Information Sciences 485 (2019) 427–440

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain

Sheng Cao[a,b], Gexiang Zhang[c], Pengfei Liu[d], Xiaosong Zhang[d,b], Ferrante Neri[e,*]

[a] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China
[b] Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China
[c] School of Electrical Engineering, Southwest Jiaotong University, Chengdu, Sichuan 610031, China
[d] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China
[e] Institute of Artificial Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, UK

A R T I C L E   I N F O

Article history:
Received 6 August 2018
Revised 12 February 2019
Accepted 13 February 2019
Available online 14 February 2019

Keywords:
Blockchain
eHealth systems
Electronic health record

A B S T R A C T

The wide deployment of cloud-assisted electronic health (eHealth) systems has already shown great benefits in managing electronic health records (EHRs) for both medical institutions and patients. However, it also causes critical security concerns. Since once a medical institution generates and outsources the patients' EHRs to cloud servers, patients would not physically own their EHRs but the medical institution can access the EHRs as needed for diagnosing, it makes the EHRs integrity protection a formidable task, especially in the case that a medical malpractice occurs, where the medical institution may collude with the cloud server to tamper with the outsourced EHRs to hide the medical malpractice. Traditional cryptographic primitives for the purpose of data integrity protection cannot be directly adopted because they cannot ensure the security in the case of collusion between the cloud server and medical institution.

In this paper, a secure cloud-assisted eHealth system is proposed to protect outsourced EHRs from illegal modification by using the blockchain technology (blockchain-based currencies, e.g., Ethereum). The key idea is that the EHRs only can be outsourced by authenticated participants and each operation on outsourcing EHRs is integrated into the public blockchain as a transaction. Since the blockchain-based currencies provide a tamper-proofing way to conduct transactions without a central authority, the EHRs cannot be modified after the corresponding transaction is recorded into the blockchain. Therefore, given outsourced EHRs, any participant can check their integrity by checking the corresponding transaction. Security analysis and performance evaluation demonstrate that the proposed system can provide a strong security guarantee with a high efficiency.

1. Introduction

Modern technologies are steadily becoming integrating part of the health system. For example, artificial intelligence techniques are used in diagnostics in glaucoma [2] hyperactivity [35] and Parkinson's disease [19].

* Corresponding author.
E-mail address: fneri@dmu.ac.uk (F. Neri).

https://doi.org/10.1016/j.ins.2019.02.038
0020-0255/© 2019 Elsevier Inc. All rights reserved.

Among the existing technologies, electronic health (eHealth) systems, i.e. information systems which store and process patient data to enhance the efficiency of the health system has become, in the past twenty years, an important emerging technology. Compared with traditional paper-based systems, eHealth systems provide a more efficient, less error-prone, and more flexible service for both patients and medical institutions [12]. The wide deployment of eHealth systems has brought deep impact on human society [11,33]. As modern eHealth systems are data intensive [31], applying cloud computing technologies [8,9] in eHealth systems has shown great potentials [25,42] and long list of unprecedented advantages in managing the electronic health records (EHRs) in reality [27]. Such mechanism is also well known as cloud-assisted eHealth systems. In reality, cloud-assisted eHealth systems not only enable medical institutions to efficiently and flexibly manage EHRs with the aid of cloud storage services [39], but also make a great contribution to the judgement and dispute resolution in medical malpractice [44].

While cloud-assisted eHealth systems make these advantages more appealing than ever, critical privacy and security concerns in EHRs outsourcing have been raised seriously. From the perspective of EHRs owners including patients and medical institutions, the content of EHRs should not be leaked for privacy reasons, since EHRs are one of the most sensitive and personal data for them [22]. However, existing cloud service providers would not accept liability for privacy protection of EHRs against adversaries in their Service Level Agreements (SLA) and only promise to protect the privacy as much as possible [3]. Furthermore, unlike traditional EHR management paradigm, where medical institutions or patients store their EHRs locally, both medical institutions and patients would not physically own their EHRs once outsourcing EHRs to cloud servers. As such, the correctness and integrity of outsourced EHRs are being put at risk in practice [46]. We stress that the correctness and integrity not only mean that the contents of EHRs are not modified, but also mean that the time when the EHRs were generated and outsourced is also not tampered with.

Traditional cryptographic primitives that have been widely applied in cloud storage systems for the purpose of data confidentiality and integrity protection, such as public-key encryption [6,13,21,24], digital signature [7,26], and message authentication code [5], cannot be directly adopted in cloud-assisted eHealth systems, due to the following reasons.

First, different from traditional cloud storage systems, the EHRs' owner is not always the EHRs creator to generate the EHRs. Specifically, a patient's EHRs are generated and outsourced by a delegated doctor, where the EHRs would not be signed by the patient before outsourcing to reduce the communication and computation costs on the patient. Second, as the EHRs are outsourced by the doctor without the patient's participation, traditional encryption algorithms cannot be straightforward utilized. In particular, the size of EHRs may be large, and cannot be encrypted by public-key encryption schemes due to efficiency reasons. It is also challenging to agree the key of symmetric-key encryption algorithm between the doctor and the patient. Moreover, ensuring the integrity and correctness of outsourced EHRs is more challenging than ever when the doctor outsources the EHRs on behalf of the patient. Since the doctor is only trusted by the patient during the treatment period and may be malicious after the treatment period [39]. Typically, a doctor may forge, modify, or delete outsourced EHRs to cover up his mistakes in a medical malpractice.

To ensure the confidentiality of outsourced EHRs, existing scheme [44] employs a smartphone-based key agreement scheme to establish a secure channel between the patient and the doctor. However, it requires the patient to equip a powerful smartphone for diagnosing, which is not always practical.

To ensure the correctness and integrity of outsourced EHRs, existing schemes [39,44] utilize an authentication mechanism to authenticate the doctor. However, in existing schemes, there is a strong assumption that the cloud server would not collude with the doctor to tamper with the outsourced EHRs. If the doctor incentivizes the cloud server to modify the outsourced EHRs, it is hard to detect such misbehavior. Actually, compromising the cloud server is feasible for a malicious doctor, since the cloud server in existing schemes are rational entity [3,45], and thereby will deviate from the prescribed schemes if such a strategy increases its profit in the systems. To resist the collusion between the misbehavior doctor and irresponsible cloud server, a trivial solution is to introduce a trusted server to authenticate the doctor, if and only if the doctor is authenticated to the trusted server, she/he is permitted to outsource EHRs. Nonetheless, the security of such mechanism relies on the security and reliable of the trusted server, and is confronted with the single-point-of-failure problem. In fact, it is very challenging to resist the collusion between the doctor and cloud server without introducing any trusted entity.

In this paper, we propose a secure cloud-assisted eHealth system, called TP-EHR, that ensures the confidentiality of outsourced EHR and protects outsourced EHRs from illegal modification without introducing any trusted entity. The TP-EHR system is presented in its implementation as well as its computational model. The security of TP-EHR is ensured even if the doctor colludes with the cloud server. The key idea is to utilize the blockchain technique (i.e., blockchain-based currencies) [29,38,40], which provides a tamper-proofing and distributed way to conduct transactions without a central authority. In TP-EHR, the EHRs generated by a doctor is integrated into a transaction on the blockchain. The cloud server can accept the EHRs generated by the doctor, if and only if the corresponding transaction is recorded into the blockchain. TP-EHR employs a password-based key agreement mechanism to establish secure channels between patients and doctors, which is friendly to patients without requiring any additional investments on patients' devices. Compared with existing scheme, TP-EHR can resist the password guessing attacks and thereby provides a stronger security guarantee.

Specifically, the contributions of this paper are as follows:

- We analyze existing cloud-assisted eHealth systems, and point out that existing schemes cannot ensure the correctness and integrity of outsourced EHRs when the malicious doctor colludes with the cloud server to modify outsourced EHRs which are generated by the doctor himself.

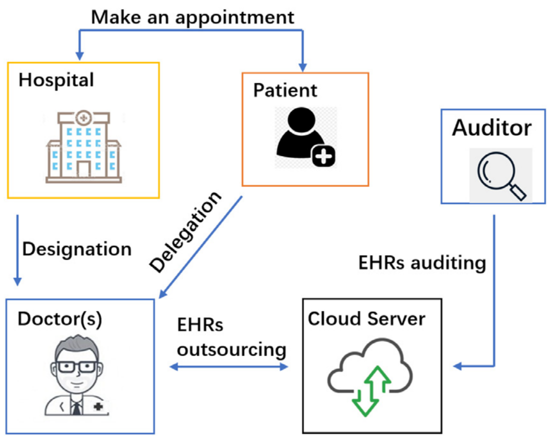

Fig. 1. System model.

- We propose a secure cloud-assisted eHealth system called TP-EHR that ensures the confidentiality, correctness, and integrity of outsourced EHRs without introducing any trusted entity, where the EHRs generated by one doctor in a treatment period are integrated into a transaction of blockchain-based currencies. TP-EHR employs a user-friendly password-based key agreement to establish secure channels between patients and doctors, which can thwart password guessing attacks without requiring additional investments on patients' devices.
- We present security analysis to demonstrate that TP-EHR guarantees the confidentiality, correctness, and integrity of outsourced EHRs. Even if the malicious doctor colludes with the cloud server, the doctor and cloud server without a large fraction of the network's computational power cannot fork the blockchain and thus cannot break the security of TP-EHR. We also conduct a comprehensive performance analysis, and show the TP-EHR is efficient and practical.

2. Related work

Cloud-assisted eHealth systems provide users including individuals and medical institutions an efficient and flexible way to manage their EHRs. Since EHRs are most personal and sensitive information for patients [18], cloud-assisted eHealth systems also suffer from challenging privacy and security threats toward outsourced EHRs [14,17,20,37,41].

To protect patients' privacy against internal adversaries (i.e., misbehavior cloud service providers) and external adversaries, EHRs are encrypted before outsourcing. Lee and Lee [23] proposed a cryptographic key management solution for protection of patients' EHRs. However, this scheme employs a trusted server to process all secret keys of patients. As a consequence, the trusted server is able to retrieve the patients' EHRs, and the privacy of patients is not well protected. Sun et al. [36] proposed a secure EHR system to protect patients' privacy without introducing any trusted entity. However, the system model of this scheme is not consistent with current cloud-assisted systems. Specifically, in these schemes, patients' EHRs are outsourced by the patients themselves, and the doctor needs to send EHRs to the patients before outsourcing. Therefore, the patients in these schemes bear heavy burden in terms of communication and computation costs.

The integrity of outsourced data has also attracted attentions in the recent literature [43,44]. These schemes mainly focus on ensuring that the outsourced data would not be lost, and the data owners generate and outsource the data to the cloud server. Nonetheless, these schemes cannot be directly adopted in eHealth systems, since the patients' EHRs are generated by delegated doctors in eHealth systems, and requiring the patients to process and outsource their EHRs after the doctors generate the EHRs would cause heavy communication and computation costs on the patients. Furthermore, the doctor is only trusted during the treatment period [39], if the malicious doctor incentivizes the cloud server to tamper with outsourced EHRs generated by himself, it is hard to detect such misbehavior. Moreover, existing schemes do not consider the timeliness of EHRs. We stress that it is also important to know when EHRs were generated in eHealth systems, since the correctness and fairness of conclusions drawn from EHRs in judgements and dispute resolutions in medical malpractices is based on the correctness and timeliness of EHRs.

The ides of using blockchain in eHealth systems has been stated at the conceptual level in the prototype described in [4,15]. A more comprehensive survey on the eHealth security can be found in [16].

3. Problem statement

3.1. Cloud-assisted eHealth systems

The system model is shown in Fig. 1. There are five different entities in TP-EHR: patients, hospital, doctor, cloud storage server, and auditor.

The procedure that a patient consults doctors in TP-EHR is illustrated as follows.

First, the patient registers with the hospital, and provides it with auxiliary information such that the hospital generates diagnosing information for the patient, where a treatment key is shared among the hospital and the patient, and the diagnosing information includes the doctor information with diagnosing time and place and some other necessary knowledge. Then the patient delegates to the doctor(s), and is diagnosed and treated at the appointed time. After the diagnosing and treating, the doctor(s) generates EHRs for the patient, and encrypts the EHRs by using the treatment key. The doctor(s) outsource(s) the ciphertexts of EHRs to the cloud storage server. Finally, the cloud storage server authenticates the doctor(s) by verifying the validity of the patient's delegation.

As described above, in a cloud-assisted eHealth system, the data (i.e., EHRs) is not generated, uploaded, and encrypted by the data owners (i.e., patients) themselves, which is different from traditional cloud storage services and introduces challenging problems on security and efficiency, see [34]. Specifically, since patients always consult doctors without heavy luggage, it is impractical to require patients to be well equipped in the system. Therefore, after generating and encrypting the EHRs, the doctor would outsource the ciphertexts to the cloud storage server without requiring patients' signatures on the EHRs. An auditor can verify the correctness and integrity of outsourced EHRs as needed.

Definition 1. TP-EHR consists of four algorithms, Setup, Appointment, Store, and Audit.

Setup. In this algorithm, the system parameters and the secret parameters are generated, and the system is initialized.

Appointment. In this algorithm, patients make appointments with the hospital. The hospital agrees a treatment key with each patient, and each patient receives an appointment information from the hospital.

Store. In this algorithm, each patient first delegates to the doctor(s). Then, the doctor(s) generates and encrypts EHRs for the patient. The cloud server authenticates the doctor(s). If the authentication succeeds, the doctor(s) outsources the ciphertexts of EHRs to the cloud server.

Audit. This algorithm enables an auditor to check the integrity and correctness of the outsourced EHRs.

3.2. Adversary model

In the adversary model, we will consider threats from two different angles: the external adversaries and internal adversaries.

3.2.1. External adversaries

External adversaries target to impersonate a patient to make an appointment with the hospital. Specifically, in existing eHealth systems, an external adversary always attempts to request for multiple tokens for diagnosing, and resells the tokens for profits. Most of existing eHealth systems utilize a password-based authentication mechanism to authenticate users. As such, external adversaries may perform keyword guessing attack to break the security of the eHealth systems.

3.2.2. Internal adversaries

- Rational cloud server. The cloud server is a rational entity, the same assumption can be found in [3,44]. By rational, we mean that the cloud server would only deviate from the prescribed scheme if such a strategy would bring benefits for it.
- Semi-trusted doctors. Doctors are semi-trusted entity, which follows existing works [44]. By semi-trusted, we mean that the doctor would be honest during the diagnosing period (i.e., the delegation time). However, after the diagnosing period, the doctor would perform the following attacks.
  1. The adversarial doctor, who may collude with a malicious patient and the cloud storage server, outsources forged EHRs. After the diagnosing period, the doctor may outsource forged EHRs to the storage server to conceal his mistake in medical malpractice.
  2. The adversarial doctor may violate the confidentiality of outsourced EHRs. The doctor may collude with the malicious cloud server to obtain EHRs generated by other doctors, and learn their contents.

In TP-EHR, we assume doctors are computer-aided, which means that doctors are equipped with computers so as to have adequate computation, communication, and storage resources to generate and outsource EHRs. Furthermore, the semi-trusted hospital is considered as a semi-trusted doctor in the adversary model for the sake of brevity.

Within the adversary model, TP-EHR should satisfy the following security requirements:

- Confidentiality. The contents of outsourced EHRs cannot be recovered by any adversary.
- Resistance against EHR forgery. The outsourced EHRs cannot be replaced by forged ones generated by any adversary.
- Resistance against EHR modification. Once EHRs are outsourced and accepted by the cloud server, any adversary cannot substitute existing EHRs for them, and cannot delete them. Furthermore, the time when the EHRs are generated should be securely recorded and cannot be modified.
- Resistance against impersonation attacks performed by external adversaries. External adversaries cannot impersonate a patient to make an appointment with the hospital by guessing the patient's password.

3.3. Design goals

In this paper, we target the security of EHRs in cloud-assisted eHealth systems, where there exist three challenges:

1. How to resist the collusion between malicious doctors and misbehavior doctors. Existing cloud-assisted eHealth systems bear a strong assumption that the cloud server would not collude with doctors to modify outsourced EHRs. However, a misbehavior doctor may incentivize the cloud server to modify outsourced EHRs generated by himself to conceal his mistakes in a medical malpractice. To ensure the security of outsourced EHRs, outsourced EHRs should not be modified, forged, and deleted by the misbehavior doctor, even if he colludes with the cloud server and the target EHRs are generated by the doctor himself.
2. How to securely time-stamp the EHRs before outsourcing. In reality, a patient's EHRs may be generated by multiple doctors, which is well known as the group consultation. As such, it is very challenging to determine the time when each part of EHRs from each doctor is generated. Therefore, how to securely time-stamp the outsourced EHRs should be well considered.
3. How to securely authenticate the patient. Existing cloud-assisted eHealth systems employ a password-based authentication mechanism to authenticate patients, since such authentication mechanism is friendly to patients, and is easily to be deployed. However, such mechanism is vulnerable to password guessing attacks, due to the low entropy and inherent vulnerability of human-memorisable passwords. To enhance the security, the cloud-assisted eHealth system should resist the password guessing attacks.

To enable secure outsourced EHRs for cloud-assisted eHealth systems under the aforementioned model, TP-EHR should achieve the following objectives.

- Functionality: TP-EHR should provide a reliable and privacy-preserving way to manage patients' EHRs, and can be applied in existing cloud-assisted eHealth systems without changing its architecture.
- Security: The security requirements under the adversary model presented in Section 3.2 should be achieved.
- Efficiency: TP-EHR should be as efficient as possible in terms of computational complexity, communication complexity, and storage costs.

4. Preliminaries

4.1. Notations, conventions, and basic theory

For two bit-strings x and y, we denote by $x||y$ their concatenation. We use $E()$ to denote symmetric encryption.

Bilinear maps

Let G and G_T be two multiplicative groups with the same order p. A bilinear map e: $G \times G \rightarrow G_T$ has the following properties, see [10,30,32]:

- Bilinearity. $e(g^a, q^b) = e(g, q)^{ab}$ for all $g, q \in G$, $a, b \in Z_p^*$.
- Non-degeneracy. For $g, q \in G$ and $g \neq q$, $e(g, q) \neq 1$.
- There exists an efficient computable algorithm to compute e.

4.2. Blockchain

Blockchains have been utilized to enhance the data security for cloud storage services.

A blockchain is a linear collection of data elements called block: all blocks are linked to form a chain and secured by cryptographic primitive. The blockchain is maintained by multiple nodes and new blocks are continuously added to the blockchain without requiring nodes to trust each other. If and only if a considerable majority of nodes is honest, the security of blockchain is guaranteed [29].

Typically, each block contains a hash pointer that points to its previous block, a timestamp, and transaction data. The block can be chained to the blockchain, only if the validity of its transaction data is verified by a majority of nodes [38,40].

The blockchain technique can be generally classified into two types: private blockchain and public blockchain.

In the private blockchain (including consortium blockchain), the nodes that maintain the blockchain are employed and authorized by a blockchain owner or a consortium comprising multiple participants who jointly own the blockchain but do not trusted each other. The key difference between the private blockchain and existing techniques (e.g., distributed backup systems) is that once a block is chained to the blockchain, as long as the majority of nodes is inaccessible to the adversaries, this block cannot be removed or modified, even if the adversary is the blockchain manager himself.

In the public blockchain, anyone can become a node to maintain the blockchain and can join or leave the blockchain system without getting permission from a centralized or distributed authority. The most prominent manifestation of public blockchain is blockchain-based cryptocurrencies, such as Bitcoin and Ethereum [29,40]. In such systems, the public blockchain serves as an open, distributed, and tamper-proofing ledger to account for the ownership of the value tokens. Transferring value tokens between two users can be considered as a state transition system, where the public ledger

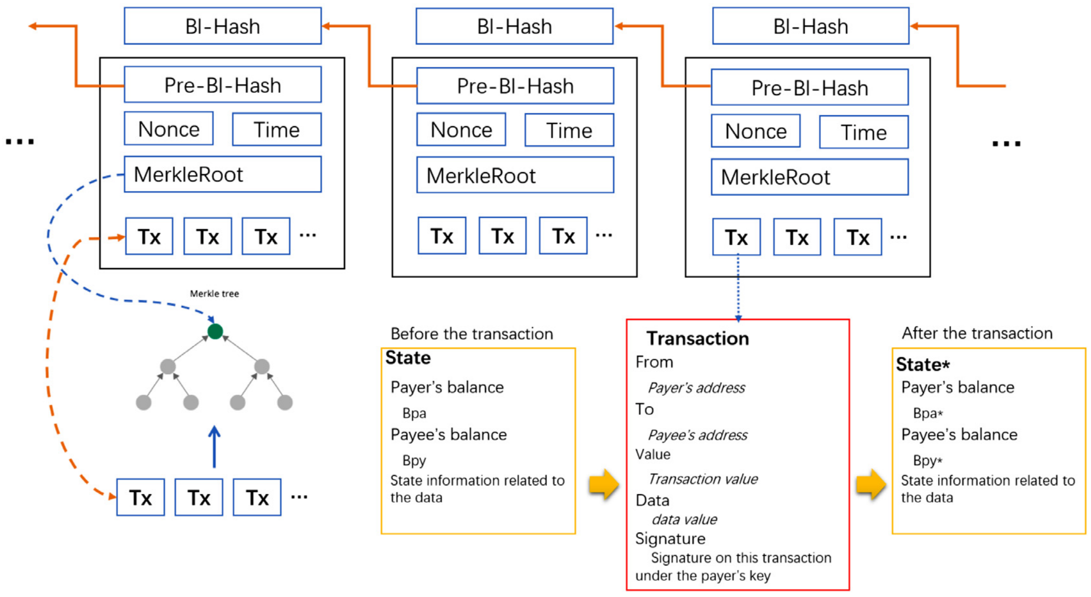

Fig. 2. A simplified Ethereum blockchain.

recorded in the blockchain reflects the ownership status of existing value tokens, and a state transition function takes a state and a transaction as input, outputs a new state which is the result, and then updates the ledger.

A simplified Ethereum blockchain is shown in Fig. 2, where Bl – Hash denotes the hash value of current block, Pre – Bl – Hash denotes the hash value of the previous block, Nonce denotes the solution of the proofs of work (PoW) puzzle, Time denotes the timestamp of the block, Tx denotes the transaction, and all transactions are authenticated by using Merkle hash tree with the MerkleRoot as its root value.

In Ethereum, the state is made up of objects called "account". In general, there are two types of accounts in Ethereum: externally owned accounts and contract accounts. Externally owned accounts are controlled by private keys and can conduct a transaction. Contract accounts are controlled by their contract code. When the payer transfers Ethers from her/his (externally owned) account to the payee's (externally owned) account, if the transaction is recorded into the blockchain, the balances of these two accounts are updated. Here, the data value of the transaction can be set to any binary data the payer chooses. If a transaction is recorded into the blockchain, it cannot be removed and modified due to the security of Ethereum.

5. The proposed TP-EHR

5.1. Overview

Since patients always consult doctors without heavy luggage, TP-EHR should be as efficient as possible on the patient side and should enable patients to preprocess the costly computation.

In TP-EHR, each patient first makes an appointment with the hospital, and obtains a treatment key for diagnosing. With the treatment key, a secure channel between the patient and the hospital as well as the designated doctors is established. This process is based on a password-based authentication mechanism with resistance against password guessing attacks, which ensures that the security of TP-EHR cannot be broken by external adversaries who guess target patient's password.

Before the treatment time, the patient generates warrants to delegate to doctors. The warrant indicates the identities of delegated doctors as well as their diagnosing period and other auxiliary information. During the treatment time, doctors generate EHRs for the patient. We consider two different cases in TP-EHR. One is the single doctor case, where the patient is treated by only one doctor and thereby EHRs are generated by the doctor; another one is the multi-doctor case where the patient is treated by a group of doctors, EHRs are successively generated by these doctors, and each doctor generates EHRs based on the ones generated by the previous doctor.

For the single doctor case, the doctor should be responsible for the EHRs generated by herself/himself. The EHRs generated within one treatment period for the patient are integrated into a transaction in Ethereum. For the multi-doctor case, the doctor not only should be responsible for the EHRs generated by herself/himself, but also should be responsible for the EHRs generated by the previous doctor. As such, each part of EHRs generated by one doctor is integrated into a transaction of Ethereum to ensure the security of TP-EHR.

We further consider the timeliness of EHRs, since it is more important to know when EHRs were generated than what they were. Since EHRs generated by a doctor correspond to a transaction in Ethereum, anyone can learn the time when EHRs are generated is to extract the timestamp of the block in Ethereum that includes the transaction.

The security of TP-EHR is mainly constructed on the security of Ethereum blockchain, an adversary without a large fraction of the network's computational power cannot fork Ethereum and thus cannot break the security of TP-EHR. Even if the cloud server colludes with a malicious doctor to tamper with outsourced EHRs, it cannot modify the corresponding transaction recorded into the Ethereum blockchain.

5.2. Construction of TP-EHR

A patient $\mathcal{P}$ with identity $ID_{\mathcal{P}}$, a cloud storage server $\mathcal{S}$, a set of doctors $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_\chi$ with identities $ID_{\mathcal{D}_1}, ID_{\mathcal{D}_2}, \ldots, ID_{\mathcal{D}_\chi}$, a hospital $\mathcal{H}$ with identity $ID_{\mathcal{H}}$, and an auditor $\mathcal{A}$ are involved in TP-EHR.

5.2.1. Setup

With the security parameter ℓ, the system parameter $SP = \{p, g, G, G_T, e, H, h, H_1\}$ are determined, where G and G_T are multiplicative groups with the same prime order p, g is the generator of G, e: $G \times G \rightarrow G_T$, h, H: $\{0, 1\}^* \rightarrow G$, H_1: $\{0, 1\}^* \rightarrow Z_p$. Each doctor creates an externally owned account in Ethereum and publishes it to others, where the account is special-purpose. The cloud storage server also creates an externally owned account in Ethereum and sends it to all doctors and the auditor.

For the patient $\mathcal{P}$ with identifier $ID_{\mathcal{P}}$, $\mathcal{H}$ assigns a human-memorisable password $pw_{\mathcal{P}}$ to her/him. The patient $\mathcal{P}$ also has a secret key $\alpha_{\mathcal{P}}$, and the corresponding public key $Q_{\mathcal{P}} = g^{\alpha_{\mathcal{P}}}$.

5.2.2. Appointment

In this algorithm, $\mathcal{P}$ obtains the appointment information protected under a treatment key $tk_{\mathcal{P}}$ as follows:

- $\mathcal{P}$ randomly chooses $x \in Z_p$, computes $X = g^x$, $X^* = X \cdot (h(ID_{\mathcal{P}}))^{pw_{\mathcal{P}}}$, and sends X^* to $\mathcal{H}$.
- $\mathcal{H}$ randomly chooses $y \in Z_p$, computes $Y = g^y$, $Y^* = Y \cdot (h(ID_{\mathcal{H}}))^{pw_{\mathcal{H}}}$, and sends Y^* to $\mathcal{P}$.
- $\mathcal{P}$ computes $K_{\mathcal{P}} = (Y^*/(h(ID_{\mathcal{H}}))^{pw_{\mathcal{P}}})^x$ and sets $tk_{\mathcal{P}} = H_1(ID_{\mathcal{P}}, ID_{\mathcal{H}}, X^*, Y^*, pw_{\mathcal{P}}, K_{\mathcal{P}})$.
- $\mathcal{H}$ computes $K_{\mathcal{H}} = (X^*/(h(ID_{\mathcal{P}}))^{PW_{\mathcal{P}}})^y$ and sets $tk_{\mathcal{P}} = H_1(ID_{\mathcal{P}}, ID_{\mathcal{H}}, X^*, Y^*, pw_{\mathcal{P}}, K_{\mathcal{P}})$.
- $\mathcal{P}$ makes an appointment with $\mathcal{H}$.
- $\mathcal{H}$ appoints a set of doctors $\{\mathcal{D}_i\}(i \in I)$ for $\mathcal{P}$, where I denotes the set of appointed doctors' indexes.
- $\mathcal{H}$ sends the appointment information (protected under $tk_{\mathcal{P}}$) to $\mathcal{P}$, and sends $tk_{\mathcal{P}}$ to $\{\mathcal{D}_i\}(i \in I)$.
- $\mathcal{P}$ decrypts and parses the appointment information to get $ID_{\mathcal{D}_i}$ for $i \in I$, the valid period $TimePeriod_{\mathcal{P}}$, and some auxiliary information $Aux_{\mathcal{P}}$.

5.2.3. Store

There are two cases in this algorithm, the one includes the single doctor who is denoted by $\mathcal{D}_1$ without loss of generality; the other one includes multiple doctors.

Case 1: There is only a single doctor $\mathcal{D}_1$.

- $\mathcal{P}$ computes a warrant $w_{\mathcal{P}}$ to delegate $\mathcal{D}_1$ to generate EHRs as:

$$wa_{\mathcal{P}} = ID_{\mathcal{P}} || ID_{\mathcal{D}_1} || TimePeriod_{\mathcal{P}} || Aux_{\mathcal{P}},$$
$$w_{\mathcal{P}} = \alpha_{\mathcal{P}} \cdot H(wa_{\mathcal{P}}).$$

- $\mathcal{D}_1$ generates an EHR M for $\mathcal{P}$.
- $\mathcal{D}_1$ encrypts M as:

$$C = E(tk_{\mathcal{P}}, M || wa_{\mathcal{P}} || w_{\mathcal{P}}),$$

  where $E()$ is a secure symmetric encryption algorithm, e.g., AES.
- Based on the current time t, $\mathcal{D}_1$ extracts the hash value of the block that is the latest one to be attached into the blockchain. This hash value is denoted by $Bhash_t$.
- $\mathcal{D}_1$ creates a transaction $Tx_{\mathcal{D}_1}$ shown in Fig. 3, where she/he transfers the service charge to the cloud storage $\mathcal{S}$'s account, and sets the data value of the transaction as:

$$Bhash_t || h(ID_{\mathcal{P}}) || h(C || wa_{\mathcal{P}} || w_{\mathcal{P}}).$$

- $\mathcal{D}_1$ sends $(Bhash_t, C, wa_{\mathcal{P}}, w_{\mathcal{P}})$ to $\mathcal{S}$.
- $\mathcal{S}$ verifies that the service charge has been received, and checks the validity of $TimePeriod_{\mathcal{P}}$ and $Bhash_t$ by checking the following equation:

$$e(w_{\mathcal{P}}, g) = e(H(wa_{\mathcal{P}}), Q_{\mathcal{P}}).$$

  If the checking passes, $\mathcal{S}$ accepts $(C, wa_{\mathcal{P}}, w_{\mathcal{P}})$.

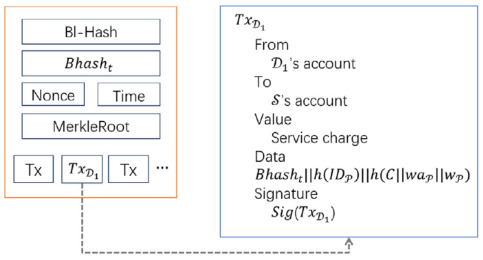

Fig. 3. Transaction on the Ethereum blockchain of single doctor case.

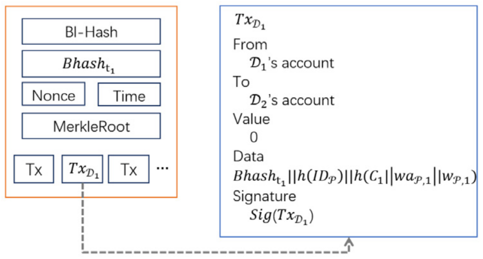
Bl-Hash
$Bhash_{t_1}$
Nonce
Time
MerkleRoot
Tx
$Tx_{\mathcal{D}_1}$
Tx
···
$Tx_{\mathcal{D}_1}$
From
$\mathcal{D}_1$'s account
To
$\mathcal{D}_2$'s account
Value
0
Data
$Bhash_{t_1}||h(ID_\mathcal{P})||h(C_1||wa_{\mathcal{P},1}||w_{\mathcal{P},1})$
Signature
$Sig(Tx_{\mathcal{D}_1})$

Fig. 4. Transaction on the Ethereum blockchain of multi-doctor case: the transaction is conducted by the first doctor.

Case 2: There is a set of doctors $\{\mathcal{D}_i\}(i \in I)$. Multiple doctors generate the EHRs for $\mathcal{P}$ in turn. Here, we assume that $\mathcal{D}_1$ is the first doctor to generate the EHRs and $\mathcal{D}_{|I|}$ is the last one to generate the EHRs.

- For each doctor $\{\mathcal{D}_i\}(i \in I)$, $\mathcal{P}$ computes a warrant $w_{\mathcal{D},i}$ to delegate $\mathcal{D}_i$ to generate EHRs as:

$$wa_{\mathcal{P},i} = ID_\mathcal{P}||ID_{\mathcal{D}_i}||TimePeriod_{\mathcal{P},i}||Aux_{\mathcal{P},i},$$
$$w_{\mathcal{P},i} = \alpha_\mathcal{P} \cdot H(wa_{\mathcal{P},i}).$$

- $\mathcal{P}$ sends warrant $(wa_{\mathcal{P},i}, w_{\mathcal{P},i})$ to $\mathcal{D}_i$, $i = 1, 2, \ldots, |I|$.

For the first doctor $\mathcal{D}_1$, she/he performs as follows:

- $\mathcal{D}_1$ generates an EHR M_1 for $\mathcal{P}$.
- $\mathcal{D}_1$ encrypts M_1 as:

$$C_1 = E(tk_\mathcal{P}, M_1||wa_{\mathcal{P},1}||w_{\mathcal{P},1}).$$

- Based on the current time t_1, $\mathcal{D}_1$ extracts the hash value of the block that is the latest one to be attached into the blockchain. This hash value is denoted by $Bhash_{t_1}$.
- $\mathcal{D}_1$ creates a transaction $Tx_(\mathcal{D}_1)$ shown in Fig. 4, where she/he transfers 0 service charge to the next doctor $\mathcal{D}_2$'s account, and sets the data value as:

$$Bhash_{t_1}||h(ID_\mathcal{P})||h(C_1||wa_{\mathcal{P},1}||w_{\mathcal{P},1}).$$

- $\mathcal{D}_1$ sends $(Bhash_{t_1}, C_1, wa_{\mathcal{P},1}, w_{\mathcal{P},1})$ to $\mathcal{D}_2$.

For the doctor $\mathcal{D}_i$, $i = 2, 3, \ldots, |I| - 1$, she/he performs as follows:

- $\mathcal{D}_i$ verifies the validity of $(Bhash_{t_{i-1}}, C_{i-1}, wa_{\mathcal{P},i-1}, w_{\mathcal{P},i-1})$ received from $\mathcal{D}_{i-1}$ by checking the following equation:

$$e(w_{\mathcal{P},i-1}, g) = e(H(wa_{\mathcal{P},i-1}), Q_\mathcal{P}).$$

- $\mathcal{D}_i$ decrypts C_{i-1} to obtain $\{M_1, \ldots, M_{i-1}\}$.
- $\mathcal{D}_i$ generates an EHR M_i for $\mathcal{P}$.
- $\mathcal{D}_i$ encrypts M_i as:

$$C_i = E(tk_\mathcal{P}, M_1||\cdots||M_i||wa_{\mathcal{P},i}||w_{\mathcal{P},i}).$$

- Based on the current time t_i, $\mathcal{D}_i$ extracts the hash value of the block that is the latest one to be attached into the blockchain. This hash value is denoted by $Bhash_{t_i}$.
- $\mathcal{D}_i$ creates a transaction $Tx_(\mathcal{D}_i)$ shown in Fig. 5, where she/he transfers 0 service charge to the next doctor $\mathcal{D}_{i+1}$'s account, and sets the data value as:

$$Bhash_{t_i}||h(ID_\mathcal{P})||h(C_i||wa_{\mathcal{P},i}||w_{\mathcal{P},i}).$$

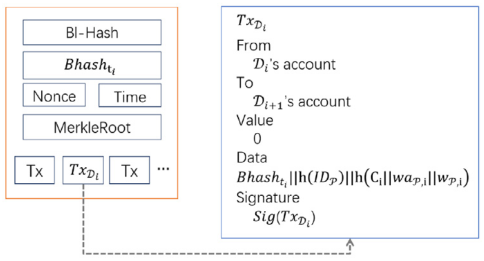
Bl-Hash
$Bhash_{t_i}$
Nonce
Time
MerkleRoot
Tx
$Tx_{\mathcal{D}_i}$
Tx
...
$Tx_{\mathcal{D}_i}$
From
$\mathcal{D}_i$'s account
To
$\mathcal{D}_{i+1}$'s account
Value
0
Data
$Bhash_{t_i}||h(ID_{\mathcal{P}})||h(C_i||wa_{\mathcal{P},i}||w_{\mathcal{P},i})$
Signature
$Sig(Tx_{\mathcal{D}_i})$

Fig. 5. Transaction on the Ethereum blockchain of multi-doctor case: the transaction is conducted by the *i*th doctor.

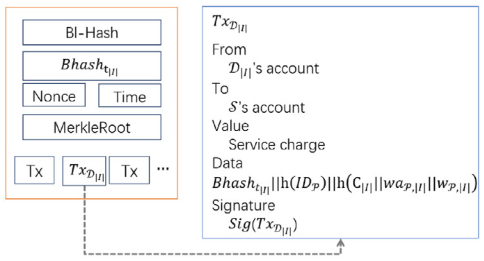

Fig. 6. Transaction on the Ethereum blockchain of multi-doctor case: the transaction is conducted by the last doctor.

- $\mathcal{D}_i$ sends $(Bhash_{t_i}, C_i, wa_{\mathcal{P},i}, w_{\mathcal{P},i})$ to $\mathcal{D}_{i+1}$.

  For the doctor $\mathcal{D}_{|I|}$, she/he performs as follows:

- $\mathcal{D}_{|I|}$ verifies the validity of ($Bhash_{t_{|I|-1}}$, $C_{|I|-1}$, $wa_{\mathcal{P},|I|-1}$, $w_{\mathcal{P},|I|-1}$) received from $mathcalD_{|I|-1}$ by checking the following equation:

$$e(w_{\mathcal{P},|I|-1}, g) = e(H(wa_{\mathcal{P},|I|-1}), Q_{\mathcal{P}}).$$

- $\mathcal{D}_{|I|}$ decrypts $C_{|I|-1}$ to obtain $\{M_1, \ldots, M_{|I|-1}\}$.
- $\mathcal{D}_{|I|}$ generates an EHR $M_{|I|}$ for $\mathcal{P}$.
- $\mathcal{D}_{|I|}$ encrypts $M_{|I|}$ as:

$$C_{|I|} = E(tk_{\mathcal{P}}, M_1|| \cdots ||M_{|I|}||wa_{\mathcal{P},|I|}||w_{\mathcal{P},|I|}).$$

- Based on the current time $t_{|I|}$, $\mathcal{D}_{|I|}$ extracts the hash value of the block that is the latest one to be attached into the blockchain. This hash value is denoted by $Bhash_{t_{|I|}}$.
- $\mathcal{D}_{|I|}$ creates a transaction $Tx_{\mathcal{D}_{|I|}}$ shown in Fig. 6, where she/he transfers the service charge to the cloud storage $\mathcal{S}$'s account, and sets the data value as:

$$Bhash_{t_{|I|}}||h(ID_{\mathcal{P}})||h(C_{|I|}||wa_{\mathcal{P},|I|}||w_{\mathcal{P},|I|}).$$

- $\mathcal{D}_{|I|}$ sends $(Bhash_{t_{|I|}}, C_{|I|}, wa_{\mathcal{P},|I|}, w_{\mathcal{P},|I|})$ to $\mathcal{S}$.
- $\mathcal{S}$ verifies that the service charge has been received, then checks the validity of $TimePeriod_{\mathcal{P}}$ and $Bhash_{t_{|I|}}$. Finally, $\mathcal{S}$ checks the following equation:

$$e(w_{\mathcal{P},|I|}, g) = e(H(wa_{\mathcal{P},|I|}), Q_{\mathcal{P}}).$$

  If the checking passes, $\mathcal{S}$ accepts ($C_{|I|}$, $wa_{\mathcal{P},|I|}$, $w_{\mathcal{P},|I|}$).

5.2.4. Audit

Given the EHR ($ID_{\mathcal{P}}$, C, $wa_{\mathcal{P}}$, $w_{\mathcal{P}}$), the auditor $\mathcal{A}$ is able to check the correctness and timeliness as follows:

- Pares the EHR and obtain $(C, wa_{\mathcal{P}}, w_{\mathcal{P}})$.
- Extract the corresponding transaction from the Ethereum blockchain and acquire the corresponding account information.
- Verify the number of created transactions matches the number of recorded EHRs, if the verification fails, reject.
- Verify the validity of $w_{\mathcal{P}}$, if the verification fails, reject.
- Verify the timeliness of the EHR by checking the transaction time, if the verification fails, reject, where the transaction time is derived from the block.
- Compute $Bhash_t||h(ID_{\mathcal{P}})||h(C||wa_{\mathcal{P}}||w_{\mathcal{P}})$ and check whether it equals to the transaction information.

If all above verification passes, the correctness and timeliness of the EHR is guaranteed.

6. Security analysis

6.1. TP-EHR is secure against EHR forgery attacks

For the single doctor case, TP-EHR can resist forgery attacks performed by any adversaries. If an adversary forges EHRs, he cannot modify the corresponding transaction in the blockchain. As such, when the auditor checks the correctness of outsourced EHRs, the forged EHRs cannot pass the checking, since the blockchain ensures that recorded transactions cannot be modified.

For the case of multiple doctors, we prove that TP-EHR achieves resistance against EHR forgery attacks in two aspects.

First, the malicious storage server cannot forge outsourced EHRs. In TP-EHR, the EHRs are encrypted before outsourcing. Due to the security of encryption algorithm, the malicious storage server cannot forge valid EHRs without valid encryption/decryption keys.

Second, a semi-trusted doctor cannot forge outsourced EHRs, even if the EHRs are generated by the doctor himself. If a semi-trusted doctor attempts to forge EHRs, he can perform two attacks:

1. A doctor generates EHRs and outsources the EHRs to the cloud storage but convinces the later that the EHRs are generated by other doctors.
2. A doctor colludes with the cloud server, to replace existing EHRs with new ones.

For attack 1. Note that although in TP-EHR, we do not require patients to authenticate their EHRs before outsourcing, patients are required to authenticate their doctor before the diagnosing, where a patient generates a warrant to delegate a doctor, and the warrant includes the patient identity, doctor identity, the valid period, and other auxiliary information. The warrant is constructed on the secure signature scheme [7] and thereby is existentially unforgeable. Therefore, it is computationally infeasible to perform attack 1 for the doctor.

In regarding to attack 2, each part of EHRs generated by one doctor is integrated into a transaction in Ethereum. When a doctor attempts to replace existing EHRs with new ones generated by him, the only thing he can do is to fork the Ethereum blockchain and enable the blockchain with the transaction corresponding to newly generated EHRs to be accepted by a majority of nodes. Therefore, the security against EHR forgery at this point is based on the security of underlying blockchain. Even if the doctor colludes a malicious patient and the cloud storage server, he cannot achieve the above goal by utilizing other strategies.

6.2. TP-EHR is secure against EHR modification attacks

As discussed before, the outsourced EHRs in TP-EHR cannot be forged by adversaries. We need to further consider EHR modification attacks that adversaries substitute existing EHRs for target ones, such that cover up the target EHRs.

In TP-EHR, for the single doctor case, we utilize two security mechanisms to resist EHR modification attacks. The first one is the secure delegation mechanism, where the patient first generates a warrant to delegate the doctor. The warrant is built on a secure signature algorithm [7] and cannot be forged. The second one is the blockchain-based tamper-resist mechanism, where the EHRs generated by the doctor as well as the corresponding warrant are integrated into a transaction in Ethereum blockchain. As such, if the adversary cannot break the security of Ethereum blockchain, he cannot break the security of TP-EHR by performing EHRs modification attacks. Again, this also holds even if the adversary colludes with a malicious patient and the cloud storage server.

The security of multiple doctors' case is extended from the single doctor one, where TP-EHR needs to ensure that the chronological order that multiple doctors generate EHRs cannot be modified. This essentially corresponds to the timeliness of outsourced EHRs which we will elaborate on it in Section 6.4.

6.3. TP-EHR is secure against impersonation attacks

In TP-EHR, impersonation attacks performed by external adversaries are considered, where an adversary extracts the treatment key by performing password guessing attacks. Once the adversary succeeds, he can impersonate the victim to make an appointment. As a consequence, the adversary can further perform various attacks, such as Distributed Denial of Service (DDoS) to corrupt TP-EHR.

TP-EHR can resist impersonation attacks due to the security of the treatment key request. The treatment key request in TP-EHR is based on a password-based authentication mechanism which resists password guessing attacks. Specifically, in the Appointment algorithm, if the password used by the patient is the same as the one stored in the hospital, the patient and the hospital would jointly execute a key agreement protocol, and a treatment key is shared between them; otherwise, the patient would obtain a random key, and cannot perform the following algorithms [1,28]. As such, TP-EHR resists password guessing attacks and thereby is secure against impersonation attacks.

6.4. TP-EHR guarantees the timeliness of EHRs

The timeliness of EHRs in TP-EHR is twofold.

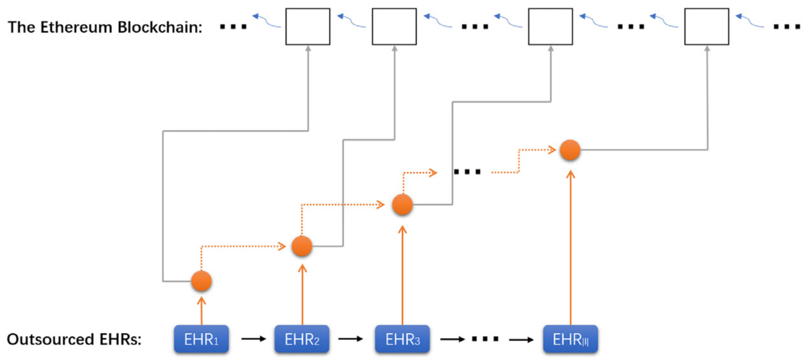

Fig. 7. Block-aided authenticated EHR chain.

First, the timeliness of EHRs is reflected in the corresponding transaction time in the blockchain. Specifically, in TP-EHR, each EHR is relative to one transaction in the blockchain. When the transaction is recorded in to the blockchain, anyone is able to efficiently extract the time when the EHR was generated from the transaction time.

Second, for a patient, the timeliness of her/his EHRs from multiple doctors is also reflected in the chronological order, where these EHRs essentially form an authentication structure with the aid of the blockchain, which is shown in Fig. 7, where gray line denotes that the EHR is recorded into the corresponding block, the dashed orange line denotes that the prior EHR is integrated into the later one. In fact, if a patient's EHRs are generated by multiple doctors, the outsourced EHRs can actually reflect identities of the doctors in a chronological order.

According to the above analysis, we can see that the timeliness of outsourced EHRs in TP-EHR are based on the observation in the underlying blockchain: the outsourced EHRs are as hard to fork as the underlying blockchain such that an adversary without a large fraction of the network's mining hashrate cannot fork Ethereum and thus cannot break the timeliness of outsourced EHRs in TP-EHR.

6.5. On the necessity of blockchain in TP-EHR

Without the blockchain, TP-EHR is vulnerable to EHR forgery and removal attacks without detection.

In particular, if a doctor attempts to forge EHRs that have been outsourced to the cloud server to conceal his mistake in medical malpractices, he can incentivize the cloud server to collude with him, and the cloud server can arbitrarily forge and remove existing EHRs from its storage. Note that in reality, due to efficiency reasons, the patients would not be required to sign the EHRs generated from doctors. In fact, the EHR forgery and removal attacks can also be performed without detection, even if EHRs signed by the patients.

To ensure the security of outsourced EHRs, most of existing schemes assume that the cloud server would not collude with the doctors to forge/remove existing EHRs [27,36]. Under this assumption, there is an authentication mechanism between the cloud server and doctors in these schemes, which protects outsourced EHRs from illegal modification.

In TP-EHR, each EHR generated by a doctor is integrated into a transaction of the underlying blockchain. As long as the security of the blockchain remains tamper-resistant to adversaries, we ensure the correctness and integrity of outsourced EHRs in TP-EHR. This would not introduce any additional security mechanisms, strong assumptions, and trusted entities. Therefore, the blockchain technology plays a key role in TP-EHR to ensure the security.

7. Performance evaluation

We evaluate the performance of TP-EHR in terms of communication and computational overhead. We conduct the experiments on a computer with Window 10 system, an Inter Core 2 i5 CPU, and 8 GB DDR 3 of RAM. We utilize C language and MIRACL Library to implement TP-EHR. The security level is selected to be 80 bits.

7.1. Communication overhead

In the experiment, we assume the appointment information is a string with 512 bits. We would not show the communication between the doctors and the cloud server, since it depends on the size of patients' EHRs.

For the hospital, the communication costs include two parts. One is to interact with patients for appointment, another one is to designate doctors.

For the patient, her/his communication costs include two parts. One is to make an appointment with the hospital, another one is to delegate doctors.

In Figs. 8 and 9, we show the communication overhead on the patient, hospital, and doctor, which shows that entities in TP-EHR would not bear heavy communication costs.

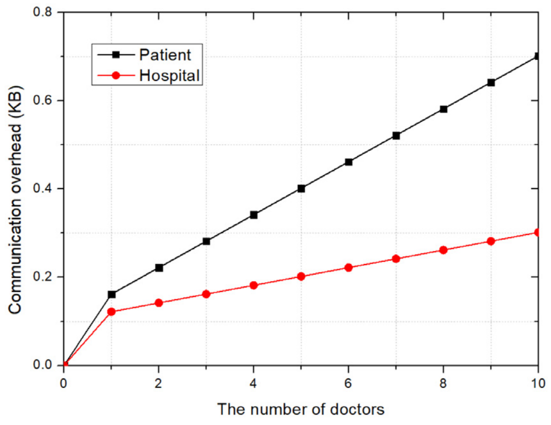

Fig. 8. Communication overhead on the patient and hospital.

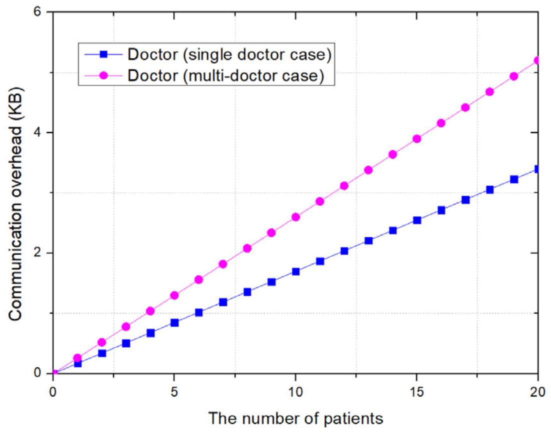

Fig. 9. Communication overhead on the doctor.

Table 1
Notation of operations.

Symbol	Operation
Exp_G	Exponent operation in G
Mul_G	Group operation in G
$Hash_{Z_p}$	Hash a value into Z_p
$Pair_{G_T}$	Computing pairing $e(\chi, \varsigma)$ where $\chi, \varsigma \in G$
CTrans	Conduct a transaction in Ethereum
Enc	Encrypt a message by using symmetric encryption algorithm
$Hash_G$	Hash a value into G

TP-EHR is built on the Ethereum blockchain. To date, a light client protocol of Ethereum has been released, which enables users to conduct transaction without maintaining and storing the Ethereum blockchain. Therefore, TP-EHR is efficient in terms of communication overhead.

7.2. Computational overhead

In the experiment, we would not analyze the computational costs on generating EHRs for doctors, since the computational costs to generate EHRs are subject to multiple factors, such as the type and size of EHRs.

We first estimate the computational costs in terms of basic cryptographic operations, the notations are shown in Table 1.

We show the computation costs on the hospital, patient, doctor, and cloud server in Table 2, where n denotes the number of doctors.

We also show the computation delay on the hospital, patient, doctor, and cloud server in Fig. 10.

As shown in the experiment results, in TP-EHR, it takes within 1 s to outsource EHRs to the cloud storage for a doctor who is equipped with a computer.

Table 2
Computation costs.

	Computation costs
Hospital	$2Exp_G + 3Mul_G + Hash_{Z_p} + Hash_G$
Patient	$(2+n) \cdot Exp_G + 3Mul_G + (n+1) \cdot Hash_G + Hash_{Z_p}$
Doctor (single doctor case)	$Enc + CTrans + Hash_{Z_p}$
Doctor (multi-doctor case)	$2Pair_{G_T} + Hash_G + Enc + CTrans + Hash_{Z_p}$
Cloud server	$2Pair_{G_T} + Hash_G$

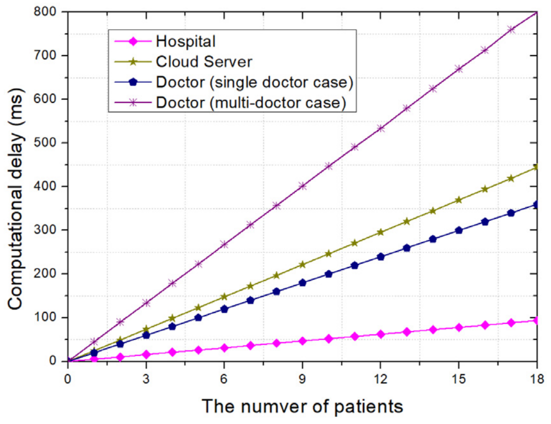

Fig. 10. Computational delay.

Actually, conducting a transaction in Ethereum takes averagely 3 min to confirm it. Consequently, for a patient in the multi-doctor case, the time interval between the doctor $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$ to generate EHRs only requires 3 min. This requirement is practical, since the time interval between two successive doctors to generate EHRs for the same patient is much larger than 3 min in reality.

7.3. Monetary costs to store EHRs

In TP-EHR, the main monetary costs to store EHRs are caused by conducting transactions in Ethereum. Specifically, in Store of TP-EHR, once a doctor generates an EHR for the patient, she/he needs to create a new transaction to record the EHR and protect it from illegal modification. Recording a transaction in Ethereum requires a transaction fee for the doctor. At the time of writing this paper (October 2018), conducting a transaction in Ethereum averagely takes 8 US cents, which is acceptable in reality.

8. Conclusion and future work

In this paper, we have proposed TP-EHR, a blockchain-based secure eHealth system that ensures the confidentiality of outsourced EHRs and prevents outsourced EHRs from illegally modifying. The security of TP-EHR can be guaranteed even if a malicious doctor who generates and outsources EHRs colludes with the cloud server to tamper with the EHRs. TP-EHR is constructed on the Ethereum blockchain, where the EHRs generated by one doctor in a treatment are integrated into a transaction of Ethereum, such that the correctness and integrity of EHRs are based on the security of Ethereum, and the time when the EHRs are generated can be efficiently extracted. The security analysis has demonstrated that TP-EHR is secure against various attacks existing in actual cloud-assisted eHealth systems. We have also conducted a comprehensive performance analysis, which proves that TP-EHR is practical and efficient in terms of communication and computation overhead.

For the future work, we will investigate how to utilize the blockchain technology to enhance eHealth systems. It is promising to integrate the blockchain technology into eHealth systems to improve their quality of services.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2016QY061205 and 2018YFB0804103), Science and Technology Service Industry project of Sichuan Provincial Science and Technology Department (2017GFW0002 and 2017GFW0088), Achievements Transformation Project of Sichuan Provincial Science and Technology Department (2017CC0006), Provincial Academy of Science and Technology Project of Sichuan Provincial Science and Technology Department (2017JZ0015), the National Natural Science Foundation of China (61672437 and 61702428), the New Generation Artificial Intelligence Science and Technology Major Project of Sichuan Province (2018GZDZX0043) and the Sichuan Science and Technology Program (19ZDZX0045, 2018GZ0185, 2018GZ0086, 2018GZ0095).

References

[1] M. Abdalla, D. Pointcheval, Simple password-based encrypted key exchange protocols, in: Proceedings of CT-RSA, 2005, pp. 191–208.
[2] U.R. Acharya, S. Bhat, J.E. Koh, S.V. Bhandary, H. Adeli, A novel algorithm to detect glaucoma risk using texton and local configuration pattern features extracted from fundus images, Comput. Biol. Med. 88 (2017) 72–83.
[3] F. Armknecht, J.-M. Bohli, G.O. Karame, Z. Liu, C.A. Reuter, Outsourced proofs of retrievability, in: Proceedings of ACM CCS, 2014, pp. 831–843.
[4] A. Azaria, A. Ekblaw, T. Vieira, A. Lippman, Medrec: using blockchain for medical data access and permission management, in: Proceedings of OBD, IEEE, 2016, pp. 25–30.
[5] M. Bellare, J. Kilian, P. Rogaway, The security of the cipher block chaining message authentication code, J. Comput. Syst. Sci. 61 (3) (2000) 362–399.
[6] D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, SIAM J. Comput. 32 (3) (2003) 586–615.
[7] D. Boneh, B. Lynn, H. Shacham, Short signatures from the weil pairing, J. Cryptol. 17 (4) (2004) 297–319.
[8] X. Cai, S. Wang, X. Lu, W. Li, Y. Liang, Parametric and adaptive encryption of feature-based computer-aided design models for cloud-based collaboration, Integr. Comput. Aided Eng. 24 (2) (2017) 129–142.
[9] S. Caíno-Lores, A. García, F. García-Carballeira, J. Carretero, Efficient design assessment in the railway electric infrastructure domain using cloud computing, Integr. Comput. Aided Eng. 24 (1) (2017) 57–72.
[10] F. Caraffini, F. Neri, A study on rotation invariance in differential evolution, Swarm Evol. Comput. (2019) To appear. https://www.sciencedirect.com/science/article/pii/S2210650218301482.
[11] F. Caraffini, F. Neri, L. Picinali, An analysis on separability for memetic computing automatic design, Inf. Sci. 265 (2014) 1–22.
[12] V. Casola, A. Castiglione, K.-K.R. Choo, C. Esposito, Healthcare-related data in the cloud: challenges and opportunities, IEEE Cloud Comput. 3 (6) (2016) 10–14.
[13] X. Du, M. Guizani, Y. Xiao, H.-H. Chen, Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks, IEEE Trans. Wireless Commun. 8 (3) (2009) 1223–1229.
[14] X. Du, Y. Xiao, M. Guizani, H.-H. Chen, An effective key management scheme for heterogeneous sensor networks, Ad Hoc Netw. 5 (1) (2007) 24–34.
[15] A.C. Ekblaw, MedRec: Blockchain for Medical Data Access, Permission Management and Trend Analysis, Massachusetts Institute of Technology, 2017 Ph.D. thesis.
[16] J.L. Fernández-Alemán, I.C. Señor, P.Á.O. Lozoya, A. Toval, Security and privacy in electronic health records: a systematic literature review, J. Biomed. Inform. 46 (3) (2013) 541–562.
[17] C. Gao, S. Lv, Y. Wei, Z. Wang, Z. Liu, X. Cheng, M-sse: an effective searchable symmetric encryption with enhanced security for mobile devices, IEEE Access 6 (2018) 38860–38869.
[18] X. Hei, X. Du, S. Lin, I. Lee, Pipac: patient infusion pattern based access control scheme for wireless insulin pump system, in: Proceedings of IEEE INFOCOM, 2013, pp. 3030–3038.
[19] T.J. Hirschauer, H. Adeli, J.A. Buford, Computer-aided diagnosis of Parkinson's disease using enhanced probabilistic neural network, J. Med. Syst. 39 (11) (2015) 179.
[20] L. Jiang, Y. Cheng, L. Yang, J. Li, H. Yan, X. Wang, A trust-based collaborative filtering algorithm for e-commerce recommendation system, J. Ambient Intell. Hum. Comput. (2018) 1–12.
[21] L. Jiang, T. Li, X. Li, M. Atiquzzaman, H. Ahmad, X. Wang, Anonymous communication via anonymous identity-based encryption and its application in iot, Wireless Commun. Mob. Comput. 2018 (2018) Article ID 6809796, 8 pages.
[22] M. Korytkowski, Novel visual information indexing in relational databases, Integr. Comput. Aided Eng. 24 (2) (2017) 119–128.
[23] W.-B. Lee, C.-D. Lee, A cryptographic key management solution for hipaa privacy/security regulations, IEEE Trans. Inf. Technol. Biomed. 12 (1) (2008) 34–41.
[24] J. Li, X. Chen, M. Li, J. Li, P.P. Lee, W. Lou, Secure deduplication with efficient and reliable convergent key management, IEEE Trans. Parallel Distrib. Syst. 25 (6) (2014) 1615–1625.
[25] J. Li, X. Huang, J. Li, X. Chen, Y. Xiang, Securely outsourcing attribute-based encryption with checkability, IEEE Trans. Parallel Distrib. Syst. 25 (8) (2014) 2201–2210.
[26] T. Li, W. Chen, Y. Tang, H. Yan, A homomorphic network coding signature scheme for multiple sources and its application in IoT, Secur. Commun. Netw. 2018 (2018) Article ID 9641273, 6 pages.
[27] H. Lin, J. Shao, C. Zhang, Y. Fang, Cam: cloud-assisted privacy preserving mobile health monitoring, IEEE Trans. Inf. Forensics Secur. 8 (6) (2013) 985–997.
[28] J. Liu, N. Asokan, B. Pinkas, Secure deduplication of encrypted data without additional independent servers, in: Proceedings of ACM CCS, 2015, pp. 874–885.
[29] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.
[30] F. Neri, Linear Algebra for Computational Sciences and Engineering, Springer, 2016.
[31] F. Padillo, J.M. Luna, F. Herrera, S. Ventura, Mining association rules on big data through MapReduce genetic programming, Integr. Comput. Aided Eng. 25 (1) (2018) 31–48.
[32] L. Pan, G. Puaun, G. Zhang, F. Neri, Spiking neural *P* systems with communication on request, Int. J. Neural Syst. 27 (8) (2017) 1–13.
[33] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, Z.-H. Zhan, Secure data uploading scheme for a smart home system, Inf. Sci. 453 (2018) 186–197.
[34] M. Smolik, V. Skala, Large scattered data interpolation with radial basis functions and space subdivision, Integr. Comput. Aided Eng. 25 (1) (2018) 1–14. Preprint.
[35] C. Sridhar, S. Bhat, U.R. Acharya, H. Adeli, G.M. Bairy, Diagnosis of attention deficit hyperactivity disorder using imaging and signal processing techniques, Comput. Biol. Med. 88 (2017) 93–99.
[36] J. Sun, X. Zhu, C. Zhang, Y. Fang, Hcpp: cryptography based secure EHR system for patient privacy and emergency healthcare, in: Proceedings of IEEE ICDCS, 2011, pp. 373–382.
[37] H. Tian, Z. Chen, C.-C. Chang, Y. Huang, T. Wang, Z.-A. Huang, Y. Cai, Y. Chen, Public audit for operation behavior logs with error locating in cloud storage, Soft Comput. (2018) 1–14.
[38] F. Tschorsch, B. Scheuermann, Bitcoin and beyond: a technical survey on decentralized digital currencies, IEEE Commun. Surv. Tutor. 18 (3) (2016) 2084–2123.
[39] Y. Wang, Q. Wu, B. Qin, W. Shi, R.H. Deng, J. Hu, Identity-based data outsourcing with comprehensive auditing in clouds, IEEE Trans. Inf. Forensics Secur. 12 (4) (2017) 940–952.
[40] G. Wood, Ethereum: a secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper 151 (2014) 1–32. http://ljk.imag.fr/membres/Jean-Guillaume.Dumas/Enseignements/ProjetsCrypto/Ethereum/ethereum-yellowpaper.pdf.
[41] H. Yan, X. Li, Y. Wang, C. Jia, Centralized duplicate removal video storage system with privacy preservation in IoT, Sensors 18 (6) (2018) 1814.
[42] X. Zhang, H. Wang, C. Xu, Identity-based key-exposure resilient cloud storage public auditing scheme from lattices, Inf. Sci. 472 (2019) 223–234.
[43] Y. Zhang, C. Xu, H. Li, X. Liang, Cryptographic public verification of data integrity for cloud storage systems, IEEE Cloud Comput. (5) (2016) 44–52.
[44] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, X. Lin, Healthdep: an efficient and secure deduplication scheme for cloud-assisted ehealth systems, IEEE Trans. Ind. Inf. 14 (9) (2018) 4101–4112.
[45] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, X. Zhang, Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation, IEEE Trans. Inf. Forensics Secur. 12 (3) (2017) 676–688.
[46] Y. Zhang, C. Xu, S. Yu, H. Li, X. Zhang, SCLPV: secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors, IEEE Trans. Comput. Social Syst. 2 (4) (2015) 159–170.