*future internet*

*Article*

# eHealth Integrity Model Based on Permissioned Blockchain †

**Tomasz Hyla *** **and Jerzy Pejaś**

Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, 70-310 Szczecin, Poland; jpejas@zut.edu.pl

* Correspondence: thyla@zut.edu.pl

† This paper is an extended version of our paper published in International Conference on Cyber Security and Communication Systems, Melbourne Institute of Technology, Melbourne, Australia, 10–12 December 2018.

Received: 17 January 2019; Accepted: 20 March 2019; Published: 24 March 2019

**Abstract:** (1) Background: Large eHealth systems should have a mechanism to detect unauthorized changes in patients' medical documentation, access permissions, and logs. This is due to the fact that modern eHealth systems are connected with many healthcare providers and sites. (2) Methods: Design-science methodology was used to create an integrity-protection service model based on blockchain technology. Based on the problem of transactional transparency, requirements were specified and a model was designed. After that, the model's security and performance were evaluated. (3) Results: a blockchain-based eHealth integrity model for ensuring information integrity in eHealth systems that uses a permissioned blockchain with off-chain information storage was created. In contrast to existing solutions, the proposed model allows information removal, which in many countries' eHealth systems is a legal requirement, and is based on a blockchain using the Practical Byzantine Fault Tolerant algorithm. (4) Conclusion: A blockchain can be used to store medical data or only security-related data. In the proposed model, a blockchain is mainly used to implement a data-integrity service. This service can be implemented using other mechanisms, but a blockchain provides a solution that does not require trusted third parties, works in a distributed eHealth environment, and supports document removal.

**Keywords:** eHealth; blockchain; integrity; transactional transparency; proof of existence

## 1. Introduction

Nowadays, healthcare-information systems are becoming increasingly popular worldwide. They have many undeniable advantages compared to paper-based documentation. Several eHealth systems are used during the full healthcare process. Starting from systems for managing patient documentation (e.g., an Electronic Health Record (EHR)), managing organizational issues in a healthcare site (such as patient admission), and ending with financial systems. The basic concept of the EHR is that it is a virtual container for health-related documentation for a subject of care (for a precise definition, see ISO 13606 [1]). The security of every system depends on the used security measures. Because of the rising complexity of eHealth systems, i.e., many interconnected healthcare providers and many healthcare sites, the need to allow EHR integrity verification is rising. An eHealth system should contain a tool allowing to detect any changes in patient medical documentation. One of the most promising technologies that allows implementing EHR integrity is blockchains.

One really needs to understand the real reasons of using blockchains in eHealth systems. In larger eHealth systems that consist of several different systems managed by different providers, a service should provide transactional transparency of an individual's personal medical record. On the other hand, resistance to data loss has currently been successfully implemented using other mechanisms. Stored data in cloud systems or in professional data centers is usually replicated between a few physical locations or hard drives. Even when a user sees one logical storage device, redundancy causes a system to be secure against data loss caused by device failure.

Through the rest of this paper, the term 'eHealth system' means an information system that manages medical data and processes, among others: health-related documents (EHRs), permissions given by a subject of care or other entity that allow access to a specified medical record, audit logs, and other data related to the healthcare process, such as billing information or appointment queues.

### *1.1. Related Works*

Information security must be taken into account during the design of eHealth systems due to the sensitive nature of medical records. Nowadays, cyberattacks on eHealth systems can impact the basic security properties of medical records, i.e., availability, confidentiality, and integrity. Security requirements for information systems that process medical records are described in ISO TS 18308 [2]. Furthermore, eHealth security is an active research area due to the implementation of new technologies, including cloud technology, Internet-of-Things (IoT) devices, or blockchains, and growing numbers of attacks. Rezaeibagha et al. [3] reviewed the literature regarding the security and privacy of EHR systems, and identified essentials features for EHR security and privacy. The standards and guidelines related to health-information-system development were analyzed by Carvalho et al. [4]. The security and privacy requirements for storing medical records in the Cloud from the perspective of healthcare providers and cloud-service providers were analyzed by Rodrigues et al. [5] and by Sahi et al. [6]. Rodrigues et al. [5] concluded that storing EHRs in the Cloud means that precautions must be taken to ensure data security. The security of eHealth systems was also discussed by Sahama et al. [7]. The problem of key management and authentication in IoT devices working in an eHealth system was reviewed by Aghili et al. [8]. They proposed an authentication and key management protocol with support for privilege transfer that is both secure and energy-efficient.

Blockchain technology can be used to build new mechanisms that enhance the security of eHealth systems. Blockchains should be used, among other situations (according to Reference [9]), where multiple contributors exist, more trust is required, the need for reliable tracking activity exists, and where data must be reliable over time. For this reason, a blockchain is seen by many as a type of technology that could improve the security of eHealth systems. In eHealth applications, typically permissioned (private) blockchains are used where data are embedded into a blockchain (on-chain storage) or just record hashes are embedded into a blockchain (off-chain storage). For example, Liu et al. [10] proposed the Advanced Blockchain approach, where data are encrypted and stored on-chain (embedded into transactions), Azaria et al. [11] proposed MedRec, which is a decentralized record-management system based on a blockchain, and Liang et al. [12] proposed a user-centric health-data sharing solution for mobile environments. In proposals where data are stored on-chain, there is a problem with scalability and efficiency, as many authorized nodes have to get a full blockchain copy. A 2019 study conducted by Park et al. [13] confirmed that it is possible to exchange EHR data in a private blockchain network, but many improvements are required, including data-size reduction and security issues. Another study, conducted by Roehrs et al. [14], presented results from testing OmniPHR using a dataset of more than 40,000 patients, achieving an average response time of below 500 ms. In OmniPHR, not all nodes store a complete blockchain copy. Such an approach was used to manage large datasets.

A blockchain was used in several other eHealth applications. For example, a private blockchain (Hyperledger Fabric) was used by Ichikawa et al. [15] to ensure the integrity and availability of stored personal health records. They created a tamper-resistant mHealth application that enables cognitive behavioral therapy for insomnia using a smartphone. Next, Zhou et al. [16] proposed the MIStore, which is a blockchain-based medical-insurance storage system.

A secure EHR system based on blockchain technology and an attribute-based cryptosystem was proposed by Wang and Song [17]. Zhou et al. [18] proposed BeeKeeper 2.0, a confidential blockchain-enabled IoT system using a decentralized outsourcing computation scheme. Lastly, Li et al.

proposed a blockchain-based data-preservation system (DPS) for medical data. This system allows users to store the preservation in a blockchain, and to prove the primitiveness and originality of the stored data.

### *1.2. Motivation*

The motivation for this work was to create a solution that allows ensuring transactional transparency in distributed eHealth environments, so every patient or auditor can verify if given medical histories or logs are complete and unchanged. Such a service is necessary because eHealth systems are increasingly complex and distributed, and process rising amounts of data. For this reason, it is difficult to verify the integrity of distributed medical records using organizational measures or simple integrity mechanisms, implemented in individual databases. In distributed eHealth environments, the nodes at healthcare sites by default operate according to the law. However, due to numerous reasons, from which the most important are cyberattacks and rogue administrators, unauthorized changes might occur in a medical history or log. The detection of these changes allows restoring the data from database replicas or backups that are mandatory in eHealth systems.

### *1.3. Contributions*

In this paper, a Blockchain-based eHealth Integrity Model (BEIM) is proposed that ensures information integrity in eHealth systems and uses permissioned blockchains with off-chain information storage. In contrast to existing solutions, the paper shows how to allow for information removal, as it is a legal requirement to have such an option in many countries. Several security problems related to blockchains that must be considered during blockchain development in eHealth systems are also discussed. The proposed model can easily be integrated with systems using service-oriented architectures.

Blockchain technology can be used to store complete medical data or only security-related data. In the BEIM, a blockchain is mainly used to implement a data-integrity service. The blockchain provides a solution that does not require trusted third parties, and enables the easy synchronization of a common history of events coming from multiple sources. Furthermore, it allows to link together records related to one patient that are simultaneously produced in many healthcare sites. As is discussed above, one of the most important applications of blockchain technology in EHRs concerns data-integrity preservation and the unified logging of medical records. However, in many cases, a blockchain is proposed to store all EHRs (e.g., Li et al. [19]). In the BEIM, the blockchain is used as designed, i.e., to create rigorous and tamper-resistant data registers and transactions related to them. Furthermore, the BEIM uses a blockchain that is not related to cryptocurrency.

The main research question was to find if it is possible to design a model allowing to achieve transactional transparency in eHealth systems that does not require trusted third parties and supports document removal. The model should preserve medical-record confidentiality.

The present work is the full version of our paper. In comparison to the conference version [20], it provides an extended introduction, new sections on security assumptions and adding and removing records, an improved and extended discussion, results from a performance test, and additional information.

### *1.4. Paper Organization*

The remainder of the paper is organized as follows. Section 2 contains description of the BEIM. Section 3 contains a discussion with emphasis on the long-term security aspects of blockchains, and privacy issues related to blockchain security analysis and performance. The paper ends with our conclusions.

## 2. Blockchain Based eHealth Integrity Model

This section begins with the description of security assumptions and requirements for security services that provide transactional transparency. These services are difficult to implement using currently available security mechanisms. Next, the model components and the way in which they can be integrated with eHealth systems, built using service-oriented architecture, are presented. Then, record addition and removal is described. A detailed verification algorithm that allows verifying integrity of EHRs, access rights, or log entries, as well as proof of existence of records is presented at the end of the section.

### *2.1. Methodology*

Design-science methodology was used to create the model. Based on the problem of transactional transparency, assumptions and requirements were specified. Next, the architecture and algorithms were designed. After that, the security of the model was evaluated, followed by performance evaluation that includes test results from implementation of a basic integrity-verification scenario. The design of the model was updated using intermediate steps from evaluation.

### *2.2. Security Assumptions*

The BEIM was built using a blockchain as the basic component to ensure record integrity. A blockchain is a distributed ledger where multiple participants share the same copy of the ledger. The distributed ledger is protected against tampering because it is not required that all participants be trusted. Furthermore, a consensus protocol must be in place that allows maintaining ledger consistency. Recently, Yoshihama and Saito [21] listed and described integrity and confidentiality requirements. The integrity requirements are (for full definitions, see Reference [21]):

1. Agreement on Transaction Validity: only a valid transaction can be recorded in the ledger;
2. Tamper Evidence: the ledger cannot be tampered with and is consistent among participants;
3. Finality: no transaction is rejected after it is approved and recorded in the ledger.

The blockchain used in the BEIM must meet the above three requirements. Additionally, it must be possible to implement the following privacy (confidentiality) requirements (for full definitions, see Reference [21]):

1. Anonymity to Third Parties: no third party can see the true identity of the subjects, and
2. Confidentiality of Transaction Content: third-party subjects or nodes cannot view transaction content.

However, in the BEIM, anonymity only concerns patients. The nodes or systems that create and add transactions to the blockchain do not have to be anonymous. Moreover, it is assumed that the used hash functions must be secure cryptographic hash functions, and a public key (asymmetric) encryption algorithm must be considered as secure.

### *2.3. Requirements*

In the presented model, it was assumed that the eHealth system has a distributed architecture. In such an architecture, one or more central (usually government-owned) clouds and many interconnected healthcare sites (e.g., hospitals) exist. In the BEIM, it is assumed that all healthcare sites use software certified by a government authority that allows access to central health services. The BEIM allows implementing the following security services that can together provide transactional transparency in the eHealth system using a permissioned blockchain. These are services allowing the verification of:

1. EHR integrity;
2. access permissions to records in each EHR; and

3. audit logs related to each EHR.

In all three cases, it should be possible to detect if all records are present, unmodified, and that the order in which they were added is preserved (backdating detection). Additionally, it should be possible to prove that a document was not added.

A fundamental blockchain property is the inability to delete or modify data that already exist in a blockchain. This contradicts a legal requirement for almost every eHealth system to have the option to remove patient records. Usually, in case of an erroneous entry, a new entry is added. However, there are legal cases that require a removal option. Two types of transaction ('add' and 'remove') are used to implement that option. Therefore, there is no need to remove transactions from the blockchain.

Taking into account the sensitive nature of medical records, transactions in a blockchain should not contain any unencrypted data about patient health. Transactions can only contain metadata about medical documents (records), access rights, or log entries.

### *2.4. Architecture*

In the BEIM, we assume that the eHealth system is designed using a service-oriented architecture. Hospitals, medical centers, or general practitioners (GPs) might internally use software for managing medical records from different vendors that is compliant with the eHealth system. Therefore, a blockchain could be added to existing systems with minimum work (see Figure 1). There are two types of nodes, authorities and client nodes. In the proposed model, the authorities are nodes located inside eHealth clouds, and in major hospital information systems. Smaller hospitals and GP offices that store some medical records can only run client nodes. Authority nodes run a blockchain protocol, and client nodes store only a copy of the blockchain. Thus, the model can be deployed in all types of healthcare sites. In Figure 1 the yellow connections are secure logical links used for running a blockchain protocol. The dotted yellow lines show secure links used by client nodes to download and synchronize the current blockchain.

Due to the integrity requirements that are presented in Section 2.2, the consensus mechanism should be implemented using the Practical Byzantine Fault Tolerant (PBFT) algorithm. Therefore, the permissioned blockchains that use PBFT consensus must be used in the BEIM, e.g., Multichain [22] or Hyperledger Fabric [23]. Other popular consensus algorithms, such as Proof-of-Work (PoW), Proof-of-Stake (PoS), or Proof-of-Authority (PoA), do not meet one or more integrity requirements. The PoW and PoS algorithms cannot be used as, in practice, they can only be applied to cryptocurrencies [24]. These algorithms also do not fulfil the Finality property [21]. PoA [25] is a consensus protocol in which $N$ trusted nodes (authorities) participate that are uniquely identified, and their majority $N/2+1$ are assumed to be honest. The PoA consensus is based on a mining rotation scheme. Examples of PoA algorithms are Aura [26] and Clique [27]. However, according to analysis provided by Angelis et al. [25], PoA does not ensure consistency.

Due to the privacy (confidentiality) requirements, in the BEIM, the blockchain is permissioned, and only certified nodes have access to it. Only off-chain storage is used, where only hashes and other metadata are stored in the transactions submitted to the blockchain. The transaction body is encrypted and identified by a covert patient identifier. The real identifier is only available to nodes processing the patient's documents, i.e., adding or verifying the EHR or logs. Each node that creates a transaction encrypts it (except for a covert user identifier) using a user's public key that is only used to encrypt blockchain transactions. The private key that enables the decryption of blockchain-transaction content for any subject of care is given by the central eHealth service to an entity that has an EHR integrity-verification access right. The right is given to entities that have access right to the specified EHR. After verification, the private key is destroyed by the verifier's application.

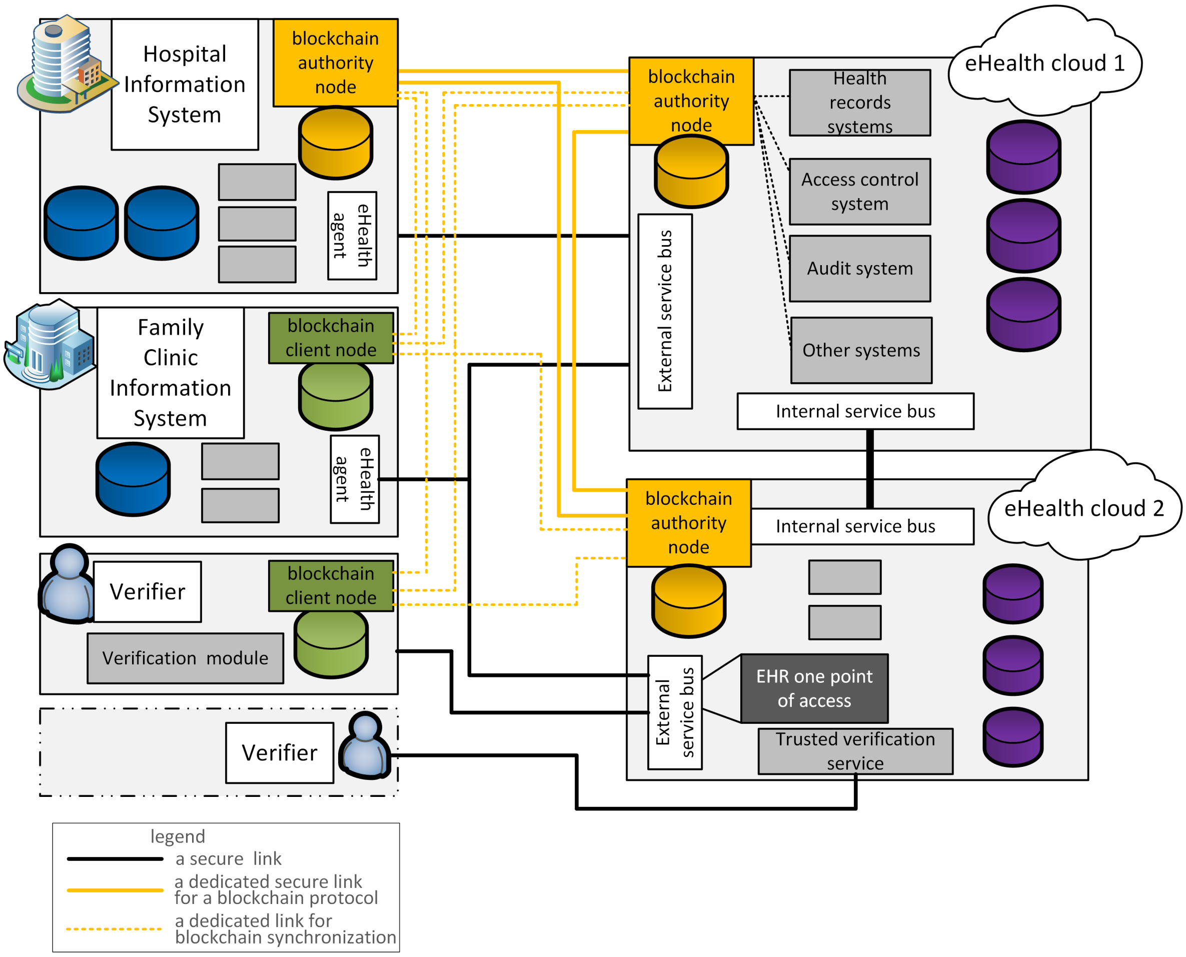

**Figure 1.** Integration of the Blockchain-based eHealth Integrity Model (BEIM) into a service-oriented architecture.

There are three types of transactions:

1. add a new record to the patient's EHR and (b) remove an existing record from the patient's EHR;
2. add access rights and (b) remove access rights; and
3. add log entry and (b) remove log entry.

The structure for each type of transaction is similar:

- a user's covert ID;
- transaction Type 1|2|3, subtype $a|b$;
- record/entry consecutive number $j$ (for each subject of care) that is also a transaction number;
- a hash from record/entry $j$; and
- a timestamp.

### 2.5. Adding and Removing Records

The algorithm for adding record $r$ or entry $e$ to blockchain $B$ is straightforward:

1. obtain a covert ID and public blockchain encryption key $Pk_{ID}$ for a user's public ID (from central eHealth service);
2. obtain last transaction number $j$ of a user with the covert ID (from central eHealth service);
3. prepare transaction $t$ Type 1|2|3, Subtype $a$, with hash $h$ of record $r$ or entry $e$, transaction number $j = j + 1$, and current time;
4. encrypt $t$ using $Pk_{ID}$;

5. send $t$ to $B$; and
6. verify if $t$ is embedded in $B$; if not, repeat from Step 2.

When it is necessary to remove a record, it is required to add a Subtype $b$ transaction to a blockchain using the algorithm for removing a record. The algorithm is initiated by a user with a role that has removal access right. The input for the algorithm is a user's public ID, record $r_j$ (or entry $e_j$) that is intended for removal, and removal order $RO_j$. The algorithm is as follows (compare with Figure 2):

1. calculate hash $h_j$ of record $j$;
2. obtain a covert ID for a user's public ID and user's blockchain encryption key pair $(Pk_{ID}, Dk_{ID})$;
3. select all transactions $T$ with the covert ID from blockchain $B$;
4. decrypt $T$ using $Dk_{ID}$;
5. verify if a transaction from $T$ contains $h_j$. If not, return $false$;
6. find in $T$ last transaction number $j$ of a user with the covert ID;
7. prepare transaction $t$, Type $1|2|3$, Subtype $b$ with hash $h$ of record $r$ or entry $e$, transaction number $j = j + 1$, and current time;
8. encrypt $t$ using user's public blockchain encryption key $Pk_{ID}$;
9. send $t$ and $RO_j$ to $B$ (blockchain nodes only accept $t$ when $RO_j$ matches $h_j$); and
10. verify if $t$ is embedded in $B$; if not, repeat from Step 3.

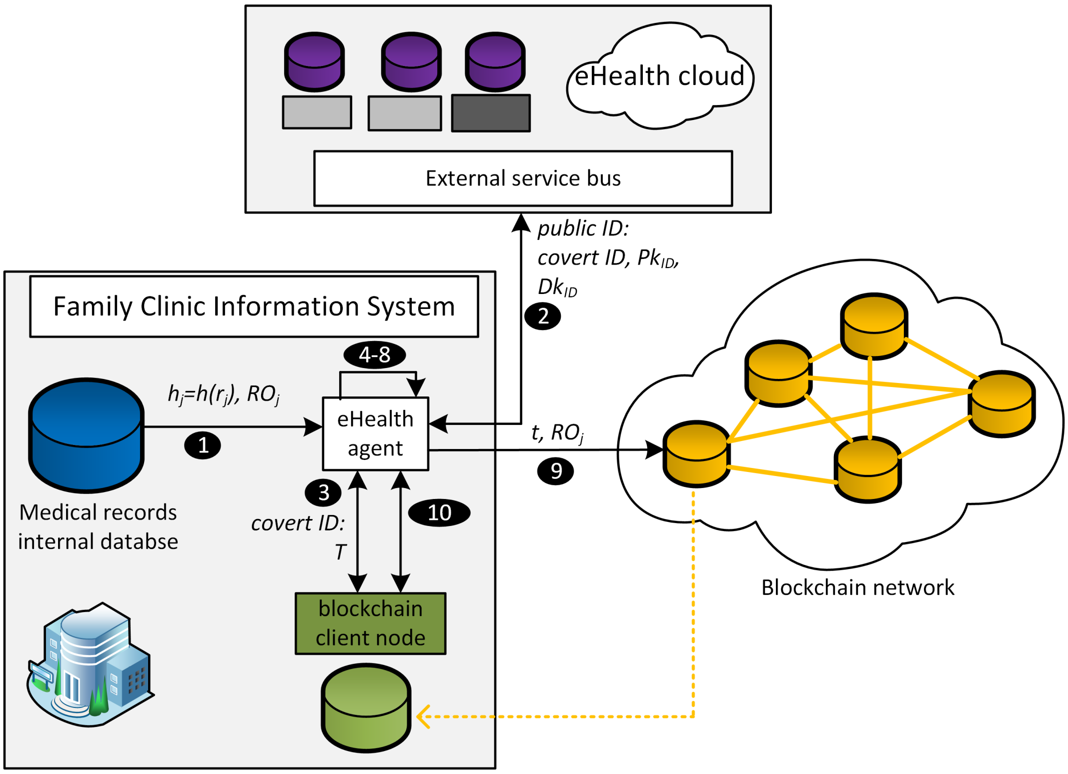

**Figure 2.** Illustration of adding Subtype *b–remove* transactions.

### 2.6. Verification

The first step in a verification procedure is to select transactions $T$ that have specified covert ID from blockchain $B$. Then, a verifier decrypts these transactions. In all cases, a verifier must have access rights to user's decryption key. The Figure 3 shows different cases of integrity verification. On the left side are visible records from a user's EHR. The middle part contains information if a hash from a given record match a hash stored in blockchain $B$ and information in which $j$ transaction in $B$ the hash of the record is stored. On the left side are depicted verification results. In BEIM it is possible to verify:

The first step in a verification procedure is to select transactions $T$ that have specified a covert ID from blockchain $B$. Then, a verifier decrypts these transactions. In all cases, a verifier must have access rights to a user's decryption key. Figure 3 shows different cases of integrity verification. On the left side are visible records from a user's EHR. The middle part contains information when a hash from a given record matches a hash stored in blockchain $B$, and information in which $j$ transaction in $B$, the record hash, is stored. On the left side, verification results are depicted. In the BEIM, it is possible to verify:

1. proof of existence of a record (transaction Type 1) or an entry (transaction Types 2 and 3). A verifier calculates a hash from a record; when a blockchain contains a transaction with that hash and the transaction creation time matches the record creation time, verification is positive;
2. EHR integrity, access-rights entries, or log-entry integrity; in this case, a verifier:
   (a) selects all entries (records) (Types 1, 2, or 3) in the EHR that have the same public ID and calculates their hashes H;
   (b) for every hash $h$ from $H$ and related entry number $j$, a verifier tries to find transaction $t$ containing hash $h' = h$ in transaction set $T$. $T$ contains selected transactions from blockchain $B$ where the cover ID corresponds to the public ID; the following situations might occur (compare with Figure 3):
      i. an item that the blockchain contains is a Type $a$ transaction with hash $h' = h$; the blockchain does not contain Type $b$ transactions with $h' = h$ (Figure 3, Case 1);
      ii. the blockchain does not contain Type a transactions with hash $h' = h$, but contains Type $a$ transactions with number $j' = j$ (Figure 3, Case 2);
      iii. the blockchain does not contain Type $a$ transactions with hash $h' = h$ and entry number $j' = j$ (Figure 3, Case 3);
      iv. the blockchain contains Type $a$ and $b$ transactions with hash $h' = h$ (Figure 3, Case 4).
   (c) When a transaction $t$ from $T$ is not matched in Step 2b, the following situations might occur:
      i. the blockchain contains Type $a$ and $b$ transactions with the same hash $h'$, but hash $h = h'$ is not present in $H$ (Figure 3, Case 5);
      ii. the blockchain contains Type $a$ transaction $t$ with $h'$ and $j'$, but there is no entry with number $j = j'$ and $h = h'$ (Figure 3, Case 6);
   (d) verifier returns *true* when only Situations 2(b)i and 2(c)i occur; otherwise, a verifier returns *false*. When 2(b)ii is met, it means that entry $j$ was modified. When 2(c)ii is met, entry $j$ was not added to a blockchain, and when 2(c)ii is met, entry $j$ is missing. Additionally, when 2(b)iv is met, it means that entry $j$ was not removed as declared, which invades privacy and means that a node misbehaves.

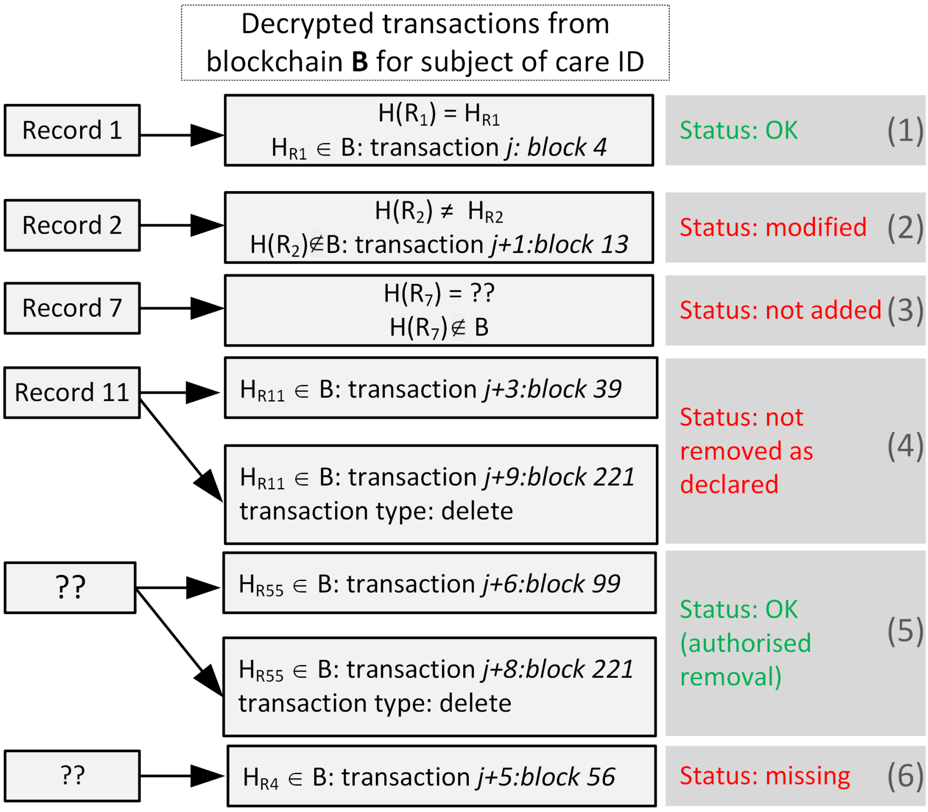
Decrypted transactions from blockchain **B** for subject of care ID
Record 1
H(R1) = HR1
HR1 ∈ B: transaction j: block 4
Status: OK (1)
Record 2
H(R2) ≠ HR2
H(R2)∉B: transaction j+1:block 13
Status: modified (2)
Record 7
H(R7) = ??
H(R7)∉ B
Status: not added (3)
Record 11
HR11 ∈ B: transaction j+3:block 39
HR11 ∈ B: transaction j+9:block 221
transaction type: delete
Status: not removed as declared (4)
??
HR55 ∈ B: transaction j+6:block 99
HR55 ∈ B: transaction j+8:block 221
transaction type: delete
Status: OK (authorised removal) (5)
??
HR4 ∈ B: transaction j+5:block 56
Status: missing (6)

**Figure 3.** Examples of different cases during integrity verification.

## 3. Discussion

The privacy of medical records is a fundamental feature in eHealth systems. In this section, the privacy impact of the proposed model is discussed, together with the impact of the consensus algorithm on integrity property. The performance of the BEIM algorithms is also discussed with the results of the verification speed test.

### 3.1. Privacy

In the BEIM, a blockchain is permissioned, i.e., it is not available publicly to everyone, and most of it is encrypted. The two privacy requirements, i.e., Anonymity to Third Parties and Confidentiality of Transaction Content, must be met. These requirements are met by usage transaction content encryption and the usage of covert identifiers.

Integrity verification requires access to a complete blockchain. More precisely, access is required to prove that a document or a log entry is missing. The BEIM uses an asymmetric key pair $(Pk_{ID}, Dk_{ID})$ for the encryption of transactions related to one user. Public key $Pk_{ID}$ is widely available for everyone, but $Dk_{ID}$ is only given to verifiers (and to entities that add Subtype *b–remove* transactions, which also verify the existence of records they remove). A verifier must have access rights to the given EHR or one of its parts to verify it. Additionally, the verifier is given blockchain access rights that include a right to access a service that translates a public ID to a covert ID, and a right to access decryption key $Dk_{ID}$. These additional rights enable the verifier to only obtain information about the number of user records and their hashes, which are a subset of information that is already available to the verifier.

The transaction's creator obtains the transaction sequence number from an eHealth system service, so they do not require right-to-access decryption key $Dk_{ID}$. Therefore, the creator does not have knowledge of the transaction content, but only has access to a right to service that translates a public ID to a covert ID. The user that removes the record and adds a Subtype *b* transaction must have the right to verify if the removed record was added to the blockchain earlier. However, this step can be omitted when earlier verification was performed by another user.

When one of the nodes becomes dishonest, it is removed by a consensus mechanism in a blockchain. However, a dishonest node still has access to a historical blockchain after exclusion, and can make it public. In such situations, transactions in a blockchain are viewed as a set of random encrypted transactions with random identification strings. An adversary from a blockchain can only learn how many entries (total medical records, access-rights entries, and log entries) were added and deleted to the EHR with a specific covert ID. However, as long as the covert ID remains secret, it is not possible to link a covert ID with a patient identifier. Optionally, in order to increase security, the covert ID can change over time. This is the same situation as in cryptocurrencies, where only a public key (a unique user-wallet ID) is known, but the relation between a public key and user identity is secret.

### 3.2. Blockchain Security

The main purpose of the blockchain in the BEIM is to provide the same copy of the ledger to each node and ensure the integrity requirements described in Section 2.2. One of the main differences between the different blockchains is the different types of used consensus algorithms. The blockchain with Proof-of-Work consensus is not used because of a number of reasons related to long-term security aspects. To begin with, the PoW consensus in practice requires a blockchain that supports cryptocurrency [24]. Additionally, besides the risk related to the long-term validity of cryptographic algorithms, several current issues in PoW blockchains, like bitcoin or Ethereum, must be considered.

The blockchain should be decentralized. However, recent observations about bitcoin show trends toward the centralization of mining pools. Centralization of mining power is a threat [28], as it increases the chances, e.g., for a 51% attack or for a selfish mining attack. Additionally, cryptocurrencies like bitcoin are not coordinated, i.e., there is no authority that might force changes (like increasing the number of transactions in a block) in the bitcoin protocol. In the long term, this might prevent a cryptocurrency from adapting to market requirements, and cause the cryptocurrency to be abandoned by users.

On top of that, the implicit assumption that PoW blockchains are trust-free is not clear because blockchain users must have a certain amount of trust in blockchain providers and software developers [29]. For example, in 2010, a software bug caused problems with bitcoin when it was updated from version 0.7–0.8 [30]. Other problems, like legal restrictions imposed by governments concerning cryptocurrencies, and enormously high [31] energy consumption used by mining that equalled 0.22% of global power consumption (in March 2018) [32,33], should also be considered.

Taking into consideration the above issues, in practice, it is only possible to use blockchains that use the PBFT [25] or Proof-of-Authority consensus. However, according to analysis using the CAP theorem [34] carried out by De Angelis et al. [25] in 2018, PoA does not provide adequate consistency guarantees in blockchains deployed over the Internet for scenarios where data integrity is important. This is mainly because current PoA algorithms can give up consistency for availability. To the contrary, PBFT keeps a blockchain consistent at the cost of availability. It is worth to mention that the two main PoA algorithms, Aura and Clique, lack appropriate documentation. Therefore, in large-scale eHealth systems deployed over the Internet (e.g., using Virtual Private Networks) for which the BEIM was designed, the PBFT should be chosen as the consensus mechanism.

To sum up, all three integrity requirements, i.e., Agreement on Transaction Validity, Tamper Evidence, and Finality, are satisfied by the PBFT algorithm; PoW and PoS do not satisfy Finality [21], and PoA does not satisfy Tamper Evidence [25]. Additionally, cryptocurrency blockchains implementing PoW and PoS are not guaranteed by any means to be available in the long term.

### 3.3. Performance

All blockchain client and authority nodes store a copy of the blockchain. The algorithms for adding, removing, and verifying are polynomial time algorithms. It might appear that the most time-consuming operations are those responsible for searching transactions with a specific covert ID. The blockchain has an ordered data structure. Therefore, it can be stored in a data format allowing for the fast searching for elements with a certain key. Transactions with a given covert ID are scattered unevenly across the blockchain. There are several solutions that can be used to speed up the retrieval of transactions with a given covert ID, starting from using an additional tree data structure with cover IDs as leaves, having pointers to the transactions, and ending at mapping block structures to relational database tables.

The integrity-verification algorithm is the most complex algorithm in the model. The time complexity of the algorithm is $O(n^2)$. The algorithm was implemented using C#, and two datasets were generated. The first test dataset consisted of 100 million random transactions that belong to one million users (100 transactions per user). The second one consisted of 100 different files with size ranging from 1 KB to 1 MB that simulates different medical documents. Hashes from those documents were inserted into the random transactions from the first set. All data transactions were stored in a Dictionary data structure that is an implementation of a hash table. The time complexity for the search operation for that data structure is $O(1)$.

The test was carried out on a test computer with an Intel Core i7-8750H @2.20 Ghz processor, 32 GB RAM, and an SSD drive. The purpose of the test was to measure how much time it would take for the integrity-verification algorithm to verify a set of 100 files (the second dataset) using transactions from the first dataset. Total execution time was 4718 ms, and almost all of that time was used for hashing documents. The time excluding hashing was only 64.3 ms, which is fast and suitable for practical purposes. The selection of 100 transactions from a 100 million transaction set took less than 0.01 ms, but such high speed required 16 GB of memory to store the transactions. In the test, RSA encryption with a 2048-bit key pair was used, and it took 64 ms to decrypt the 100 transactions. Other operations took 0.3 ms. The test results are an average after five repetitions.

Additionally, the covert ID could be encrypted with other data in a transaction, so a blockchain could be publicly stored. However, in such case, a verifier during each verification would have to decrypt a complete blockchain, and then would have to select all transactions with a specific covert ID. This is possible, but would require enormous amount of computation when a blockchain contains millions of transactions, especially when using an asymmetric encryption algorithm.

## 4. Conclusions and Future Works

A module implementing the proposed model could easily be integrated in eHealth systems based on a service-oriented architecture. Only interfaces that provide data to the module must be added. The module does not require the change of an eHealth system's internal structure. The execution time of the integrity-verification time, as the test has shown, is short. Hence, the model is suitable for practical implementation.

Security issues related to cryptocurrencies and PoA properties mean that, in practice, only PBFT blockchains (e.g., Hyperledger Fabric v0.6) can be used to secure eHealth systems. Another security aspect of blockchain usage is the privacy of the stored data. Privacy is ensured by using an encrypted transaction body and a covert identifier. A verifier must have access to all transactions in a blockchain related to a specified identifier, when they want to detect missing documents or logs. The map of covert identifiers to real identifiers is only available to entities that have an access right to the user's EHR. As long as the covert identifiers are kept secret (they should be secured using the same security mechanisms that are used to secure access to the EHR), an adversary can only obtain general anonymous statistical data.

The main novelty of the paper is the solution to the problem of transactional transparency based on a blockchain with the option to remove documents or logs. The model contains two transaction types: 'add' (Type *a*) and 'remove' (Type *b*). This mechanism allows the removal of any record, because only security-related metadata are stored in a blockchain. The only trace of a deleted record is two transactions with no information about the content or the origin of that document.

Of course, authorised nodes should accept the removal of a type of transaction from the blockchain after verifying an authorized removal order. The structure of the removal order depends on system policy, but it is usually a digitally signed hash of a transaction. The removal order should be automatically countersigned by the system (probably transparently to a user in some cases) or by an authorized supervisor. Such an approach eliminates the possibility to use this as a standard option instead of submitting a new version of a record and keeping the old one for a reference. The order is not itself stored in the blockchain.

It should be mentioned that the implementation of such a solution without a blockchain is a difficult task. The main problem solved by the blockchain is the creation of a single history of documentation that can be simultaneously created in many nodes in a distributed environment. An alternative solution based on timestamps from trusted third parties requires no more than a few central nodes. When the number of nodes rises, a synchronization protocol is required, and that leads to the introduction of the blockchain.

Future work can include implementing a proof of concept for the model, and its testing in a laboratory environment. The test has shown that the verification algorithm is suitable for practical purposes, but it is not entirely certain how the number of nodes, the latency between nodes, and different attacks impact overall node performance. The test will be carried out to verify the impact of these factors.

**Author Contributions:** Conceptualization, T.H.; methodology, T.H. and J.P.; software T.H. and J.P.; investigation, T.H. and J.P.; validation, T.H. and J.P.; project administration, T.H.; writing—original draft preparation, T.H.; writing—review and editing, T.H. and J.P.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. ISO. *ISO 13606-1:2008 Health Informatics: Electronic Health Record Communication: Part 1: Reference Model*; International Organization for Standardization: Geneva, Switzerland, 2008.
2. ISO. *ISO 18308:2011 Health Informatics: Requirements for an Electronic Health Record Architecture*; International Organization for Standardization: Geneva, Switzerland, 2011.
3. Rezaeibagha, F.; Win, K.T.; Susilo, W. A Systematic Literature Review on Security and Privacy of Electronic Health Record Systems: Technical Perspectives. *Health Inf. Manag. J.* **2015**, *44*, 23–38. [CrossRef]
4. de Carvalho Junior, M.A.; Feijó Ortolani, C.L.; Pisa, I.T. Health Information System (HIS) security standards and guidelines history and content analysis. *J. Health Inform.* **2016**, *8*, 95–102.
5. Rodrigues, J.; de la Torre, I.; Fernández, G.; López-Coronado, M. Analysis of the Security and Privacy Requirements of Cloud-Based Electronic Health Records Systems. *J. Med. Internet Res.* **2013**, *15*, e186. [CrossRef] [PubMed]
6. Sahi, A.; Lai, D.; Li, Y. Security and privacy preserving approaches in the eHealth clouds with disaster recovery plan. *Comput. Biol. Med.* **2016**, *78*, 1–8. [CrossRef] [PubMed]
7. Sahama, T.; Simpson, L.; Lane, B. Security and Privacy in eHealth: Is it possible? In Proceedings of the 2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013), Lisbon, Portugal, 9–12 October 2013; pp. 249–253. [CrossRef]
8. Aghili, S.F.; Mala, H.; Shojafar, M.; Peris-Lopez, P. LACO: Lightweight Three-Factor Authentication, Access Control and Ownership Transfer Scheme for E-Health Systems in IoT. *Future Gener. Comput. Syst.* **2019**, *96*, 410–424. [CrossRef]
9. Engelhardt, M.A. Hitching Healthcare to the Chain: An Introduction to Blockchain Technology in the Healthcare Sector. *Technol. Innov. Manag. Rev.* **2017**, *7*, 22–34. [CrossRef]
10. Liu, W.; Zhu, S.S.; Mundie, T.; Krieger, U. Advanced block-chain architecture for e-health systems. In Proceedings of the 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 12–15 October 2017; pp. 1–6.

11. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. MedRec: Using Blockchain for Medical Data Access and Permission Management. In Proceedings of the 2016 2nd International Conference on Open and Big Data (OBD), Vienna, Austria, 22–24 August 2016; pp. 25–30.
12. Liang, X.; Zhao, J.; Shetty, S.; Liu, J.; Li, D. Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–5. [CrossRef]
13. Park, Y.R.; Lee, E.; Na, W.; Park, S.; Lee, Y.; Lee, J.H. Is Blockchain Technology Suitable for Managing Personal Health Records? Mixed-Methods Study to Test Feasibility. *J. Med. Internet Res.* **2019**, *21*, e12533. [CrossRef] [PubMed]
14. Roehrs, A.; da Costa, C.A.; da Rosa Righi, R.; da Silva, V.F.; Goldim, J.R.; Schmidt, D.C. Analyzing the performance of a blockchain-based personal health record implementation. *J. Biomed. Inform.* **2019**, *92*, 103140. [CrossRef] [PubMed]
15. Ichikawa, D.; Kashiyama, M.; Ueno, T. Tamper-Resistant Mobile Health Using Blockchain Technology. *JMIR Mhealth Uhealth* **2017**, *5*, e111. [CrossRef] [PubMed]
16. Zhou, L.; Wang, L.; Sun, Y. MIStore: A Blockchain-Based Medical Insurance Storage System. *J. Med. Syst.* **2018**, *42*, 149. [CrossRef] [PubMed]
17. Wang, H.; Song, Y. Secure Cloud-Based EHR System Using Attribute-Based Cryptosystem and Blockchain. *J. Med. Syst.* **2018**, *42*, 152. [CrossRef] [PubMed]
18. Zhou, L.; Wang, L.; Ai, T.; Sun, Y. BeeKeeper 2.0: Confidential Blockchain-Enabled IoT System with Fully Homomorphic Computation. *Sensors* **2018**, *18*, 3785. [CrossRef] [PubMed]
19. Li, H.; Zhu, L.; Shen, M.; Gao, F.; Tao, X.; Liu, S. Blockchain-Based Data Preservation System for Medical Data. *J. Med. Syst.* **2018**, *42*, 141. [CrossRef] [PubMed]
20. Hyla, T.; Pejaś, J. EHealth Integrity Model Based on a Permissioned Blockchain. In Proceedings of the International Conferences on Cyber Security and Communication Systems, Melbourne, Australia, 10–12 December 2018; Agbinya, J.I., Ed.; Melbourne Institute of Technology: Melbourne, Australia, 2018; pp. 238–247.
21. Yoshihama, S.; Saito, S. Study on Integrity and Privacy Requirements of Distributed Ledger Technologies. In Proceedings of the 2018 IEEE Confs on Internet-of-Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics, Halifax, NS, Canada, 30 July–3 August 2018; pp. 1657–1664.
22. Coin Sciences Ltd. MultiChain. 2018. Available online: https://www.multichain.com/ (accessed on 10 March 2019).
23. Cachin, C. Architecture of the Hyperledger Blockchain Fabric. In Proceedings of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers, Chicago, IL, USA, 25–29 July 2016.
24. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet-of-Things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
25. Angelis, S.D.; Aniello, L.; Baldoni, R.; Lombardi, F.; Margheri, A.; Sassone, V. PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain. In Proceedings of the Italian Conference on Cyber Security, Politecnico di Milano, Milan, 6–9 February 2018.
26. Parity Technologies. Aura. 2018. Available online: https://github.com/paritytech/parity/wiki/Aura (accessed on 10 March 2019).
27. Ethereum. Clique. 2018. Available online: https://github.com/ethereum/EIPs/issues/225 (accessed on 10 March 2019).
28. Beikverdi, A.; Song, J. Trend of centralization in Bitcoin's distributed network. In Proceedings of the 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Takamatsu, Japan, 1–3 June 2015; pp. 1–6. [CrossRef]
29. Glaser, F. Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis. In Proceedings of the 50th Hawaii International Conference on System Sciences HICSS 2017, Hilton Waikoloa Village, HI, USA, 4–7 January 2017. doi:10.24251/HICSS.2017.186.
30. Park, J.H.; Park, J.H. Blockchain Security in Cloud Computing: Use Cases, Challenges, and Solutions. *Symmetry* **2017**, *9*, 164. [CrossRef]

31. Koblitz, N.; Menezes, A.J. Cryptocash, cryptocurrencies, and cryptocontracts. *Des. Codes Cryptogr.* **2016**, *78*, 87–102. [CrossRef]
32. O'Dwyer, K.J.; Malone, D. Bitcoin mining and its energy footprint. In Proceedings of the 25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, Ireland, 26–27 June 2014; pp. 280–285.
33. Digiconomist. Bitcoin Energy Consumption Index. Available online: https://digiconomist.net/bitcoin-energy-consumption (accessed on 10 March 2019).
34. Gilbert, S.; Lynch, N. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services. *SIGACT News* **2002**, *33*, 51–59. [CrossRef]