper-based system has been fuelled by new advancements in the realm of information technology (Seol et al, 2018). The EHR is the electronic equivalent of a patient's full medical record, promising benefits such as the improved sharing of information, saving time for medical professionals, a cost reduction, reducing the number of

errors, and a general improvement in the quality of patient care (Dekker & Etalle, 2007; Thakkar & Davis, 2006). EHR systems are subject to privacy regulations as they deal with patient information such as a patient's medical history, vital signs, and demographic information, all of which are considered sensitive. As a result of this highly sensitive information contained in EHRs, they face constant cyberattacks, and the number of these attacks are on the rise (Kshetri & Carolina, 2018). In 2015, more than 112 million records were exposed through data breaches (Kshetri & Carolina, 2018; Ronquillo et al., 2018).

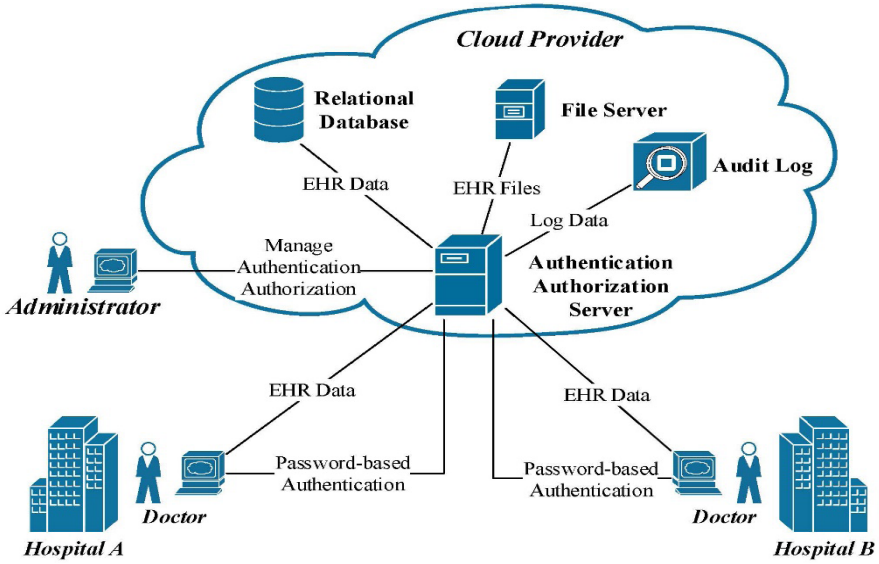
EHRs are soft targets for those with nefarious purposes because they may not be as well protected, but contain a wealth of personal information. The stolen data is either sold on the black market or the hackers hold the EHRs for ransom. The WannaCry ransomware cyberattack in 2017 affected countless healthcare providers who were forced to either pay the ransom or close their doors to further patient care (Ronquillo et al., 2018). Therefore, it stands to reason that the privacy and overall security of a patient's EHR cannot be adequately ensured by the traditional centralised information storage and transport architecture (Kshetri & Carolina, 2018).

Relational databases, which are an example of a centralised client-server, multi-user architecture, are frequently used to store patient EHRs (Griggs et al., 2018). Although a client-server-based model ensures that all clients have access to a centralised store of information, this model does run the risk of clients losing access to the information if the server is unavailable, resulting in a single point of failure (Liang et al., 2017). A modern version of the client-server model is that of cloud services or cloud computing. This model allows client devices to access a remote virtualised server via an internet connection. A virtual server provides benefits such as improved scalability and flexibility, greater availability, and a reduction in overall operational costs (Ziglari & Negini, 2017). Such an always-on and accessible solution greatly improves the efficiency and availability of an EHR solution, but does raise further concerns with regard to privacy and security. The improvement of accessibility of the EHRs not only holds true for those who may legally access them, but also for those with harmful intent.

An overview of the current EHR model is illustrated in Figure 1. The model provides an overview of the various types of activities undertaken and the actors involved in the traditional model, as well as how the data flows between these activities and actors. By studying these details, it is possible to identify a few high-level processes used to support the current client-server EHR system implementation. These processes are the transport of data, authentication, authorisation, and auditing. The transport of data is very dependent on the underlying hardware, the communication protocols in use on the hardware, and the connections established by the various operating systems and related software, such as relational databases and client-based software. Adding security may be done on the level of the lower-level transport stream or be built into the database and client connection, but is largely left to the EHR

developer/infrastructure creator's discretion. The sub-sections that follow delve deeper into the concepts of authentication, authorisation, and auditing.

Figure 1: EHR system overview diagram (Adlam & Haskins, 2019



Authentication

Authentication refers to the process of identifying which user is requesting access to the system, so as restrict access to that system's functions. Users' identities are also required for audit purposes (Cilliers, 2017). As with many centralised systems, EHR systems make use of password-based authentication (Kshetri & Carolina, 2018). Although automated password generation, effective organisational password policies, and the application of multi-factor authentication can largely mitigate the risks of password-based authentication, these measures are not universally in place. In most instances, the passwords are manually created by humans and are considered to be weak and easily cracked by utilising techniques such as social engineering, password guessing, and brute-forcing (Kshetri, 2017). Passwords are commonly stored in a centralised relational database, which may represent a single point of failure if the passwords are not hashed to avoid an attacker retrieving them. When an authentication database is compromised, this often leads to secondary attacks as passwords are frequently reused. Password-based authentication is therefore considered vulnerable to cyberattacks (Mosakheil, 2018).

Authorisation

System functions should be available only to those with the appropriate rights. These rights are determined by the process of authorisation and applied through a variety of authorisation mechanisms (Cilliers, 2017). When applied correctly, authorisation

serves to mitigate the risk of disclosing information to unauthorised persons. Ferraiolo et al. (2003) define role-based access control (RBAC) as a system by which users gain access to computer system objects based upon their role in the organisation. The RBAC model is frequently used by EHR systems to authorise user activities (Seol et al., 2018). In this model, the application code, hosted on a central server, contains the encoded RBAC rules. These rules provide users with specific roles, which govern their access to resources. As the rules are stored in a central server, this presents yet another possible failure point for the system. A compromised system could allow an attacker to modify user privileges (either their own or those of other users) or allow them to hijack another user account, which could grant them privileged access to restricted areas of a system. The RBAC model does not deal well with complex attributes such as subject attributes, object attributes, action attributes, and contextual attributes. Subject attributes describe a user, object attributes describe the resource that the subject is attempting to access, action attributes describe the actions that the subject is attempting on the object, e.g. to read or write, and contextual attributes describe the environment, e.g. specific times when actions are allowed.

Since an EHR may contain complex attributes, it requires a mechanism that provides dynamic access control and may be set at a very fine-grained level (Seol et al., 2018). With RBAC restricting access only according to a user's role, it may need to consist of a multitude of custom role applications if fine-grained, dynamic access control is required. This approach may be difficult to maintain and track. An example is the *doctor* role. A *doctor* should not be able to access the records of all the patients in the system; only the records of their own patients should be accessible. An RBAC-based system would require each person with the *doctor* role to have individualised access rights assigned to access their respective patients (Franqueira & Wieringa, 2012).

Auditing

Audit logs are a recording of all the actions a user has performed on a system (Dekker & Etalle, 2007). They are useful in identifying how, when, where, why and by whom data was accessed, modified, and/or leaked. This yields a form of system auditing. Tamper-proof, immutable audit logs provide a means of ensuring data integrity by providing a consistent audit trail to aid in the discovery of data breaches and the identification of compromised user accounts (Kshetri, 2017). Unfortunately, EHR systems have no standardised means of generating audit logs.

2. Blockchain technology

A ledger is a structure that maintains details regarding transactions. Distributed ledger technology distributes the ledger among participants, which may be spread across various organisations and sites (Bashir, 2017, p. 27). Blockchain technology enhances this approach by chaining together unrelated blocks in a linked-list manner. This linked structure may be perceived as a chain of connected blocks, leading to the name "blockchain".

Overview

Simply put, a blockchain may be perceived as a form of distributed database which is under the control of a group of individuals. The blockchain network consists of a series of interconnected devices referred to as nodes. To add a record to this database requires that a user (on a specific node) proposes a transaction. The transaction is then broadcast to all its peer nodes, which in turn validate the transaction using known algorithms. A verified transaction is combined into a block along with other transactions. The technique for adding the block to the blockchain results in a transaction that is practically immutable (Bashir, 2017, p. 27). Fundamentally, this does not constitute new technology, but existing technology applied differently. A term frequently associated with blockchain technology is "cryptocurrency", although this is not entirely accurate. Cryptocurrency is an application of blockchain technology and thus it may be considered a subset of blockchain technology, but not an equivalent term (Bashir, 2017, p. 23).

All the peers in a blockchain network need to agree as to the validity of the history of transactions in the chain. This agreement is referred to as consensus (Bergquist, 2017). Consensus may be calculated using two approaches, namely a proof-based or a Byzantine fault tolerance-based approach. Proof-based consensus works on the principle that a leader is elected based on a form of proof that provides a specific node with the authority to propose a new value. In Byzantine fault tolerance-based consensus, new values are proposed during rounds of voting (Bashir, 2017, p. 28).

Depending on who has access to or maintains the blockchain infrastructure, blockchain networks may be classified as public, permissioned (enterprise), or private. Public blockchain networks are open to the public and anyone can partake in the consensus process (Bashir, 2017, p. 26). As public blockchain networks utilise identities based on pseudonyms, it is challenging to establish and control the identities of participants. Enterprise blockchain networks, also known as permissioned blockchains, are being developed to cater to enterprise use cases (Emmadi et al., 2019). Permissioned blockchain systems are controlled by a quorum of organisations and, as a result, are classified as semi-decentralised. The membership of a permissioned blockchain system is strictly controlled and transactions are generally confidential between participants. Privacy, confidentiality, authorisation, user identity, and auditability are key features omitted in public blockchain networks, but permissioned blockchain networks are integrating them to support enterprise-based use cases.

Table 1: Comparison of blockchain types

Criteria	Public	Permissioned	Private
Architecture	Decentralised	Semi-decentralised	Centralised
Immutability	Virtually tamper-proof	Tamper-evident	Tamper-evident
Transparency	Full transparency	Semi-transparent	Semi-transparent, No transparency
Transaction speed	Slow	Fast	Fast

Consortium networks may consist of various enterprise entities which require the private and secure sharing of information. This focus on privacy is one of the challenges to the adoption of enterprise-grade blockchain technology (Bashir, 2017, p. 461). A measure of privacy can be ensured by applying varying levels of isolation so that only authorised parties are granted access to confidential information. However, the use of a shared ledger in blockchain technology serves to promote transparency, which may be seen as a polar opposite to privacy. A goal of permissioned blockchain technology is, therefore, to attempt a balance between privacy and transparency (Emmadi et al., 2019). Private blockchain systems are classified as centralised since they are largely owned and operated by a single organisation. Various types of blockchain networks have been mentioned so far. Table 1 provides a summarised comparison of these different types of networks.

The widespread adoption of blockchain has led to the development of various independent implementations of the technology. Each of these technologies has its strengths and suitability for various applications. Table 2 compares popular blockchain platforms in terms of network type, consensus algorithm, data privacy, smart contract languages, and application (what it is used for).

Table 2: Comparison of popular blockchain platforms

Feature	Platform		
	Bitcoin	**Ethereum**	**Hyperledger Fabric**
Application	Cryptocurrency	Multi-purpose	Multi-purpose
Consensus	Proof-of-work	Proof-of-work, proof-of-stake	Solo, Kafka
Data privacy	-	ZKP	TLS, ZKP, Channels
Smart contract language	Go, C++	Solidity, Serpent, LLL	Go, Java
Type	Public	Public, private, permissioned	Private, permissioned

Blockchain platforms

Bitcoin was introduced in a white paper in the autumn of 2008. The Bitcoin open-source software was released in 2009, and the founder of Bitcoin remains anonymously known as Satoshi Nakamoto (Laurence, 2017, p. 32). Bitcoin is a popular cryptocurrency, the success of which sparked the blockchain revolution. Bitcoin makes use of an extensive consensus algorithm known as proof-of-work to validate transactions. Proof-of-work is known by the Bitcoin community as *mining*. Bitcoin miners use highly specialised equipment that is not only expensive, but also consumes large amounts of electricity to operate. Mining is necessary to keep the Bitcoin network safe, stable, and secure (Laurence, 2017, p. 34).

The developers of Ethereum were interested in turning Bitcoin into a blockchain platform that could support business and government use. Bitcoin was already well-established and would have needed a substantial code overhaul to support the number of transactions required for a business use case. The upgrade was considered too severe by the Bitcoin community and Ethereum was therefore released as a stand-alone platform in July 2015. It is currently the most developed and innovative blockchain in use (Laurence, 2017, p. 42).

Ethereum smart contracts are used to digitally verify or enforce that all contractual terms are met before a transaction takes place (Bergquist, 2017). The need for third-party involvement is eliminated by the use of these irreversible and intractable smart contracts, which demonstrates why they should be submitted for thorough testing before being deployed on a production network (Bashir, 2017, p. 198).

In 2015, the Linux Foundation initiated the Hyperledger project (Laurence, 2017, p. 81). Fabric was the first production-ready framework created in 2017 under the greater Hyperledger project. The project has since grown to encompass four other frameworks, namely Burrow, Indy, Iroha, and Sawtooth Lake (Hyperledger Architecture Working Group, 2017). Hyperledger Fabric was created to address issues such as confidentiality, privacy, and scalability (Bashir, 2017, p. 362) and also to facilitate the delivery of blockchain networks suitable for use in a business environment. Many of its modules are swappable, making it possible for developers to select a suitable consensus algorithm, such as Kafka ordering, before creating a custom blockchain network (Saraf & Sabadra, 2018). The role of the Kafka ordering service is to maintain the order of the blocks in the blockchain (Saraf & Sabadra, 2018). Swappable modules, such as the Kafka service, provide a Hyperledger Fabric implementation with a large measure of flexibility and scalability that is not available in some other types of blockchain networks, such as Bitcoin. Another feature of Fabric is its use of transport layer security (TLS), which provides a form of encrypted tunnel between two nodes and is used to preserve privacy.

Peer-to-peer technology is used to facilitate the creation of channels in Hyperledger Fabric, enabling participants to share confidential information and allowing the information to be viewable only by participants on a particular channel (Bashir, 2017, p. 362). Participants are allowed to belong to multiple channels on the same network. Many programming languages are supported in Hyperledger Fabric, via the use of container technologies, enabling developers to create chain-code (smart contracts) in languages such as Java, Node.js, and Go (Bashir, 2017, p. 362). Although a Fabric transaction is anonymous, confidential, and private, it may be traced and linked to participants by authorised auditors. This is facilitated by the membership service, with which all participants need to register to access the network (Saraf & Sabadra, 2018).

Users interacting with the Fabric network are identified by the use of digital certificates. These certificates are issued (or revoked) by the Fabric Certificate Authority (Fabric CA) (Hyperledger, 2021, p. 51). The digital certificate contains encoded authorisation attributes, as part of an attribute-based access control (ABAC) system. This allows the digital certificate to be used as a means of identifying participants and restricting their access to specific aspects of the blockchain network.

This section by no means presents all the various blockchain platforms, as providing further in-depth discussions of, among others, Kadena, Dfinity, Corda, and the various Hyperledger platforms would require a separate publication. However, this overview of the three technologies discussed does provide some insight into the wide variety of technologies available. With these technologies in mind, the following section discusses how aspects of them may be applied to security-related aspects of EHRs.

3. Applying blockchain technology to EHRs

Blockchain technology is not a one-off, drop-in replacement for all security-related aspects of EHRs. Individual features of blockchain technology may, however, present opportunities to address aspects of EHR security dimensions related to authentication, authorisation, audit logs, data storage, and transactions. The following sub-sections discuss how each of these EHR security-related services may be augmented, replaced, or enhanced using blockchain technology.

Authentication

Authentication is used to identify a user requesting access to the system. Systems need to be able to identify users to restrict access to system functions. Users' identities are also required for audit purposes (Cilliers, 2017).

Enterprise systems traditionally utilise password-based authentication, which often relies on a centralised architecture such as a relational database (Kshetri & Carolina, 2018). Blockchain technology can leverage smart contracts and public key

infrastructure (PKI) to replace password-based authentication with certificate-based authentication.

Permissioned blockchain technology can utilise smart contracts and certificate-based authentication to replace the traditional password-based authentication mechanism. Certificate-based authentication removes the human factor from the authentication process. Certificates are often created with a 2048-bit key size, which is much larger than an average password size. It is considered to be impractical to brute-force a certificate, as a standard desktop computer would take years to crack it. Certificates come with an expiration date, which can reduce the risk of prolonged data exposure. Blockchain technology can leverage smart contracts to validate user certificates and effectively mitigate the risk of a single point of failure.

Authorisation

Appropriate authorisation mechanisms should be employed to restrict user access to specific system functions. The actions that an authorised user may perform on a system are determined by the process of authorisation (Cilliers, 2017). The application of authorisation mitigates the risk of disclosing information to users who should not have access to it (Seol et al., 2018).

Enterprise systems predominantly utilise centralised authorisation architecture. Blockchain technology can replace the prominent centralised authorisation architecture with a distributed architecture. Enterprise systems commonly rely on a role-based access control (RBAC) model to restrict access to information. Blockchain technology can leverage PKI and smart contracts to create a distributed attribute-based access control (ABAC) model.

A user's certificate may be encoded with attributes to restrict their access to specific resources. Permissioned blockchain technology makes use of this attribute-based access control to enable a fine-grained access control model. This access-restriction may be based on action attributes, contextual attributes, object attributes, and subject attributes. The actions that a user is allowed to take on a system, such as reading and writing, are determined by action attributes. The types of actions a user is allowed to take may also depend on their operating system and the platform they are using to access the system, or the time of day; these attributes are referred to as contextual attributes. Object attributes are used to enforce which system object types may be accessed by the user, e.g. a medical record or information related to a specific department. Lastly, subject attributes are descriptive attributes related to the specific user requesting system access, such as departmental or job title. When used in conjunction, these different types of attributes may be encoded into smart contracts and distributed across the blockchain network. This type of distributed, authorisation architecture helps to establish a network without a single point of failure.

Audit logs

An audit log is a recording of all the actions that a user has performed on a system. Audit logs are useful in identifying how, when, where, why, and by whom data was accessed, modified, and/or leaked. Tampering with audit logs frequently occurs to cover a criminal's tracks (Dekker & Etalle, 2007). Enterprise systems predominantly utilise a centralised audit log architecture. Audit logs are commonly stored locally in a file or remotely on a relational database, but these methods of storage are not considered to be immutable. Blockchain technology could provide a distributed and practically immutable audit log. Permissioned blockchain networks make use of a membership service to identify users interacting in the blockchain network (Bashir, 2017, p. 362). The user's identity can be used to record all the actions performed by the user on the blockchain network. Permissioned blockchain technology can be used to generate a semi-decentralised, tamper-evident, and standardised audit log for EHR systems.

Data storage

Data storage is the act of recording information electronically. Data can be stored by utilising a variety of structures and architectures, all of which have advantages and disadvantages.

Enterprise systems predominantly use a centralised client-server model to store data. Centralised data storage such as a relational database used by enterprise systems provides a high degree of transaction throughput, but could be vulnerable due to a single point of failure (Liang et al., 2017). Blockchain technology can replace the common centralised client-server model with an append-only storage approach for EHR systems (Bashir, 2017, p. 438). This storage model can ensure data integrity from data creation to data retrieval. A single point of failure can also be averted with the use of blockchain technology (Kshetri & Carolina, 2018). Permissioned blockchain technology can be used to enhance the privacy of data being stored on a blockchain network. Cryptographic techniques such as zero-knowledge proofs can be used to store data privately and to ensure that data integrity can be maintained without revealing private information (Bünz et al., 2017).

Transactions

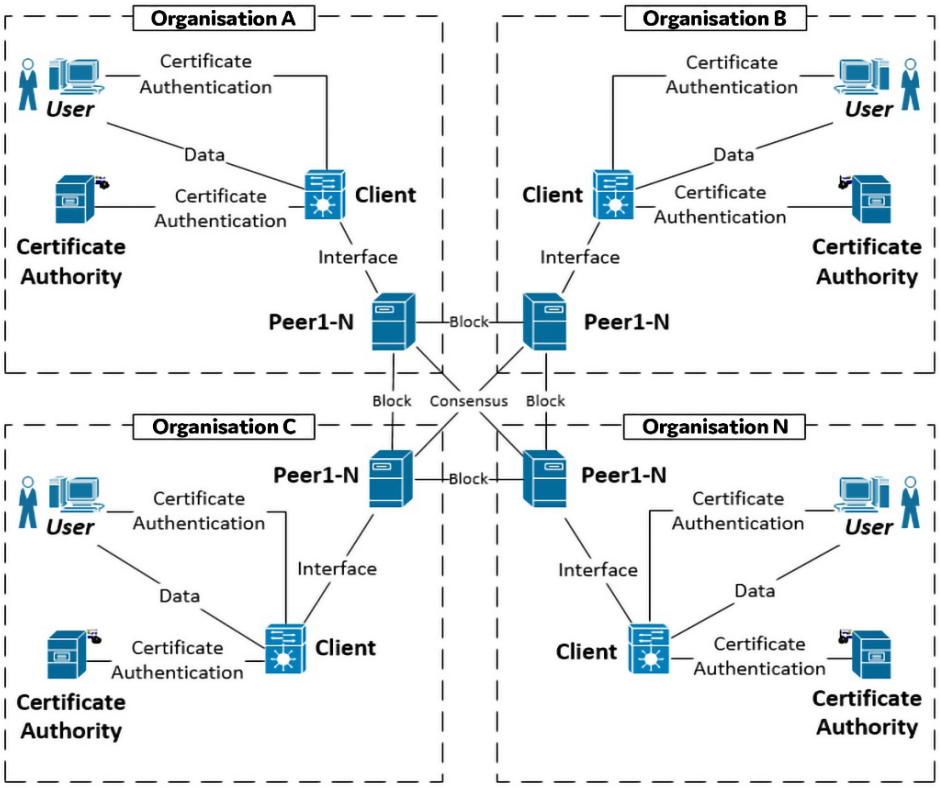
Transactions are used to add, update, or retrieve data from databases. Data transactions in this context also apply to the sharing of information between authorised parties. Information travelling on the network is a prime target for interception by those with nefarious purposes. Therefore, the common transaction process, used by enterprise systems and which still relies on a centralised client-server model, is a prime candidate for replacement by a peer-to-peer, permissioned blockchain replacement.

4. Implementation

Although the theoretical grounding seems to support the idea that certain aspects of the EHR infrastructure could be replaced with selected blockchain technologies, it is necessary to determine whether this is practically feasible. To that end, we created a series of test setups to determine whether the theoretical assumptions are accurate. Although the functionality explained in this section could be ported to any permissioned blockchain network that supports smart contracts and certificate-based authentication, Hyperledger Fabric was selected because of the features it provides, its level of customisability, and the authors' familiarity with the platform.

Figure 2 illustrates a generic version of a permissioned blockchain network, consisting of certificate authorities (CAs), clients, and peers. The role of the CAs is to issue, revoke, and/or validate digital certificates. Clients are the point of interaction with the blockchain network, and peers store the linked-list of blocks, which form part of the blockchain. The peers may be synchronised in a permissioned blockchain platform via a variety of different consensus algorithms.

Figure 2: Proposed network topology



Organisations are linked together using their respective peers. Each organisation requires at least one of each network component, as illustrated in Figure 2. The organisations are advised to include multiple peers for internal redundancy purposes. The following section uses this diagram as a baseline setup to present recommendations for applying blockchain technology to security-related services in an electronic healthcare record infrastructure.

5. Recommendations

Using the generic blockchain network described in section 4, test setups were created to address issues related to authentication, authorisation, audit logs, data storage, and transactions, as they relate to EHRs. During the creation of these test setups, various lessons were learned, which may be used to inform and/or guide anyone who wishes to implement blockchain technology to replace or augment EHR processes.

The remainder of this section, therefore, presents recommendations for applying blockchain technology to security-related services in an electronic healthcare record infrastructure. The recommendations may be used to augment individual aspects of an existing EHR system or used as a combined solution. The combination of these recommendations presents a unique EHR domain-specific overview for any systems administrator or architectural designer planning to integrate blockchain technology into an EHR system. The recommendations are grounded in theory and reinforced by insights gained from experimentation.

Use digital certificates as a means of EHR user authentication

Problems addressed

Human error or a lack of strong password selection may result in a compromised EHR system. In addition, a central database, serving as an authentication server in an EHR system, may be compromised, resulting in a disruption of service or data theft.

Motivation

Password-based authentication is not secure enough for sensitive information. The human factor in password-based authentication is the main weak point and, as passwords are often created by humans, they are usually short and weak. This is because it is difficult for humans to remember long and complex passwords. Wherever possible, make use of certificate-based authentication as it limits the human factor in password selection.

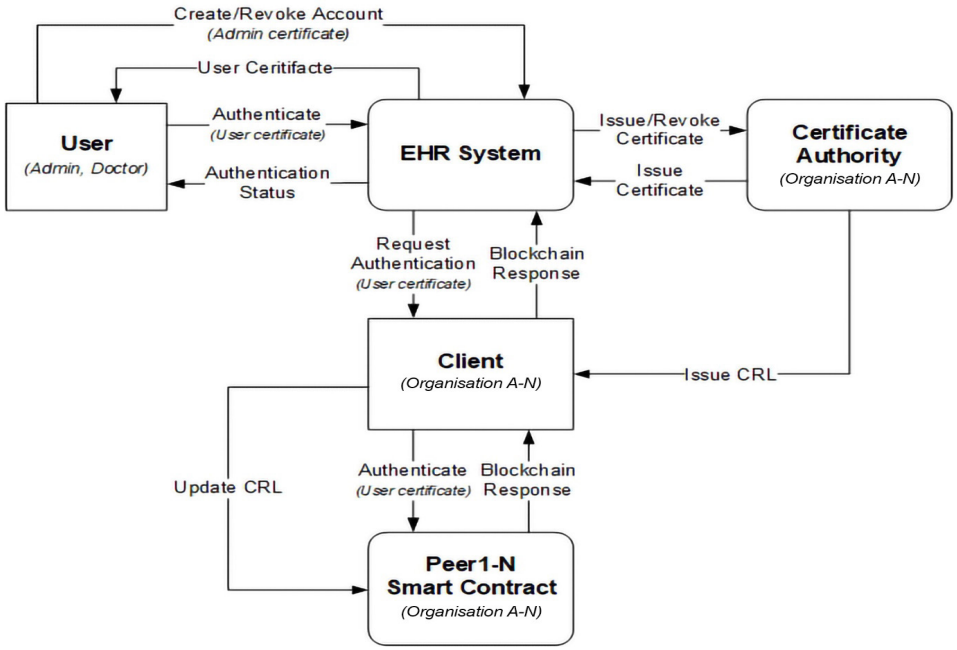
Passwords are commonly stored in a centralised database. When an authentication database is compromised, passwords can be stolen and this can result in breaches. These breaches may allow an attacker to gather sensitive information, which may, in turn, result in even more widespread breaches. Digital certificates are published with two keys, known as a private key and a public key. The public key is derived from the private key and both these keys are required for authentication. The private keys of

certificates are commonly stored on the user's machine and users are encouraged to safeguard their private keys by storing them in a hardware security module (HSM) or trusted platform module (TPM). Not all certificates are stored in a centralised architecture. The certificate revocation list (CRL) is also distributed across all the peers in the blockchain network. It is thus increasingly difficult to tamper with the authentication mechanism, as the majority of the peers in the blockchain network need to be compromised.

The blockchain approach

The information flow of the blockchain-based authentication model is illustrated in Figure 3. Administrators of the EHR system can create and revoke digital certificates and the certificates are issued and revoked by the certificate authority. When a user's digital certificate has been revoked, a CRL is generated.

Figure 3: Blockchain authentication data flow diagram



The peers in the blockchain network receive a CRL update command to update the CRL stored on the peers. Users can authenticate with the EHR system by providing the system with their certificate. The certificate is then passed to the client, which is then sent to a peer in the network. The peer authenticates the user by running the authentication smart contract, which validates the certificate and returns a response. This response determines the authentication status of a user. The pseudocode outlined in Figure 4 presents an example of an authentication function.

Figure 4: Authentication pseudocode

```
Do sanitisation check on the function inputs;
  Get the users certificate;
  Validate users certificate;
  if the user certificate has been signed by an trusted CA then

    if user certificate is not present on CRL then
      Grant user access;
    else
      Deny user access;
    end if

  else
    Deny user access;
  end if
```

Employ an attribute-based access control (ABAC) model based on EHR attributes

Problems addressed

Role-based access control (RBAC) systems require role definitions for restricting various actions and access to resources. This results in a role explosion in an EHR system, with many user and resource types requiring access control, as each customised user role, such as doctor or administrator, would require a new system role to restrict specific actions and rights on the various system attributes. In addition, RBAC rules are hosted on centralised, relational databases. A compromised relational database could provide an attacker with the ability to add, modify, or remove user privileges or even allow a specific privileged user account to be compromised to give an unauthorised user access to the system.

Motivation

RBAC-based control rules are frequently hosted on a centralised server and embedded into application code. User access requests are compared to the role description stored in a centralised relational database, presenting a possible single point of failure. The RBAC model does not deal well with complex attributes such as subject attributes, object attributes, action attributes, and contextual attributes. An EHR system contains many sensitive attributes, such as patient diagnosis, medication, or even health insurance information, and could contain multiple user-types, such as doctor, healthcare worker, and administrator, which would each need to have their own access rules defined for the various attributes in the system. For instance, a hospital administrator may need to access a patient's health insurance information, but not their medication. The large number of rules which may be required, as well as the centralised storage requirements of an RBAC-based model, make an attribute-based access control (ABAC) model a more suitable solution in the EHR domain.

ABAC presents a means to provide a more fine-grained access control model as access may be restricted based upon a combination of various actions, and contextual, object, and subject attributes. This combined approach yields a system of complex rules which may be encoded into the broader network structure. ABAC makes it a bit more impractical for an attacker to gain unauthorised access as it is based on a series of attributes and access control rules. The digital certificates in an ABAC system contain the encoded ABAC attributes and the access control rules are encoded into smart contracts which are distributed across the network. An attacker would therefore need to steal an administrator's digital certificate or compromise the majority of peers in the specific blockchain network for any illicit action to go undetected.

The blockchain approach

The information flow of the blockchain-based authorisation model is illustrated in Figure 5. The data flow diagram is based on Figure 3, but omits details regarding authorisation to focus on the authorisation process. Users should first be authenticated before the authorisation process is performed. This structure assumes a user has a valid certificate when attempting to execute an operation on the EHR system. The EHR system contacts the blockchain client to invoke the authorisation smart contract stored on the peers in the network. The smart contract validates the user attributes stored in the certificate against the authorisation rules embedded in the smart contract. The smart contract then returns an authorisation response. The pseudocode outlined in Figure 6 presents an example of an authorisation function.

Figure 5: Blockchain authorisation data flow diagram

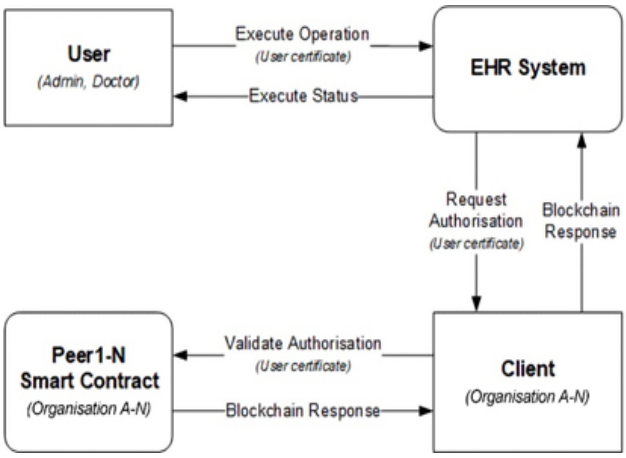


Figure 6: Authorisation pseudocode

```
Do sanitisation check on the function inputs;
   Get the users certificate;
   Validate users certificate;
   Get user's identity from certificate;
   Get user's attributes from certificate;

   if the user attributes match the authorisation rules
      Grant access to the user;

      if userID is present on a stored list
         Grant further access;
      else
         Deny further access;
      end if;

   else
      Deny access to the user;
   end if;
```

Preserve EHR data integrity using a blockchain-based audit log model

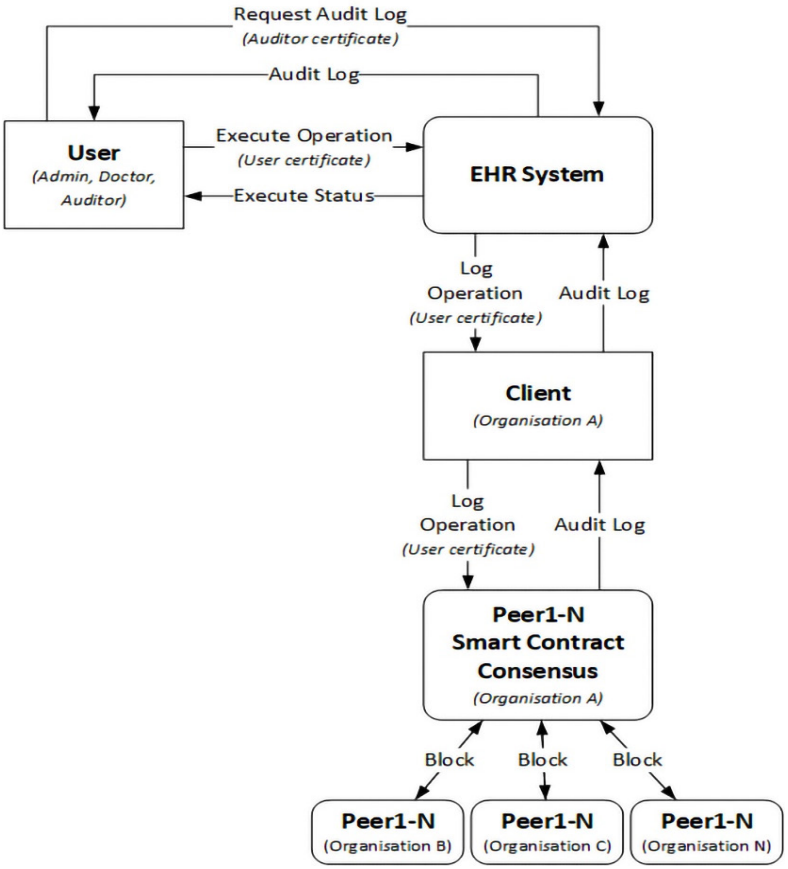
Problem addressed

Traditional EHR audit log systems are built around a centralised architecture. Audit logs are often stored in a relational database or on a file server. While these methods of storage work practically, they represent a single point of failure. Compromising one of these storage methods would enable cybercriminals to erase their tracks, thus allowing their actions to remain undetected. As a result, the integrity of the data stored in the relational database cannot be guaranteed. When dealing with the health information of patients, compromised data could have deadly consequences.

Motivation

Blockchain-based audit logs are permanent and tamper-evident. Blockchain is an append-only data structure that is distributed across several peers and cybercriminals would have to attack the majority of the peers in the network simultaneously to corrupt the audit log. This attack would not go unnoticed. Even if cyber-criminals did hijack a user's account, the changes made by the account would not go undetected. The changes made to the audit log would be appended, leaving the previous records intact. This could then be used to flag suspicious accounts and track the cybercriminals responsible. Data integrity can therefore be preserved through the use of blockchain technology.

Figure 7: Blockchain audit log data flow diagram



The blockchain approach

The information flow of the blockchain-based audit log model is illustrated in Figure 7. The data flow diagram is based on Figures 3 and 5. The audit log data flow diagram omits the authentication and authorisation process to simplify the diagram. The process assumes that a user has been authenticated. When a user attempts to execute an operation on the EHR system, an event is triggered, which sends metadata to the client. Metadata could include details such as the actions performed by a user on an object and the result of the performed actions. The metadata is sent from the client to the peers in the blockchain network. At a later stage, authorised auditors can request the audit log from the EHR system. The audit log could also be used by doctors to validate the integrity of EHR data in their possession. Figure 8 outlines pseudocode to retrieve an audit log by range. Figure 9 provides pseudocode to append an audit log entry to a blockchain network.

Figure 8: GetAuditLog() pseudocode

```
Function AuditLog GetAuditLog(Date startDate, Date endDate) {

  Do sanitisation check on the function inputs;
  Get the user certificate;
  Validate user certificate;
  Get user's attributes from certificate;

  if the user attributes match the authorisation rules
     return AuditLogByRange(startDate, endDate);
  else
     Deny access to the user;
     CreateUnauthorisedLogEntry(log);
  end if; }
```

Figure 9: AppendAuditLog() pseudocode

```
Function AppendAuditLog(AuditLog log) {

  Do sanitisation check on the function inputs;
  Get the device certificate;
  if device certificate is not valid then
     Deny access to the device;
  end if

  Get device attributes from certificate;
  if the device attributes match the authorisation rules
     Get user certificate;
     if user certificate is valid then
        CreateLogEntry(log);
     else
        CreateUnauthorisedLogEntry(log);
     end if
  else
     CreateUnauthorisedLogEntry(log);
     Deny access to the device;
  end if;  }
```

Ensure EHR data immutability by adopting a blockchain-based storage model

Problem addressed

Traditional storage models are mostly built around a centralised architecture such as a relational database. Relational databases do not store data in an immutable manner. This practice may not align with all the policies and regulations regarding the storage of EHRs.

Motivation

The nature of blockchain technology is to store records immutably in an append-only format. Storing EHRs in a blockchain network is mostly in line with the policies surrounding EHRs. Laws and policies differ from country to country or region to region, but the Health Professions Council of South Africa (HPCSA) stipulates that when health records are stored in an electronic format, they should be stored in an append-only format (HPCSA, 2016). Copies of the records should be made and stored in different physical locations. The copies are used to detect tampering with EHRs. Health records should also be kept for at least five years. The South African Protection of Personal Information (POPI) Act, however, states that users should be able to request that their personally identifiable information be purged from a service (RSA, 2013, sect. 24).

Relational databases satisfy the personally identifiable information requirements by supporting the purging of records as stipulated in legislation such as the POPI Act. Blockchain technology is aligned with these policies, as the data stored in a blockchain network is distributed across peer nodes situated in different physical locations. As blockchain technology stores data immutably, it cannot satisfy this requirement. Blockchain technology can, however, support the traditional centralised infrastructure by storing a hash of data that is contained in a relational database. This method of storing EHRs would enable stakeholders to run integrity checks on data stored in the traditional architecture. The hash of the data stored in the traditional storage model can be compared with the hash stored in the blockchain storage model. The two hashes should be identical to pass an integrity check. Blockchain technology can thus support the integrity of EHRs stored in traditional architecture.

The blockchain approach

The information flow of the blockchain-based storage model is illustrated in Figure 10. The data flow diagram is based on Figures 3 and 5. The storage data flow diagram omits the authentication and authorisation process to simplify the diagram. The process assumes that a user has been authenticated; when that user wants to view, add to, or edit a patient's record, they can do so by contacting the EHR system. The EHR system would then request or send the information to the blockchain client. The blockchain client then forwards the request to one of the peers in the network and the relevant smart contract code is then executed on the peer. The smart contract then proposes a transaction to the blockchain network. This transaction is bundled together into a block and appended to the blockchain. The consensus algorithm ensures that all the peers are synchronised.

Figure 10: Blockchain storage data flow diagram

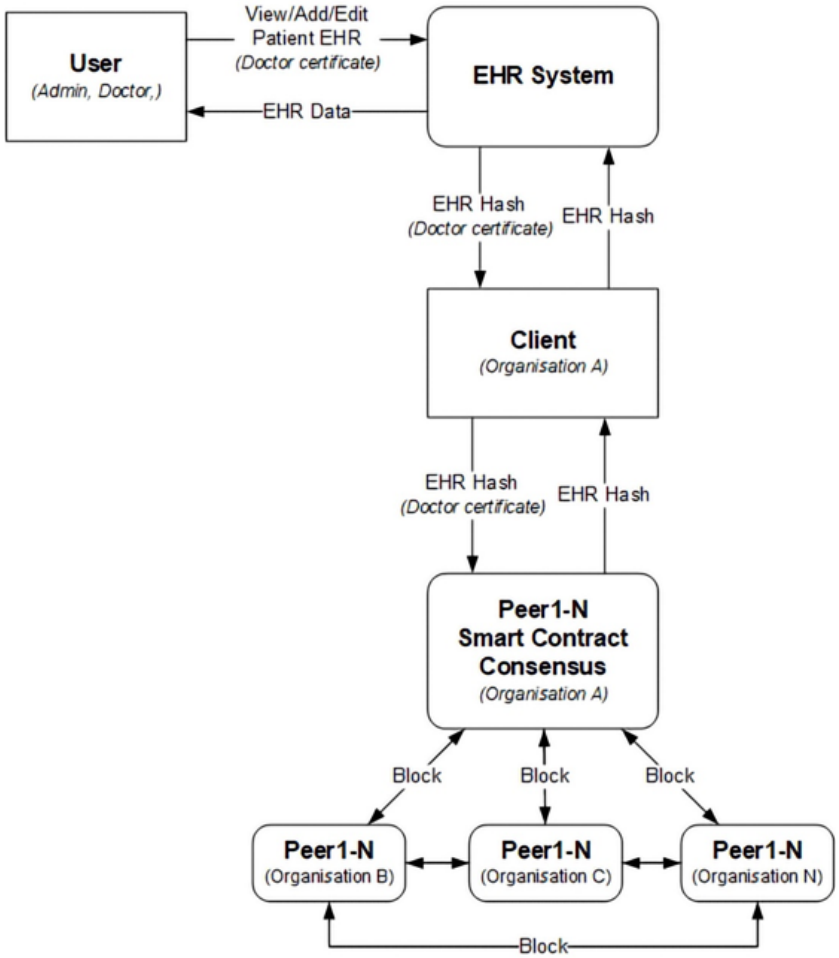


Figure 11 outlines pseudocode for adding or updating a record in the blockchain network. Retrieving records from the blockchain could be achieved with the pseudocode written in Figure 12.

Figure 11: AppendData() pseudocode

```
Function AppendData(Data data) {

   Do sanitisation check on the function inputs;
   Get the user certificate;
   Validate user certificate;
   Get user's attributes from certificate;
   if the user attributes match the authorisation rules
      CreateDataEntry(data);
   else
      Deny access to the user;
   end if;  }
```

Figure 12: GetData() pseudocode

```
response Function GetData(Key key) {

   Do sanitisation check on the function inputs;
   Get the user certificate;
   Validate user certificate;

   Get user's attributes from certificate;
   if the user attributes match the authorisation rules
      if user is present on authorisation list
         return GetData(key);
      else
         Deny access to the user;
      end if
   else
      Deny access to the user;
   end if;  }
```

Transact private EHR data using a blockchain-based transaction model

Problem addressed

The traditional transaction model relies on a third party to handle private information. The information would be sent from the sending client to a centralised third-party server back to the receiving client. This approach can increase the risk of a man in the middle attack.

Motivation

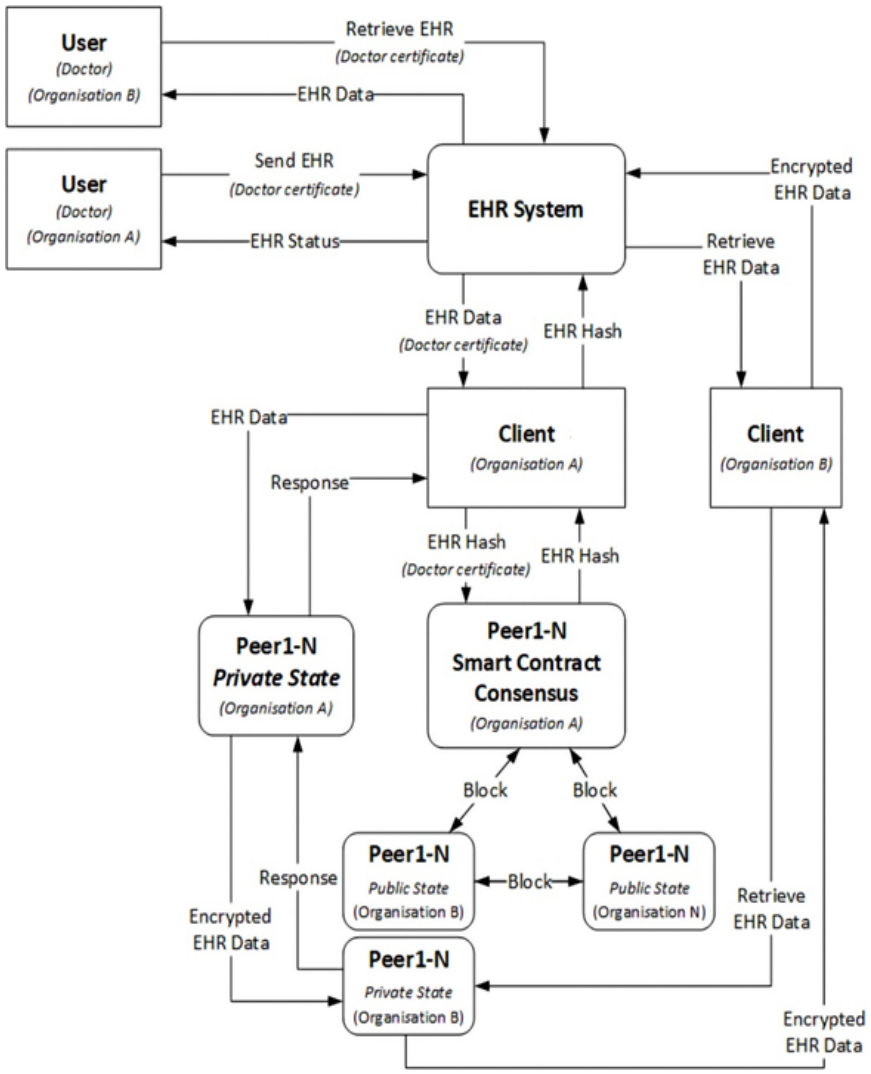
Blockchain technology enables a sending client to send private information directly to the receiving client without relying on a third party to relay the information, thus enabling healthcare providers to transact a patient's health records without relying on a third party.

Blockchain technology could be used as a synchronisation service to transact information between organisational data stores. Coupling the blockchain-based transaction model with the authentication, authorisation, and audit log model would establish a robust sync service with full audibility across multiple organisations, which could include hospitals, private practices, pathology laboratories, medical insurance companies, pharmacies, auditors, or medical boards. This would, in turn, enable patients to visit any healthcare provider that is a part of the blockchain network. The patient could then provide authorisation to the healthcare provider to sync their information with its data stores, essentially providing the healthcare provider with their EHR. The authentication, authorisation, and audit log model would be distributed across all the organisations, logically forming a single unified system. Blockchain technology could thus provide a robust semi-decentralised transaction model.

The blockchain approach

The information flow of the blockchain-based transaction model is illustrated in Figure 13. The data flow diagram is based on Figures 3 and 5. The transaction data flow diagram omits the authentication and authorisation process to simplify the diagram. The process assumes that a user has been authenticated. Users from one organisation can send a patient's EHR to another organisation, and the EHR system encrypts the record and sends it to the blockchain client. The blockchain client forwards the encrypted EHR data to all the organisational peers involved in the transaction. The respective peers store this transaction in their private state. This means that only the organisations involved in this transaction would have the EHR data stored in their peer's private state. The blockchain client then creates a hash of the transacted EHR data and broadcasts that to all the peer's public states. These peers also include peers from other organisations that are not a part of the transaction. This is to ensure a level of transparency and auditability across organisational bounds. The organisations' part of the transaction can then retrieve the EHR record from their peer's private state. The sending organisation could specify an expiration date for the transaction, meaning that the data would be available to an organisation only for a specific period. After the period has expired, the data is automatically purged from the blockchain and only the hash of the transaction remains.

Figure 13: Blockchain transaction data flow diagram



The *TransactData* method in Figure 14 outlines pseudocode to transact data between organisations that are part of the blockchain network. Retrieving transacted data could be achieved with the pseudocode function presented in Figure 15.

Figure 14: TransactData() pseudocode

```
Function TransactData(Transient data, Recipients recipients, Time TimeToLive) {

  Do sanitisation check on the function inputs;
  Get the users certificate;
  Validate users certificate;
  Get user's identity from certificate;
  Get user's attributes from certificate;

  if the user attributes match the authorisation rules
    if userID is present on authorisation list
        for each recipient
          Set data expiration date;
          Encrypt data with recipient public key;
          Encrypt data with recipient peer public key;
          Send data to receipt's peer private state;
        next
        StorePublic(senderSignature, recipients, dataHash) for all peers;
    else
        Deny further access;
    end if;

  else
     Deny access to the user;
  end if;   }
```

Figure 15: GetTransactedData() pseudocode

```
response Function GetTransactedData(Key key) {

   Do sanitisation check on the function inputs;
   Get the users certificate;
   Validate users certificate;
   Get user's identity from certificate;
   Get user's attributes from certificate;

   if the user attributes match the authorisation rules
     if userID is present on authorisation list
         Return GetPeerPrivateSate(key);
     else
         Deny further access;
     end if;

   else
      Deny access to the user;
   end if;  }
```

6. Conclusions and future work

To aid in the application of blockchain technology to existing or new EHR infrastructure, this study set out to present recommendations for applying blockchain technology to security-related services in an electronic healthcare record infrastructure. To this end, experimental setups were created to address specific requirements of EHRs, using blockchain technology. The insights gained from these experiments were condensed into a series of recommendations for the application of blockchain technology to security-related services in EHRs.

Although it is most certainly a viable alternative, blockchain technology is not necessarily the best solution in all cases. Its immutability is a strength when it comes to preserving details, but also a weakness in a world governed by privacy laws, regulations, and Acts, such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States, the European General Data Protection Regulation (GDPR), and South Africa's POPI Act. Implementation of the various technologies may also require expertise that may not be found among the administrators of existing EHR systems. The costs associated with the change in infrastructure may also be prohibitive. Therefore, the application of blockchain technologies may be a better choice for implementing new EHR systems and not for the conversion of existing systems.

The work presented in this study is experimental in nature and has been implemented only in a virtual environment. A future study will focus further on the shortcomings and strengths of EHRs, by surveying the stakeholders of existing EHR systems. These insights may then be used, along with further experimentation, to derive a model for the application of blockchain technology to security-related services in EHR systems. Future studies may also delve into how the concept of self-sovereign identity (SSI), which allows a person to have sole control over who may access their personal information (Ferdous et al., 2019), may be integrated into a blockchain-based EHR system.

References

Adlam, R., & Haskins, B. (2019). A permissioned blockchain approach to the authorization process in electronic health records. In IEEE (Ed.), *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)* (pp. 1–8). https://doi.org/10.1109/IMITEC45504.2019.9015927

Bashir, I. (2017). *Mastering blockchain.* Packt Publishing.

Bergquist, J. H. (2017). *Blockchain technology and smart contracts privacy-preserving tools.* Master's thesis, Uppsala University, Sweden. http://uu.diva-portal.org/smash/get/diva2:1107612/FULLTEXT01.pdf

Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., & Maxwell, G. (2017). Bulletproofs: Short proofs for confidential transactions and more. In IEEE (Ed.), *2018 IEEE Symposium on Security and Privacy* (pp. 315–334). https://doi.org/10.1109/SP.2018.00020

Cilliers, L. (2017). Exploring information assurance to support electronic health record systems. In IEEE (Ed.), *2017 IST-Africa Week Conference (IST-Africa)* (pp. 1–8). https://doi.org/10.23919/ISTAFRICA.2017.8102363

Dagher, G. G., Mohler, J., Milojkovic, M., & Marella, P. B. (2018). Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustainable Cities and Society*, *39*, 283–297. https://doi.org/10.1016/j.scs.2018.02.014

Dekker, M. A. C., & Etalle, S. (2007). Audit-based access control for electronic health records. *Electronic Notes in Theoretical Computer Science*, *168*(1), 221–236. https://doi.org/10.1016/j.entcs.2006.08.028

Emmadi, N., Vigneswaran, R., Kanchanapalli, S., Maddali, L., & Narumanchi, H. (2019). Practical deployability of permissioned blockchains. In W. Abramowicz, & A. Paschke (Eds.), *Business information systems workshops* (pp. 229–243). Springer International. https://doi.org/10.1007/978-3-030-04849-5_21

Ferdous, M. S., Chowdhury, F., & Alassafi, M. O. (2019). In search of self-sovereign identity leveraging blockchain technology. *IEEE Access*, *7*, 103059–103079. https://doi.org/10.1109/ACCESS.2019.2931173

Ferraiolo, D., Kuhn, D. R., & Chandramouli, R. (2003). Role-based access control. Artech House.

Franqueira, V. N. L., & Wieringa, R. J. (2012). Role-based access control in retrospect. *IEEE Computer*, *45*(6), 81–88. https://doi.org/10.1109/MC.2012.38

Griggs, K. N., Ossipova, O., Kohlios, C. P., Baccarini, A. N., Howson, E. A., & Hayajneh, T. (2018). Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. *Journal of Medical Systems*, *42*(7), 130. https://doi.org/10.1007/s10916-018-0982-x

Guo, R., Shi, H., Zhao, Q., & Zheng, D. (2018). Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems. *IEEE Access*, *6*, 11676–11686. https://doi.org/10.1109/ACCESS.2018.2801266

Health Professions Council of South Africa (HPCSA). (2016). *Booklet 9: Guidelines on the keeping of patient records.*

Hyperledger. (2021). Hyperledger-fabricdocs documentation: Release master. Hyperledger. https://buildmedia.readthedocs.org/media/pdf/hyperledger-fabric/release-1.4/hyperledger-fabric.pdf

Hyperledger Architecture Working Group. (2017). Hyperledger architecture, volume 1. https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf

Kshetri, N. (2017). Blockchain's roles in strengthening cybersecurity and protecting privacy. *Telecommunications Policy*, *41*(10), 1027–1038. https://doi.org/10.1016/j.telpol.2017.09.003

Kshetri, N., & Carolina, N. (2018). Blockchain and electronic healthcare records. *IEEE Computer Society*, *51*(12), 59–63. https://doi.org/10.1109/MC.2018.2880021

Laurence, T. (2017). *Blockchain for dummies*. Wiley.

Liang, X., Zhao, J., Shetty, S., Liu, J., & Li, D. (2017). Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In IEEE (Ed.), *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* (pp. 1-5). https://doi.org/10.1109/PIMRC.2017.8292361

Menachemi, N., & Collum, T. H. (2011). Benefits and drawbacks of electronic health record systems. *Risk Management and Healthcare Policy*, *4*, 47–55. https://doi.org/10.2147/RMHP.S12985

Mosakheil, J. H. (2018). Security threats classification in blockchains. *Culminating Projects in Information* Assurance, 48. https://repository.stcloudstate.edu/msia_etds/48/

Republic of South Africa (RSA). (2013). Protection of Personal Information Act 4 of 2013. *Government Gazette*, Vol. 581, No. 37067.

Ronquillo, J. G., Winterholler, J. E., Cwikla, K., & Szymanski, R. (2018). Health IT, hacking, and cybersecurity: National trends in data breaches of protected health information. *Journal of the American Medical Informatics Association*, *1*, 15–19. https://doi.org/10.1093/jamiaopen/ooy019

Saraf, C., & Sabadra, S. (2018). Blockchain platforms: A compendium. In IEEE (Ed.), *2018 IEEE International Conference on Innovative Research and Development (ICIRD)* (pp. 1–6). https://doi.org/10.1109/ICIRD.2018.8376323

Seol, K., Kim, Y.-G., Lee, E., Seo, Y.-D., & Baik, D.-K. (2018). Privacy-preserving attribute-based access control model for XML-based electronic health record system. *IEEE Access*, *6*, 9114–9128. https://doi.org/10.1109/ACCESS.2018.2800288

Thakkar, M., & Davis, D. C. (2006). Risks, barriers, and benefits of EHR systems: A comparative study based on size of hospital. *Perspectives in Health Information Management*, *3*(5), 1–19.

Ziglari, H., & Negini, A. (2017). Evaluating cloud deployment models based on security in EHR system. In IEEE (Ed.), *2017 International Conference on Engineering and Technology (ICET)* (pp. 1–6). https://doi.org/10.1109/ICEngTechnol.2017.8308142