2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C) | 978-1-6654-7836-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/QRS-C55045.2021.00098

Hierarchical Cloud-Based Consortium Blockchains for Healthcare Data Storage

Alvin Thamrin
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
athamrin@umassd.edu

Haiping Xu
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
hxu@umassd.edu

***Abstract*—Blockchain technology can be used as a healthcare solution that allows hospitals to store and share electronic health records (EHRs) in a secure and reliable manner. However, the number of hospitals that can participate in a blockchain network is limited by the inclusion of big data such as multimedia files, which presents an obvious scalability problem. In this paper, we introduce a hierarchical cloud-based consortium blockchain framework for storing big data including multimedia files within cloud-based local hospital blockchain networks and sharing them with hospitals outside the networks through high-level blockchain networks, called city blockchain networks and state blockchain network. We present procedures for concurrently searching EHRs, creating access control policies for authorized access, and retrieving EHRs through hierarchical blockchain networks. The experimental results show that our approach is feasible and efficient for accessing and sharing EHRs using hierarchical cloud-based consortium blockchains throughout a country.**

Keywords—Hierarchical blockchains, electronic health records, multimedia files, cloud-based blockchain, access control policies

I. Introduction

Blockchain technology has been a popular subject for research and exploration of its potential use in the healthcare sector [1], [2]. Blockchain is a decentralized data storage that records information in chunks of data defined as blocks [3]. These blocks are chained together chronologically through cryptography and can be used to efficiently record information in a verifiable and permanent way. A consortium blockchain, also called a *federated* blockchain, is a permissioned blockchain [4]. Unlike a public blockchain, the access to a consortium blockchain is restricted to certain nodes. In the context of healthcare, a consortium blockchain can support the storage and preservation of patients' medical data and history with local hospitals in a decentralized manner, allowing patients to have ownership over their stored medical data. In earlier work, we introduced a cloud-based blockchain scheme to achieve data accessibility, redundancy, and security for storing and sharing electronic health records (EHRs) on a local scale [2]. The approach allows for big data including multimedia files to be safely stored in a cloud-based blockchain, and for information to be efficiently retrieved via a lite blockchain that stores EHRs' metadata and text-based information only. However, due to the inclusion of multimedia files, there is a limit to the number of hospitals that can participate in a consortium blockchain network, which presents an obvious scalability problem. More specifically, the number of participating hospitals is generally correlated with the frequency of EHRs added to the blockchain each day, which can lead to a highly inflated blockchain size if the number of hospital participants is high. Thus, through our previous approach, local areas (e.g., cities) should form their own blockchain networks to keep the number of hospital participants small. Unfortunately, creating multiple blockchain networks rather than one unified network presents new challenges. Simply put, we will not be able to rely on the use of blockchain methods for communication between hospitals in different network groups because in its current design, different blockchains inherently do not work together. This dilemma may be solved if we use an off-chain approach to store big data outside the blockchain [5], [6]; however, the off-chain approach has its major drawbacks. For example, depending on how off-chain storage is set up, issues such as database bottleneck, lack of redundancy and accessibility can present significant challenges. On the other hand, when all data is stored in a blockchain using on-chain methods, the data becomes immutable, redundant, tamper-proof, and available. Therefore, the goal of this research is to propose a new on-chain method for cross-network communication and information retrieval between hospitals in different blockchain networks.

In this paper, we introduce a hierarchical cloud-based consortium blockchain framework for healthcare data storage, which contains multiple layers of blockchain networks including hospital blockchain networks (HBNs), city blockchain networks (CBNs), and a state blockchain network (SBN). In the first layer, a HBN is designed as a consortium blockchain shared by hospitals within a local area such as a city. In this layer, unlike the framework we previously proposed in [2], each HBN consists of multiple hospital super peer agents as participants and a unique city super peer agent as the regulator or manager of the network. In the second layer, we group all city super peer agents within a state as participants of a CBN that is managed by a unique state super peer agent. A CBN is designed as a consortium blockchain shared by city super peer agents within a state. This design allows agents from various HBNs within the same state to communicate with each other through city super peer agents for the purpose of data sharing. Similarly, in the third layer, all state super peer agents within a country are grouped as participants of a consortium blockchain SBN. Agents from various CBNs in the country can communicate with each other and share data through state super peer agents. Based on this layered design of hierarchical cloud-based consortium blockchain networks, our approach expands the scope of accessible EHRs that can be stored and shared in a secure and reliable manner across all hospitals within a country, while circumventing the scalability issue mentioned earlier.

2693-9371/21/$31.00 ©2021 IEEE
DOI 10.1109/QRS-C55045.2021.00098

II. Related Work

There have been many previous efforts to develop effective storage and sharing systems using the blockchain technology. Such research typically utilizes off-chain approaches to deal with the problem of storing sensitive data or big data in blockchain systems. Su et al. proposed a secure data sharing solution for sensitive financial data using blockchain and proxy re-encryption technology [5]. In their approach, sensitive data are stored in off-chain distributed databases; while access control rules, hash value and storage address of the data are stored in the blockchain. Jeong et al. proposed a video surveillance storage and sharing system using the Hyperledger Fabric platform [6]. The video themselves are encrypted and stored off-chain using distributed InterPlanetary file system (IPFS); while the metadata of the video is stored in the blockchain. Additionally, the videos can only be viewed (not downloaded) by authorized users via a CDN (Content Delivery Network), a network used for transmitting encrypted videos. Wang and Song proposed a blockchain framework using an attribute-based cryptosystem for the development of a secure EHR storage and sharing system [7]. In their approach, the EHRs are stored in the cloud with their metadata recorded in the blockchain. Unlike the common off-chain approach described above, our on-chain approach stores all data in the blockchain through a cloud-based blockchain scheme, which provides the advantages of a complete blockchain storage solution in terms of data immutability, integrity, and availability.

The existing research on novel designs in blockchain architecture is summarized as follows. Cui et al. proposed a compacted directed acyclic graph (CoDAG)-based blockchain protocol to be used in the field of Industrial Internet of Things (IIoT) [8]. The authors introduced and developed protocols and algorithms to maintain and secure their proposed CoDAG-based IIoT architecture. Fernandes et al. proposed a scalable blockchain scheme for sharing EHRs among patients, healthcare professionals, and health institutions [9]. In their proposed blockchain architecture, one blockchain is used to record patient visits, while another blockchain is created for each health institution to record links to EHRs that are stored in external systems. Egala et al. proposed a decentralized Internet of Medical Things (IoMT) smart healthcare system, called Fortified-Chain, which provides a decentralized EHR and automation of smart contract-based services without compromising the security and privacy of the system [10]. In their approach, a blockchain-based distributed data storage system (DDSS) network consists of peers associated with a hospital storing patient medical data. A subset of patient non-critical information is generated and published to a global DDSS network that supports communication between third-party healthcare services and local DDSS networks. Although the above approaches proposed novel blockchain architectures to facilitate communication among peers, they all use external storage to store sensitive information or big data. Unlike these approaches, we introduce a hierarchical blockchain architecture to create an effective cloud-based on-chain system for storing and sharing EHRs among hospitals from different cities and states. By allowing different blockchain networks located in different cities and states to communicate and share data with each other, we can effectively spread the storage of EHRs across multiple blockchains used by different networks. Thus, our hierarchical blockchain approach provides a scalable solution for storing sensitive information and big data in cloud-based blockchain networks across the country.

There is also a lot of research discussing how to implement access control mechanisms in blockchain systems to prevent unauthorized access to confidential data by unwanted users. Buzachis et al. proposed a Blockchain-as-a-Service based solution for Health Information Exchange (BaaS-HIE) activities to address security issues in health information system such as patient privacy, medical record integrity, fine-grained access control, and private and auditable healthcare data sharing [11]. Their approach involves the use of a private blockchain based on Ethereum protocol and smart contracts as access control management for medical records. Xia et al. proposed a blockchain-based system called MeDShare, which was developed to solve the problem of sharing medical data between data custodians in a trust-less environment [12]. The design employs smart contracts and an access control mechanism to trace and monitor the behavior of any stored data, so that once any violation of data permissions is detected, the offending user will have access revoked. Guo et al. proposed a hybrid architecture of blockchain and edge nodes to facilitate EHR management [13]. Attribute-based multi-signature scheme and attribute-based encryption scheme were used to authenticate a user's signature without revealing sensitive information and to encrypt EHR data, which are stored separately on the edge nodes. In contrast to the mechanisms described above, our approach involves the implementation of different scopes of role-based access control (RBAC) policies [14] that restrict user access to EHRs stored in different healthcare facilities in different cities and states. There are three layers of networks in our approach, each implementing its own RBAC policies, namely local hospital-wide policies, city-wide policies, and statewide policies. As a result, our method provides a more comprehensive and reliable mechanism than other methods because it is designed to work in a larger environment.

III. A Framework for Hierarchical Blockchains

A. Hierarchical Cloud-Based Consortium Blockchains

The architecture of hierarchical cloud-based consortium blockchains consists of three layers of blockchain networks: hospital, city, and state layers. As shown in Fig. 1, the hospital layer consists of multiple HBNs, each of which involves several hospitals located within a city and their end users (i.e., doctors, nurses, and patients). To simplify matters, in this paper, we define a city as a general term for any form of governmental jurisdiction below the state level. A cloud-based and lite block scheme is implemented in an HBN, allowing big data to be stored in a cloud-based blockchain without incurring scalability issues for regular peers [2]. Each hospital is represented by a hospital super peer agent β_{HOS}, who maintains its private cloud. A number of agents β_{HOSs} representing various hospitals within a city handle the approval or rejection of requests from end users, represented by regular peer agents β_{REPs}, on access to a patient's EHRs stored in a cloud-based blockchain of an HBN. An HBN is directly connected to a city super peer agent β_{CIT} that acts as a network regulator and representative of the city. In the city layer, there are a number of CBNs, each of which involves a number of city super peer agents β_{CITs} from the same state. A CBN is directly connected to a state super peer agent β_{STA} that

acts as a network regulator and representative of the state. Unlike the hospital and city layer, the state layer contains only one SBN, which involves all state super peer agents β_{STAs} from the country.

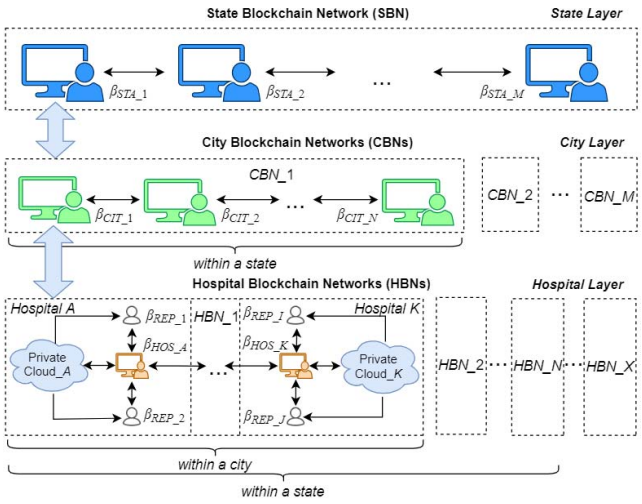

Fig.1. The architecture of hierarchical cloud-based consortium blockchains

Note that the purpose of this hierarchical design is to allow searching and retrieving EHRs from various hospitals across cities and states through city super peer agents β_{CITs} and state super peer agents β_{STAs}. The details about searching and retrieving EHRs across cities and states are described in Section IV. In the following sections, we define the block structure in the blockchain of HBN, CBN and SBN.

B. Block Record Types and State / City Blocks

There are two different types of block records that can be stored in an SBN's state blockchain. These are statewide record for access control policies SR_{ACP} and statewide access record SR_{AR}. A record SR_{ACP} stores the access control policies enforced by the relevant state super peer agent β_{STA} in the SBN, which is established to check for any requests concerning the access of a patient's EHRs stored in HBNs across states. An SR_{ACP} is defined as a triple (P, L, T), where P is a set of policies; L is a set of locations (states, cities, and hospitals) where the policies must be enforced; and T is the timestamp when the policies are created. Unlike an SR_{ACP}, a record SR_{AR} stores access requests or search information of a patient's EHRs in hospitals across states, and it serves as a history log that keeps track of regular peer actions to ensure accountability. A record SR_{AR} is defined as 5-tuple (N, D, O, T, I), where N is the request number associated with the requestor and the requestee; D is the details of the request; O is the outcome of the request, which can be approved or rejected; T is the time when the request is created; and I is the index link that points to the nearest previous block that contains an SR_{AR} of the same regular peer. This enables all access records of a regular peer to be linked together in a linked list for efficient retrieval of access records.

Similarly, a city block shares the same structure as that of a state block and stores city-wide records for access control policies CR_{ACP} and city-wide access record CR_{AR}. A record CR_{ACP} stores the access control policies enforced by the relevant city super peer agent β_{CIT} in a CBN, which is established to check for any requests concerning the access of a patient's EHRs stored in HBNs across cities within the same state; while a record CR_{AR} stores access requests or search information of a patient's EHRs in hospitals across cities within the same state. Fig. 2 shows the same structure of a new state or city block B_{h+1} from a state or city blockchain, respectively, where h is the length of the current blockchain.

Block B_{h+1}			
Header	**(State / City) Block Records**		**Verification Info**
$hash(B_h)$	**Access Control Policies**	**Access Records**	$hash(B_{h+1})$
Time Stamp	SR_{ACP_1} / CR_{ACP_1}	SR_{AR_1} / CR_{AR_1}	$ds[B_{h+1}]_v$ list
Block ID	...	...	
BC Length: h	SR_{ACP_n} / CR_{ACP_n}	SR_{AR_l} / CR_{AR_l}	

Fig. 2. The structure of a new state or city block B_{h+1}

From the figure, we can see that block B_{h+1} consists of three sections: header, state or city block records, and verification information. The header section contains the previous block's hash value $hash(B_h)$, the timestamp when B_{h+1} is created, the block ID of B_{h+1}, and the current blockchain length h. The state or city block records section contains two lists of records (SR_{ACP} records, SR_{AR} records) or (CR_{ACP} records, CR_{AR} records), respectively. The verification information section contains the hash value of B_{h+1} and a list of digital signatures $ds[B_{h+1}]_v$, where each peer v is a city or state super peer who approves B_{h+1} during the consensus process. Note that the purpose of developing the SBN and multiple CBNs is to facilitate accessing EHRs across states and cities; however, a patient's EHRs are not stored in a state block or a city block; instead, they are recorded in a hospital block of a blockchain of an HBN.

C. Block Record Types and Hospital Blocks

Unlike city and state blockchains, a hospital blockchain in an HBN can be one of two blockchain variants: the cloud-based hospital blockchain (CHB) and its simplified version, called lite hospital blockchain (LHB) [2]. A CHB stores the same block record types as in its corresponding LHB, but contains any number of multimedia files that do not exist in the LHB. There are four different types of block records that can be stored in a CHB or an LHB, namely HR_{UPR}, HR_{ACP}, HR_{MER}, and HR_{AR}. A record HR_{UPR} stores the user profile and account information of a regular peer in an HBN. An HR_{UPR} is defined as a 6-tuple (I, N, R, U, S, T), where I is the regular peer's identification in the HBN; N is the regular peer's full name; R, U and S are the regular peer's private key, public key and secret symmetric key, respectively; and T is the timestamp when the HR_{UPR} is created. An HR_{UPR} is created whenever a peer's user profile is updated or when a new peer joins the HBN. A record HR_{ACP} stores access control policies enforced by the relevant hospital super peer agent β_{HOS} in the HBN. An HR_{ACP} has the same structure as an SR_{ACP} in a state blockchain, except that the set of locations contain only the names of hospitals as an HR_{ACP} only records access control policies related to the hospitals within the same city. An HR_{ACP} is created to check any requests regarding access to a patient's EHRs stored in different hospitals within the same city where the patient currently resides. A record HR_{MER} stores

medical reports of a patient as well as the metadata of the associated multimedia files resulting from a doctor's visit. An HR_{MER} is defined as 6-tuple (I, H, X, M, T, I), where I are the identifications of all peers involved in the doctor's visit, including the patient, the nurse and the doctor; H is the name of the hospital visited by the patient; X includes a summary of the visit and any text-based medical data; M is the metadata of any multimedia files generated from the doctor's visit; T is the timestamp when the HR_{MER} record is created; and I is the index link that points to the nearest previous block that contains an HR_{MER} record of the same patient. Finally, a record HR_{AR} stores access requests or search information of a patient's EHRs in hospitals within the same city. The structure of an HR_{AR} is the same as that of an SR_{AR} in a state blockchain. Note that the HR_{AR} records were not included in our earlier design [2]; however, they become necessary for a hospital super peer agent to validate previous searches and accesses in the HBN and ensure accountability by tracking regular peer actions, just as the city or state super peer agent does in the CBN or SBN, respectively. Fig. 3 shows the structure of a new cloud-based block CB_{h+1} with the four types of hospital block records, where h is the length of the current blockchain.

Cloud-based Block CB_{h+1}				
Header	**Hospital Block Records**			
$hash(LB_h)$	***User Profiles***	***Access Control Policies***	***Medical Records***	***Access Records***
$hash(CB_h)$	HR_{UPR_1}	HR_{ACP_1}	HR_{MER_1}	HR_{AR_1}
Time Stamp	...	...	...	...
Block ID	HR_{UPR_m}	HR_{ACP_n}	HR_{MER_o}	HR_{AR_p}
CHB Length: h				
Verification Info	**Multimedia Files**			
$hash(LB_{h+1})$	Multimedia File 1			
$hash(CB_{h+1})$	...			
$ds[CB_{h+1}]_v$ list	Multimedia File k			

Fig. 3. Cloud-based block CB_{h+1} in a hospital blockchain

As shown in the figure, a new block CB_{h+1} consists of four sections: header, hospital block records, multimedia files, and verification information. The header section contains the hash values of the previous cloud and lite blocks, i.e., $hash(CB_h)$ and $hash(LB_h)$, the timestamp when CB_{h+1} was created, the block ID, and the length h of the current blockchain. The hospital block records section contains any number of records HR_{UPR}, HR_{ACP}, HR_{MER}, and HR_{AR}. The multimedia files section contains any number of multimedia files compressed together with their metadata recorded in the associated HR_{MER}. Lastly, the verification information section contains the hash value of the header and hospital block records, i.e., $hash(LB_{h+1})$, and the hash value of the header, hospital block records, and multimedia files, i.e., $hash(CB_{h+1})$. This section also contains a list of digital signatures $ds[CB_{h+1}]_v$, where each peer v is an agent β_{HOS}, who approves CB_{h+1} during the consensus process. While not shown, a lite block LB_{h+1} shares the same structure as CB_{h+1} but with the multimedia files section removed.

D. Block Generation and the Approval Process

Let state super peer agent $\beta_{STA\text{-}\Psi}$ be the one who generates a new state block B_{h+1}. Algorithm 1 shows the procedure for generating and approving the new state block B_{h+1} before it is added to the state blockchain.

Algorithm 1: Generating and Approving a New State Block

Input: A list of state block records Ξ containing records SR_{ACP}, SR_{AR}, and the total number of state super peer agents λ.
Output: A new state block B_{h+1} digitally signed by the majority of state super peer agents.

1. Create an empty state block B_{h+1}
2. Verify and add $hash(B_h)$, time stamp, block ID, and current blockchain length h to the header section of B_{h+1}
3. **for** each state block record φ in the list of records Ξ
4. Encrypt φ and add it to the state block records section of B_{h+1}
5. Calculate $hash(B_{h+1})$ and add it to the verification section of B_{h+1}
6. Create digital signature $ds[B_{h+1}]_\Psi$ using $hash(B_{h+1})$
7. Add $ds[B_{h+1}]_\Psi$ to the $ds[B_{h+1}]_v$ list in the verification section of B_{h+1}
8. Let ρ be a list of all other state super peer agents
9. Broadcast B_{h+1} to each element v in ρ and request approval digital signature $ds[B_{h+1}]_v$ asynchronously
10. **while** (*not* timeout) and (the size of $ds[B_{h+1}]_v$ list $\leq \lambda/2$)
11. **if** received $ds[B_{h+1}]_v$ is valid, add it to $ds[B_{h+1}]_v$ list in B_{h+1}
12. **else** discard $ds[B_{h+1}]_v$
13. **if** (timeout) **return** *null* // not approved by the majority
14. **else return** B_{h+1}

According to the algorithm, agent $\beta_{STA\text{-}\Psi}$ first creates an empty state block B_{h+1}, and then completes the header section in B_{h+1}. For each state block record φ in the record list Ξ, $\beta_{STA\text{-}\Psi}$ encrypts it using its public key before adding it to the state block records section in B_{h+1}. In the following steps, it calculates the hash value $hash(B_{h+1})$, creates the digital signature $ds[B_{h+1}]_\Psi$, and adds them to the verification section in B_{h+1}. During the approval phase (i.e., the consensus process), agent $\beta_{STA\text{-}\Psi}$ broadcasts B_{h+1} to all other state super peer agents and requests their digital signatures. If B_{h+1} has been validly signed by the majority of the state super peer agents before timeout, B_{h+1} is then returned as a newly approved state block. Due to the adoption of identical block structure for city blocks, the block generation and approval process are similar for a new city block in a city blockchain. For block generation of a new hospital block and its approval process, refer to earlier work [2].

IV. Searching and Retrieving EHRs in Hierarchical Cloud-Based Consortium Blockchains

In this section, we show the steps involved in searching and retrieving a patient's EHRs across different layers in our hierarchical blockchain approach. In the first step, all hospitals, cities and states are searched concurrently through their super peer agents for the locations of the hospitals that store the patient's EHRs. In the second step, the system seeks the patient's permission and creates the access control policies to enable retrieval of the patient's EHRs. In the third step, the patient's EHRs are retrieved from the hospitals based on the specified access control policies.

A. Concurrent Searches for Hospital Locations

Finding where a particular patient's EHRs are stored in a country involves all of the super peer agents in the three layers of our hierarchical blockchain network structure. To better illustrate the flow of the search process, we break it down into three tasks: searching for EHRs across hospitals within the same

city; searching for EHRs across cities within the same state; and searching for EHRs across states within a country. Algorithm 2 details how to search for a patient's EHRs stored in different hospitals within the same city to which the requestor belongs. The process is initiated by a requestor (e.g., a doctor) who sends a request to its hospital super peer agent β_{HOS} to search for hospitals that contain patient p's EHRs within a city. Agent β_{HOS} then forwards the request to its city super peer agent β_{CIT}, who starts the searching process.

Algorithm 2: Searching for Hospitals with a Patient's EHRs within the Same City by a City Super Peer Agent β_{CIT}

Input: A search request for hospitals containing patient p's EHRs
Output: A list of hospitals that contain patient p's EHRs

1. Let ρ_{h_list} be the list of hospital super peers under β_{CIT}'s jurisdiction
2. Let η_{h_list} be an empty list of hospitals
3. **for** each γ_h in ρ_{h_list}
4. forward the search request to γ_h asynchronously, which invokes a search in hospital h
5. **while** (*not* timeout) or | η_{h_list} | != | ρ_{h_list} |
6. **if** hospital h contains p's EHRs in γ_h's response
7. add h to the list η_{h_list}
8. **else continue** // h does not contain p's EHRs in γ_h's response
9. **return** the list η_{h_list}

According to the algorithm, agent β_{CIT} sends out concurrent requests to all hospital super peer agents under its jurisdiction in an HBN and waits for responses from them or until timeout. If β_{CIT} receives a response from a hospital super peer agent γ_h with p's EHRs, γ_h is added to a hospital list η_{h_list}; otherwise, γ_h is ignored. When all hospital super peer agents have responded or timed out, the hospital list η_{h_list} is returned, which will be sent back to β_{HOS}. Upon receiving the list of hospitals with patient p's EHRs, β_{HOS} contacts patient p for his/her consent and notifies the requestor of the list of hospitals approved by patient p.

The procedure of searching for a patient's EHRs across cities within the same state is performed by a state super peer agent β_{STA}. Similarly, the process is initiated by a requestor (e.g., a doctor) who sends a request to its hospital super peer agent β_{HOS} to search for hospitals that contain patient p's EHRs within a state. Agent β_{HOS} then forwards the request to its city super peer agent β_{CIT}, who further forwards the request to its state super peer agent β_{STA}. Algorithm 3 shows how β_{STA} initiates concurrent searches in the state. According to the algorithm, agent β_{STA} sends out concurrent requests in a CBN to all city super peer agents under its jurisdiction and waits for responses from them or until it times out. Upon receiving the search request, each city super peer agent γ_c executes Algorithm 2 to search for hospitals that contain patient p's EHRs within a city. If γ_c returns a list of hospitals η_{h_list} with p's EHRs, η_{h_list} is appended to the hospital list $\eta_{c\_h\_list}$; otherwise, γ_c's response is ignored. When all city super peer agents have responded or it times out, the hospital list $\eta_{c\_h\_list}$ is returned, which will be sent back to β_{CIT}, who further sends the list back to β_{HOS} for further processing as in the case of searching for EHRs across hospitals within the same city.

Finally, searching for a patient's EHRs across states within a country begins similarly with the previous procedure. The state super peer agent β_{STA}, who receives the request, initiates the concurrent searches by broadcasting the request to all other state super peer agents in the SBN. Each state super peer agent who receives the search request executes Algorithm 3 to search for hospitals that contain patient p's EHRs within a state. All searching results will be sent back to β_{STA} and further processed as in the case of searching for EHRs across hospitals within the same state.

Algorithm 3: Searching for Hospitals with a Patient's EHRs within the Same State by a State Super Peer Agent β_{STA}

Input: A search request for hospitals containing patient p's EHRs
Output: A list of hospitals that contain patient p's EHRs

1. Let ρ_{c_list} be the list of city super peers under β_{STA}'s jurisdiction
2. Let $\eta_{c\_h\_list}$ be an empty list of hospitals; *nResponse* = 0
3. **for** each γ_c in ρ_{c_list}
4. forward the search request to γ_c asynchronously, which invokes *Algorithm* 2 to search within city c
5. **while** (*not* timeout) or *nResponse* != | ρ_{c_list} |
6. **if** γ_c returns a list of hospitals η_{h_list} with p's EHRs
7. append η_{h_list} to $\eta_{c\_h\_list}$; *nResponse*++
8. **else** *nResponse*++; **continue** // γ_c returns an empty list
9. **return** the list $\eta_{c\_h\_list}$

B. Creating Access Conrol Policies

In our approach, we define access control policies as mandatory rules that specify which data in a blockchain can be accessed by participants based on their credentials. Thus, it is crucial that we assign appropriate permissions to each role or a participant with a certain role to prevent unauthorized access to medical data stored in the hierarchical blockchain networks [15]. Once the hospital locations containing a patient p's EHRs are discovered and the list of approved hospitals is determined by p, corresponding RBAC policies can be created to allow access of p's EHRs across hospitals, cities or states by participants with a certain role. As a general rule, RBAC policies related to accessing a patient's EHRs across hospitals, cities or states are stored in their corresponding blockchains in HBN, CBN or SBN as an HR_{ACP}, a CR_{ACP} or an SR_{ACP}, respectively. Below is an example policy C1 that can be stored as a CR_{ACP} in a CBN city blockchain.

```
policy C1 {
  summary: William Johnson #001 from Hospital1 (City1) is allowed access to
     John Smith #002's EHRs in Hospital2 (City2).
  hospitals: City1.Hospital1; City2.Hospital2
  role: doctor (William Johnson #001), patient (John Smith #002)
  condition: doctor∈City1.Hospital1 && patient∈City2.Hospital2
  conclusion: approved by patient
  owners: βCIT-City1; βCIT-City2
  expiration: 10/29/2026
}
```

Policy C1 specifies that Doctor William Johnson #001 from Hospital1 (City1) is allowed to access John Smith #002's EHRs in Hospital2 (City2). When hospital super peer agent $\beta_{HOS\text{-}Hospital1}$ on behalf of Doctor William Johnson #001 makes a request to hospital super peer agent $\beta_{HOS\text{-}Hospital2}$ to access patient John Smith #002's EHRs, $\beta_{HOS\text{-}Hospital2}$ consults with its city super peer agent $\beta_{CIT\text{-}City2}$ to verify access control policies stored in its CR_{ACP}. Since policy C1 has been stored as a CR_{ACP} in $\beta_{CIT\text{-}City2}$'s city blockchain within a CBN, $\beta_{CIT\text{-}City2}$ approves this access request. A state access control policy SR_{ACP} is similar to a city access control policy CR_{ACP}, but it specifies both the city and the state where the hospital is located. Below is an example policy S1 that can be stored as an SR_{ACP} in an SBN state blockchain.

policy S1 {
summary: William Johnson #001 from Hospital1 (City1, State1) is allowed access to John Smith #002's EHRs in Hospital3 (City3, State3).
hospital: State1.City1.Hospital1; State3.City3.Hospital3
role: doctor (William Johnson #001), patient (John Smith #002)
condition: doctor∈ State1.City1.Hospital1 && patient∈State3.City3.Hospital3
conclusion: approved by patient
owners: $\beta_{STA\text{-}State1}$; $\beta_{STA\text{-}State3}$
expiration: 04/03/2025
}

Policy S1 specifies that Doctor William Johnson #001 from Hospital1 (City1, State1) is allowed to access John Smith #002's EHRs stored in Hospital3 (City3, State3). When hospital super peer agent $\beta_{HOS\text{-}Hospital1}$ makes a request on behalf of Doctor William Johnson #001 to hospital super peer agent $\beta_{HOS\text{-}Hospital3}$ to access patient John Smith #002's EHRs, $\beta_{HOS\text{-}Hospital3}$ consults with its city super peer agent $\beta_{CIT\text{-}City3}$, who then consults with its state super peer agent $\beta_{STA\text{-}State3}$ to verify access control policies stored in its SR_{ACP}. In this case, $\beta_{STA\text{-}State3}$ can approve this access request as state access control policy S1 has been stored as an SR_{ACP} in $\beta_{STA\text{-}State3}$'s blockchain within an SBN,

Since a hospital access control policy HR_{ACP} specifies access rights within the same city, neither city nor state information is required for a hospital location. For some examples of hospital access control policies, refer to earlier work [2].

Note that an access control policy for accessing a patient's EHRs across hospitals, cities or states has multiple owners. For example, policy C1 is owned by both $\beta_{CIT\text{-}City1}$ and $\beta_{CIT\text{-}City2}$; thus, the policy is duplicated and stored as encrypted CR_{ACPs} of both $\beta_{CIT\text{-}City1}$ and $\beta_{CIT\text{-}City2}$ in their CBN city blockchain.

C. Retrieving the EHRs

Once access control policies for approved hospitals are established in HBNs, CBNs and the SBN, access requests for patient EHRs can be made by a regular peer and authenticated by the corresponding super peer agents. The process starts with a requestor (e.g., a doctor) who seeks to get access to a patient's EHRs by sending a request to its hospital super peer agent β_{HOS}. Agent β_{HOS} validates this request by checking if any relevant policies exist in its HR_{ACP}. If the access will be across the cities or states, agent β_{HOS} needs to ask its city / state super peer agent for relevant policies in their CR_{ACP} / SR_{ACP}. If relevant policies do not exist, β_{HOS} may request the creation of RBAC policies as described in Section IV.B; otherwise, agent β_{HOS} forwards the request to all hospital super peer agents associated with the approved hospitals listed in the policy records to retrieve the patient's EHRs stored in their HBNs. For each hospital super peer agent who receives the request from agent β_{HOS}, permission checks are performed based on the policy records HR_{ACP}, CR_{ACP} or SR_{ACP} stored in its corresponding blockchain in HBN, CBN or SBN. If the requested access is granted, the hospital super peer agent begins extracting the requested EHRs from its CHB and returns the access link to the patient's EHRs to agent β_{HOS}, who forwards the link to the original requestor.

V. Case Study

To demonstrate the feasibility and efficiency of our proposed approach, we conducted experiments to simulate the interactions between participants from the hierarchical blockchain networks and evaluate their performance. The experiment environment consists of multiple identical computers connected under the same domain network. The computer specifications are Intel® Core™ i7-4790k CPU @ 3.60GHz (4 CPU Cores); 16 GB RAM, Windows 10 OS (64-bit, x64-based processor); and 256 SSD Hard Drive. The domain network used in this case study has a recorded Internet speed of 600 Mbps.

A. EHRs Search Time Across All Networks

In the first experiment, we simulate the concurrent EHRs search process described in Section VI.A. We record and analyze the time taken for the search process based on the number of hospital visits made by a patient in a given period. The search time includes the moment when a requestor submits a search request to its hospital super peer agent β_{HOS} until the moment when agent β_{HOS} returns to the requestor the results of the search request collected from all other super peer agents. We assume that a hospital visit made by a patient always generates an EHR that is added to a hospital blockchain. Each hospital super peer agent maintains a separate local index file for efficient responses to any EHR-related inquiry. The experiment is conducted under three different scopes, which are searches within a city, searches within a state, and searches within a country. For searches within a city, we simulate a random number of hospital super peer agents β_{HOSs} with a range of [10, 30], along with a single city super peer agent β_{CIT} that facilitates the concurrent search. For searches within a state, we simulate a random number of city and hospital agents, where the ranges for city super peer agents β_{CITs} and hospital super peer agents β_{HOSs} are [100, 500] and [10, 30], respectively. Additionally, a single state super peer agent β_{STA} are also included to facilitate the concurrent search. Finally, for searches within a country, we simulate 50 state super peer agents β_{STAs} with the same ranges for the numbers of city and hospital super peer agents as stated previously. Fig. 4 shows the results of various test cases conducted under multiple different scopes and settings.

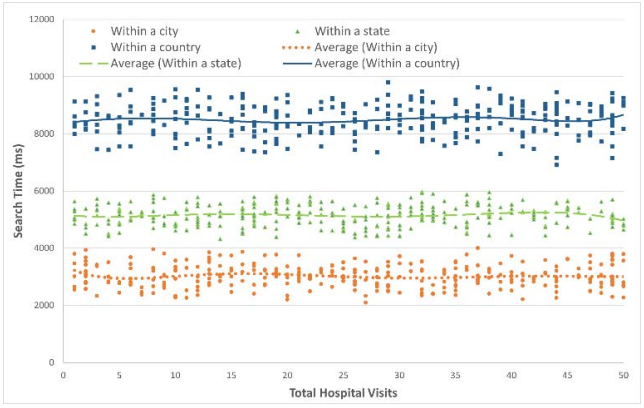

Fig. 4. Search time for a patient's EHRs of varying scope

From the figure, we can see that searches made within a smaller scope (city) have shorter search time when compared to searches made within a larger scope (state and country). This can be attributed to the increasing cost of overhead involved when additional blockchain network layers are involved in the search process. Additional observation of the figure also shows that the search time remains relatively constant regardless of the number of hospital visits made by a patient. Several factors, such

as the use of separate index files to track patients' EHRs for quick responses, and the small size of the metadata collected during the concurrent search process, contribute significantly to this stability.

B. Access Control Policy Generation and Addition

In this experiment, we simulate the creation and addition of access control policies to a blockchain for all relevant parties. We record and analyze the total time taken based on the number of policies created and added to the blockchain. The first phase of the timing process consists of disseminating data from a source hospital super peer agent $\beta_{HOS\text{-}S}$ to all relevant hospital, city, or state super peer agents and creating new HR_{ACP}, CR_{ACP}, or SR_{ACP} records from the disseminated data. The second phase of the timing process consists of adding newly created HR_{ACP}, CR_{ACP}, or SR_{ACP} records to the relevant hospital, city, or state blockchain through a consensus process described in Algorithm 1 and earlier work [2]. We assume that these processes are carried out continuously, with little or no delay between the completion of each process. Fig. 5 shows the experimental results, where the experiment is conducted under three different scenarios for creating and adding access control records to a blockchain, i.e., HR_{ACP} records only, CR_{ACP} records only, and SR_{ACP} records only.

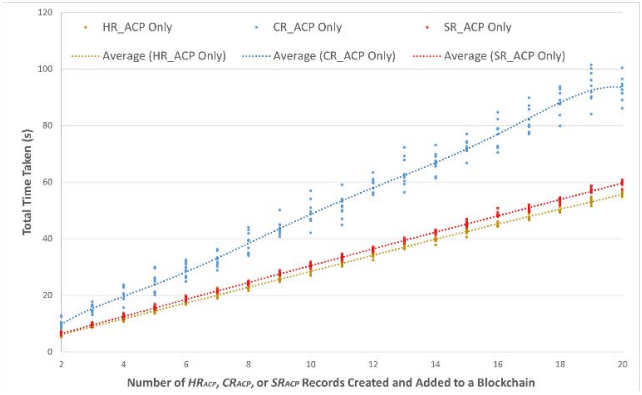

Fig. 5. Time taken to create and add HR_{ACP}, CR_{ACP}, or SR_{ACP} policy records

As in the first experiment, the number of super peer agents involved in each scenario is randomized within a specified range. For the HR_{ACP} only scenario, there is a range of [10, 30] hospital super peer agents β_{HOSs}, including $\beta_{HOS\text{-}S}$. For the CR_{ACP} only scenario, there is a range of [100, 500] city super peer agents β_{CITs}, including the $\beta_{HOS\text{-}S}$'s city super peer agent β_{CIT}. For the SR_{ACP} only scenario, there are 50 state super peer agents β_{STAs}, including $\beta_{HOS\text{-}S}$'s state super peer agent β_{STA}. From the figure, we can see that in all three scenarios, the time taken to create and add the policy records increases at a steady rate as the number of policy records involved increases. This can be mostly attributed to the second phase of the experiment to add the policy records to a blockchain; while not shown in the figure, the time recorded in the first phase remains relatively small in the overall figure. In our approach, each super peer agent is responsible for initiating its own consensus process to add newly created records to the blockchain. The number of policies created is equal to the number of consensus processes required to add those policies to the blockchain, which adds more time. Since we only deal with scenarios involving HR_{ACP}, CR_{ACP}, or SR_{ACP} only, each consensus process must take place in the same blockchain network, i.e., an HBN, a CBN, or an SBN. This means no concurrent consensus processes occur between multiple blockchain networks. Furthermore, we observe that in the CR_{ACP} only scenario, creating and adding CR_{ACP} records take the longest time compared to the time needed for creating and adding SR_{ACP} only records and HR_{ACP} only records. This is due to the significantly larger potential number of city super peer agents β_{CITs} (i.e., 100 to 500) compared to 50 state super peer agents β_{STAs} and 10-30 hospital super peer agents β_{HOSs}. This greater number of city super peer agents results in a greater overhead of time spent in the consensus process. It should be noted that the scenarios presented in this experiment are theoretical and are unlikely to occur in a real-world scenario. There would most likely be a mix of HR_{ACP}, CR_{ACP}, and SR_{ACP} generated from an initial search request. Thus, in real-world situations, the overall time taken can be reduced as adding HR_{ACP}, CR_{ACP}, and SR_{ACP} to their corresponding blockchains can be parallelized in different blockchain networks.

C. EHRs Retrieval Time in Hierarchical Blockchains

Finally, we simulate EHRs retrieval from a hospital within a city, a state, and a country. In this experiment, a source hospital super peer agent $\beta_{HOS\text{-}S}$ initiates an EHRs retrieval request on behalf of a regular peer agent to a target hospital super peer agent $\beta_{HOS\text{-}T}$. We record and analyze the total time taken to retrieve a patient's EHRs vs. the number of EHRs that are to be retrieved. The total time taken includes the time for a regular peer's retrieval request to be validated by $\beta_{HOS\text{-}S}$ and $\beta_{HOS\text{-}T}$ and the time to download and decrypt the requested EHRs stored with the target hospital. In our current design, a block can contain only one HR_{MER} record per patient. Therefore, retrieving multiple EHRs for a patient requires extracting multiple HR_{MER} from multiple independent blocks of the target hospital's CHB. To avoid long download time, multimedia files are allowed to be downloaded concurrently. In this experiment, we randomize the sizes of the multimedia files within a range of [100, 1000] MB. Fig. 6 shows the experimental results for EHRs retrieval from a hospital within a city, a state, and a country.

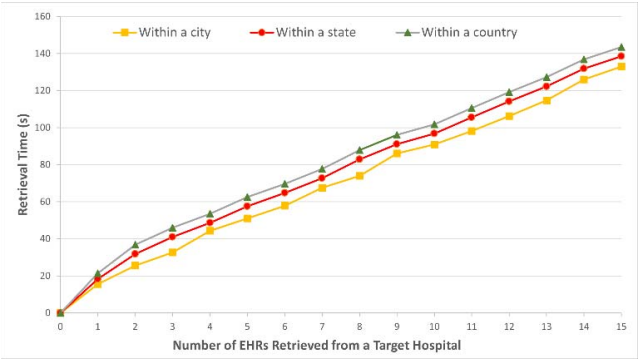

Fig. 6. Retrieval time for a patient's EHRs from a target hospital

From the figure, we can see that in all three cases, the time required to retrieve a certain amount of EHRs increases at a steady rate as the number of EHRs to be retrieved increases.

This is due to the additional overhead cost of handling concurrent retrieval of EHRs, and the increased time required agent $\beta_{HOS\text{-}S}$ to decrypt more EHRs. In addition, the retrieval time for EHRs from a target hospital across different cities and states has a slightly higher overhead compared to the retrieval time for EHRs from a target hospital within the same city. This is because request from a hospital agent with a different HBN or different CBN will need to be checked for permissions through CR_{ACP} or SR_{ACP} records in its city or state super peer agent's blockchain. Additional network interactions also incur more overhead costs and add more processing time to the overall process. Nevertheless, the total time to retrieve EHRs within a city, a state and a country is very close. Based on the results collected from Fig. 6, we can conclude that our hierarchical approach does not have a significant negative impact on the retrieval process of EHRs, regardless of the geographical difference in the locations of the agents $\beta_{HOS\text{-}S}$ and $\beta_{HOS\text{-}T}$. It is worth noting that the retrieval time can be further improved by establishing better network bandwidth to hasten the download process of EHRs and by employing better hardware to improve and speed up the decryption process.

VI. Conclusions and Future work

The cloud-based blockchain scheme in previous work [2] allows storing EHRs in the blockchain itself without significant costs to end users. However, the approach is limited to a small number of hospital participants in the blockchain network to operate effectively and efficiently. In this paper, we explore a novel approach to solving the above scalability problem using a hierarchical architecture of blockchain networks. We introduce a three-layer hierarchical blockchain structure, which consists of hospital, city, and state layers. The hospital layer consists of multiple HBNs, where each HBN represents a city-wide blockchain network with hospital super peer agents and regular peer participants. The city layer consists of multiple CBNs, where each CBN represents a statewide blockchain network with city super peer participants. The state layer consists of only one SBN, representing a country-wide blockchain network with state super peer participants. Participants at the city and state layer act as network regulators, facilitating communication and interaction between participants from lower layers. As the experimental results show, this approach allows all hospitals and peers at the hospital layer to interact and share data with each other in an effective and efficient manner, regardless of which blockchain networks they belong to.

In future work, we plan to design a more secure and reliable approach that can withstand real-world attack scenarios such as insider threat, DDOS attacks, and so on. This can be done by testing the consensus process of our approach and the capabilities of the cryptographic procedures, based on potential attacks, and improving them as needed. In addition, we plan to further improve the space/memory efficiency of our approach by implementing temporary blocks for timely publication of new blocks, where smaller temporary blocks can be merged into a larger permanent block in a cost-effective manner when a certain threshold is reached [16]. This will allow individual EHR data, including access control policy records and access records, to be quickly posted to the blockchains.

References

[1] M. T. de Oliveira, L. H. A. Reis, R. C. Carrano *et al.*, "Towards a Blockchain-Based Secure Electronic Medical Record for Healthcare Applications," In *Proceedings of the 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019, pp. 1-6.

[2] A. Thamrin and H. Xu, "Cloud-Based Blockchains for Secure and Reliable Big Data Storage Service in Healthcare Systems," In *Proceedings of the 15th IEEE International Conference on Service-Oriented System Engineering (IEEE SOSE 2021)*, Oxford Brookes University, UK, August 23-26, 2021, pp. 81-89.

[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2009. Retrieved on March 5, 2021 from https://bitcoin.org/bitcoin.pdf

[4] O. Dib, K.-L. Brousmiche, A. Durand, E. Thea, and E. B. Hamida, "Consortium Blockchains: Overview, Applications and Challenges," *International Journal on Advances in Telecommunications*, Vol. 11, No. 1 & 2, 2018, pp. 51-64.

[5] Z. Su, H. Wang, H. Wang, and X. Shi, "A Financial Data Security Sharing Solution Based on Blockchain Technology and Proxy Re-encryption Technology," In *Proceeedings of the IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, November 28-30, 2020, Chongqing City, China, pp. 462-465.

[6] Y. Jeong, D. Hwang, and K. Kim, "Blockchain-Based Management of Video Surveillance Systems," In *Proceedings of the 2019 International Conference on Information Networking (ICOIN)*, Janurary 9-11, 2019, Kuala Lumpur, Malaysia, pp. 465-468.

[7] H. Wang and Y. Song, "Secure Cloud-Based EHR System Using Attribute-Based Cryptosystem and Blockchain," *Journal of Medical Systems*, Vol. 42, Article number: 152, August 2018, pp. 1-9.

[8] L. Cui, S. Yang, Z. Chen, Y. Pan, M. Xu, and K. Xu, "An Efficient and Compacted DAG-Based Blockchain Protocol for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 6, June 2020, pp. 4134-4145.

[9] A. Fernandes, V. Rocha, A. F. d. Conceicao, and F. Horita, "Scalable Architecture for Sharing EHR Using the Hyperledger Blockchain," In *Proceedings of the IEEE International Conference on Software Architecture Companion (ICSA-C)*, Salvador, Brazil, March 16-20, 2020, pp. 130-138.

[10] B. S. Egala, A. K. Pradhan, V. R. Badarla, and S. P. Mohanty, "Fortified-Chain: A Blockchain Based Framework for Security and Privacy Assured Internet of Medical Things with Effective Access Control," *IEEE Internet of Things Journal*, Vol. 8, No. 14, July 2021, pp. 11717-11731.

[11] A. Buzachis, A. Celesti, M. Fazio, and M. Villari, "On the Design of a Blockchain-as-a-Service-Based Health Information Exchange (BaaS-HIE) System for Patient Monitoring," In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, Barcelona, Spain, June 29-July 3, 2019, pp. 1-6.

[12] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du and M. Guizani, "MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain," *IEEE Access*, Vol. 5, 2017, pp. 14757-14767.

[13] H. Guo, W. Li, E. Meamari, C. Shen, and M. Nejad, "Attribute-Based Multi-Signature and Encryption for EHR Management: A Blockchain-Based Solution," In *Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Toronto, ON, Canada, May 2020, pp. 1-5.

[14] R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, "Role-Based Access Control Models," *IEEE Computer*, Vol. 29, No. 2, February 1996, pp. 38-47.

[15] M. Meingast, T. Roosta and S. Sastry, "Security and Privacy Issues with Health Care Information Technology," In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, New York, NY, USA, August 30-September 3, 2006, pp. 5453-5458.

[16] R. Ming and H. Xu, "Timely Publication of Transaction Records in a Private Blockchain," In *2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, IEEE International Workshop on Blockchain and Smart Contracts (IEEE BSC 2020), Macau, China, December 11-14, 2020, pp. 116-123.