appXchain: Application-Level Interoperability for Blockchain Networks

MOHAMMAD MADINE[1], **KHALED SALAH**[1], **(Senior Member, IEEE),**
RAJA JAYARAMAN[2], **YOUSOF AL-HAMMADI**[1], **JUNAID ARSHAD**[3],
AND IBRAR YAQOOB[1], **(Senior Member, IEEE)**
[1]Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[2]Department of Industrial and Systems Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[3]School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, U.K.

Corresponding author: Ibrar Yaqoob (ibrar.yaqoob@ku.ac.ae)

ABSTRACT Blockchain technology has the potential to revolutionize industries by offering decentralized, transparent, data provenance, auditable, reliable, and trustworthy features. However, cross-chain inter-operability is one of the crucial challenges preventing widespread adoption of blockchain applications. Cross-chain interoperability represents the ability for one blockchain network to interact and share data with another blockchain network. Contemporary cross-chain interoperability solutions are centralized and require re-engineering of the core blockchain stack to enable inter-communication and data sharing among heterogeneous blockchain networks. In this paper, we propose an application-based cross-chain interoperability solution named appXchain which allows blockchain networks of any architecture type and industrial focus to inter-communicate, share data, and make requests. Our solution utilizes the decentralized applications as a distributed translation layer that is capable of communicating and understanding multiple blockchain networks, thereby delegating requests and parameters among them. The architecture uses incentivized verifier nodes that maintain the integrity of shared data facilitating them to be readable by the entities of their network. We define and describe the roles and requirements of major entities of inter-operating blockchain networks in the context of healthcare. We present a detailed explanation of the sequence of interactions needed to share an Electronic Medical Record (EMR) document from one blockchain network to another along with the required algorithms. We implement the appXchain solution with Ethereum-based smart contracts for two hospitals and also present its cost and security analysis. We have made our smart contracts code and testing scripts publicly available.

INDEX TERMS Blockchain, interoperability, cross-chain interoperability, decentralized application, ethereum, healthcare.

I. INTRODUCTION

Blockchain is a prominent Distributed Ledger Technology (DLT) that is inherently decentralized, transparent, and auditable, thereby enabling more trustworthy and reliable services. The adoption of blockchain in various fields can bring disintermediation, delay reductions, fraud and error reductions, and complete provenance of decisions and events. However, as diverse applications can have varying requirements, organizations and developers need to choose the optimal blockchain architecture, such as public, private, or consortium [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Kashif Bashir.

As highlighted by a recent study [2], blockchain has been adopted across diverse domains including finance and insurance, accommodation and food services, and healthcare and social assistance. Interestingly, the study found indications for potential interest in interoperability, as 9% of sectors are cross-industry, 88% have shared use of the blockchain network between different entities, and 73% plan to increase their collaboration with new partnerships.

Although blockchain introduces many benefits as highlighted above, contemporary implementation of blockchain-based systems can potentially create silos within organizations. Therefore, a major challenge impeding the widespread adoption of this technology is the lack of interoperability among different blockchains [3]–[5]. To this

end, cross-blockchain interoperability is envisaged to allow different blockchain networks to interact with each other and future blockchains without having to embed a pre-defined intercommunication layer in each network.

As blockchain is adopted by an increasing number of organizations and enterprises, there is a race to research and develop appropriate standards for cross-blockchain interoperability [6]. Specifically, the drivers to the development of mechanisms for blockchain interoperability are primarily rooted in scalability and integration. Homogeneous blockchain solutions that serve the same purpose, each for a specific region, can benefit from interoperability to scale-out to new stakeholders. Similarly, heterogeneous blockchain solutions can take advantage of a potential interoperability standard through integration procedure.

An important factor in the lack of cross-chain interoperability can be attributed to two fundamental aspects in modern blockchains i.e. 1) smart contracts, and 2) fragmentation. Blockchain uses smart contracts to automate interactions between stakeholders of a network. Since smart contracts are stored on the immutable ledger on the chain, they cannot be upgraded, and therefore even if a group of blockchain systems is manually integrated, they cannot support future solutions without a total re-write of the smart contract. Furthermore, as a result of the rapid development within blockchain technology, there is a fragmentation in the types of technologies blockchain systems use, such as the architecture type, the development framework, and the consensus algorithm. Developing standards to facilitate interoperability among existing variations of blockchain, in addition to future possible ones, is a great challenge.

A. BLOCKCHAIN INTEROPERABILITY GOALS

A basic blockchain interoperability solution is envisaged to allow heterogeneous blockchain-based systems to interact and share data; however, for the solution to be practical, it must achieve the following goals.

- The solution architecture must support various types of blockchain systems, such as public or private architectures (Ethereum, Hyperledger Fabric, etc.).
- The solution must not cause a major disruption to the blockchain network, such as by requiring repeated forking or smart contract modification for every new intercommunication link with a network.
- No manual interference by the end-users should be required.
- The dependence on off-chain infrastructures and systems shall be kept to a minimum, and in the case of usage of any off-chain approach, the off-chain entities cannot be naively trusted with their responses.
- The solution must not affect the performance or the security of the blockchain networks.

B. APPROACH AND CONTRIBUTIONS

This paper aims to design and develop an architecture for cross-chain interoperability that can be adopted in a wide range of applications and use cases. The appXchain solution allows data sharing and interaction across different blockchain systems with the ability to provide interoperability support for additional systems in a seamless manner. In particular, we propose application-level interoperability for blockchain networks, taking advantage of the adaptability and upgradability of Decentralized Applications (DApp) to develop a practical and standardized solution for cross-chain communication.

To assess the effectiveness of the appXchain solution, we use a blockchain-based healthcare system as a use case. The healthcare industry has greatly benefited from blockchain adoption and can potentially benefit even further through scaling out and utilizing data sharing opportunities achieved by cross-chain interoperability. In this respect, [7] showcases the significant impact of blockchain in healthcare, such as through Electronic Health Records (EHR) to be securely stored and accessed in a distributed and decentralized manner. Moreover, the authors also highlight how the lack of interoperability can be a barrier to blockchain adoption in the healthcare sector.

Blockchain systems for managing EHR data and giving patients control over their data typically have a private architecture. The systems are regulated by a leading entity, such as the hospital or the department of health, which is responsible for deploying the system smart contracts and verifying the identity of the stakeholders, such as the patients and the doctors.

Major contributions of this paper can be summarized as follows:

- We propose a blockchain interoperability solution based on decentralized applications, which facilitates cross-blockchain communication, data sharing, and interaction with no end-user intervention.
- We design and develop a standardized system architecture for interoperable blockchain networks that is fully-automated, secure, trustworthy, and platform-independent.
- We adapt the interoperability standard to a healthcare use case and incorporate the appXchain solution with a detailed blockchain-based patient-centered EHR management system.
- We implement functional smart contracts of hospital-regulated healthcare systems, deploy the smart contracts, and perform extensive experiments and tests to analyze the functionality of our algorithms. The smart contract code along with testing scripts is publicly made available.[1]
- We analyze the cost and performance of the appXchain solution, and assess the security of blockchain networks adopting our interoperability standard.

The remainder of the paper is organized as follows. Section II presents related work. Section III describes the individual system components and the overall architecture

---

[1]https://github.com/anon18012021/blockchain-interoperability

and data flow. In Section IV, we present algorithms and functions that we later implement for deep solution analysis. In Section V, we evaluate the appXchain solution in terms of use case functionality and cost. In Section VI, we discuss the security, challenges, and adoption of appXchain, in addition to comparing it with existing solutions. We summarize our findings in Section VII.

II. RELATED WORK

One of the earliest and most prominent works on blockchain interoperability is [8]. Vitalik Buterin is the founder of the blockchain Ethereum framework and he defined three strategies for approaching interoperability in this paper. The first is called a notary scheme, in which a trustworthy set of entities allow atomic interaction and information sharing across multiple blockchains, acting as intermediaries. The second strategy is referred to as relay, also known as sidechain, which requests one of the blockchains to be responsible for verifying the claims and information of another blockchain. Finally, the third proposed strategy is hash-locking, which inter-locks multiple operations on different blockchains using the original message of a hash, thus ensuring all interactions refer to the same initial request by the end-user in a verifiable manner. Although the suggested strategies can provide interoperability in certain use cases, practically they fall short of being a standard that stimulates scalability and maintains security of the network. For example, in the notary scheme strategy, the intermediaries must be blindly trusted by all blockchains; whereas, in the relay strategy the blockchain responsible for verifying information and transactions can be seen as a point of centralization. The study conducted in [9] focuses on enabling interoperability across different blockchain networks. Also, it has demonstrated a proof-of-concept for data sharing between two independent blockchain networks based on Hyper ledger Fabric. In [10], the authors proposed a platform named HyperService to deliver interoperability and programmability features across heterogeneous blockchain networks. The proposed framework is based on two designs: a developer-facing programming framework, and (ii) a secure blockchain-facing cryptography protocol. The evaluation results showed that the proposed solution is efficient in terms of latency and scalability. The authors in [11] have discussed that most of the existing blockchain-based solutions are vendor-locked. Coping with this issue, they have proposed a unified blockchain as a service (uBaaS) platform that aims to support both the design and deployment of blockchain-based applications. The proposed solution was evaluated using the real-world use case scenario in terms of its feasibility and scalability.

Over the few years since the emergence of Ethereum and Hyperledger, further research works have been dedicated to devising an interoperability standard that satisfies a wide range of networks. The study conducted in [12] describes the need for a generic inter-blockchain communication protocol that can exchange arbitrary data, such as tokens and smart contract interactions, between blockchains in a decentralized

and trustworthy manner. On the other hand, the researchers in [13] claimed that interoperability is impossible under the classical definition of blockchain, and thus require other mechanisms, such as game theory to prevent an interface from misbehaving.

Hashed Time-Locked Contracts (HTLC) are smart contracts that provide cross-blockchain atomic transactions by ensuring that two locked transactions are either executed or canceled after a predefined timeout [14]. Further, [15] proposes a scalable and secure HTLC that uses multi-hop locks. Disadvantages of the HTLC approach include having to provide long timeout periods during which an adversary can decide whether or not to execute their transaction based on an updated state of the networks, and allowing the high cost and complexity of the design as it requires each blockchain network to understand the hash, attributes, and smart contracts of the other network [16]. In [17], the authors introduced Interledger Protocol (ILP) as a combination of two of the strategies laid out by [8]. The ILP adopts a notary scheme and hash-locking strategies to keep the intermediary entities trustworthy and incentivized in the payment system. Although the ILP can be expanded beyond payment-focused applications, it requires understanding between the parties of the networks and demands high costs for cryptographic operations [18], [19].

Cosmos and Polkadot are unique solutions that bring interoperability to the blockchain. The solutions share the same fundamental concept of creating an interconnected network of blockchains that can understand each other [20], [21]. As it can be expected from the approach, blockchains are required to be built specifically on top of the Cosmos or Polkadot network, limiting the interoperability feature to a unique number of projects, and demanding additional skill sets from the enterprise developers.

Among the blockchain interoperability efforts, researchers have developed properties and requirements that are sought-after in the healthcare industry. The study conducted in [22] suggests that DApps must have evolvability and minimal integration complexity to address blockchain-based healthcare interoperability, and concludes that an interoperable solution must have a flexible design, use minimal resource duplication, and scalability. Additionally, the authors in [23] defined healthcare interoperability as exchanging information across multiple systems at three levels: 1) Foundational, in which receiving party does not have to interpret data. 2) Structural, in which data uses predefined formats. 3) Semantic, in which there are syntax and semantic data exchange and interpretation. Within blockchain-based healthcare applications, [24] proposes an interoperable Open-Pharma blockchain solution, which uses public Ethereum blockchain as an interoperability layer between existing vendor solutions. The blockchain is used to store encrypted patient information URLs and share them with doctors who obtain the patient's encrypted authentication ID. The solution proposed is not suitable for interoperability across multiple patient-centered solutions. Patients must share an
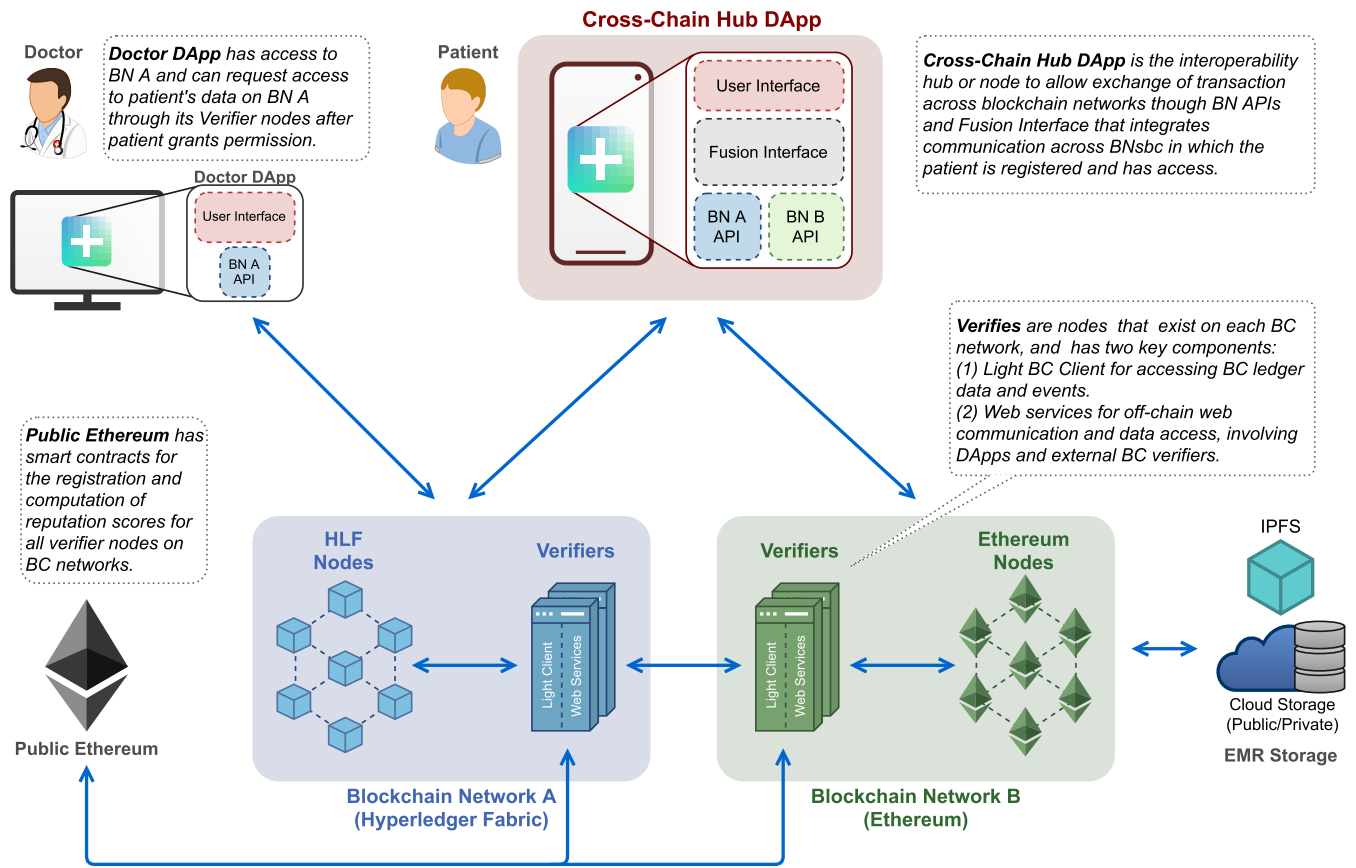
FIGURE 1. An overview diagram of blockchain networks interoperating through the patient's decentralized application.

inexhaustible encrypted authentication ID with their doctors, which if stolen, can be used by any other member. Moreover, the architecture relies on centralized services, such as SAAVHA and Amazon Web Services - Key Management Service (AWS-KMS) for membership authentication and encryption [25].

To the best of our knowledge, none of the aforementioned cross-chain interoperability solutions are capable in providing interaction and data sharing across framework-independent and domain-neutral blockchain networks with no user intervention and point of centralization. The earliest category of research, such as HTLC and multi-hop locks, mainly introduced a solution for guaranteeing the integrity of transactions, but do not provide a wide range of capabilities and use cases. On the other hand, more recent and developed research can offer secure and versatile solutions, but at the cost of high complexity and discouraging requirements for enterprises and high-scale projects.

III. DAPP-BASED CROSS-BLOCKCHAIN INTEROPERABILITY SOLUTION

In this section, we present the appXchain solution for cross-chain interoperability. To achieve a real-world context, we explain the appXchain system from the perspective of healthcare and patient medical records management.

A high-level overview of the appXchain system is presented in Figure 1 whereas detailed description of the major elements of the system along with detailed examples of the sequence of interactions in the system are presented below.

A. BLOCKCHAIN
The characteristics of a blockchain network depend on its specific platform, which can vary depending on the requirements of the application domain. Historically, blockchain-based healthcare applications have been developed as a fully decentralized public Ethereum blockchain with pseudonym patients, or as a hospital-regulated private HLF or Quorum blockchain with identifiable patients [26]–[28].

1) ETHEREUM AND SMART CONTRACTS
Along with the core blockchain ledger, Ethereum provides a programmable interface to the ledger through the use of smart contracts. Ethereum operates on the Ether cryptocurrency, which can be used for asset transfer through tokens and paying public Ethereum Virtual Machine (EVM) nodes for the execution of programs in smart contracts [26]. The complexity of Ethereum smart contracts is measured in gas units valued based on Gwei units, where 1 wei = 10^{-18} Ether. The gas measurement helps in evaluating the cost of fees when a smart contract transaction is executed, where a transaction

computation requires certain gas units which are converted into Gwei units based on the gas price, and then into USD based on the Ether price.

2) HYPERLEDGER FABRIC

Hyperledger Fabric is a project initiated by the Linux Foundation as a permissioned, programmable, and confidential blockchain framework. HLF, unlike Bitcoin and Ethereum, does not operate on a cryptocurrency, nor does it depend on Proof Of Work (PoW) for consensus. A typical HLF network has 3 components: 1) *Certificate authority*, to register identities, and issue and revoke enrollment certificates. 2) *Peer*, an endorser or committer, to update or query ledgers. 3) *Orderer*, to provide the order of transactions, create blocks, and process transactions before committers.

The consensus in HLF can be decided by the network regulator or reached with an agreement protocol, such as Practical Byzantine Fault Tolerance (PBFT) where two or more parties agree on the result [29].

B. VERIFIER NODES

Verifiers are nodes registered by the blockchain network that act as lightweight clients for accessing blockchain ledger data and events. Additionally, verifier nodes run a web service for off-chain web communication and data access. Moreover, depending on the design of the EMR storage, the verifier may need to process the patient medical records to translate them from the patient to the doctor. For instance, a patient-centered blockchain system may require Proxy Re-encryption (PRE) processing of the EMR to convert the records from being encrypted by the public key of the patient to being encrypted by the public key of the doctor [30].

C. REPUTATION SYSTEMS

Considering that verifier nodes are not necessarily trusted computing devices, their actions must be regulated to incentivize appropriate behavior and discourage malicious behavior. For secure interactions and message transfer across the blockchains, our architecture requires internal and external reputation systems.

1) INTERNAL REPUTATION SYSTEM

Blockchain networks that use public verifier nodes must implement an internal reputation system as means to encourage correct and quick results by the verifiers. Depending on the use case, the reputation score of a verifier can be either determined computationally by the smart contract or given as a rating by the entity requesting the service. An automatic score is possible if the proof of a truthful and quality result can be determined computationally, such as requiring the original plaintext message of a cryptographic hash stored in the blockchain. If automatic scoring is not possible, the reputation system can rely on the entities that the verifier interacts with to assign a rating for the interaction.

In our approach, we evaluate the rating each verification response as the product of the validity of the response by

its delay. The validity is the binary result of comparing the hash of the plaintext token against the its actual stored hash, and the verification delay is the result of linearly mapping the time difference between the response and the request to a predefined minimum and maximum values, which normally would result in a higher rating for smaller delays. The rating is then used in dynamically updating average verifier score, in addition to choosing the most reputable verifier based on the product of the current response rating and the square of the average verifier score. More details about computing the rating, verification score, and updated average are in the implementation section.

2) PUBLIC ETHEREUM REPUTATION SYSTEM

To keep track of the scores of all verifiers across various blockchain networks, a public Ethereum-based smart contract stores the details of registration of each verifier, in addition to its average reputation score and count of ratings it has received. These attributes stored about verifiers are used for choosing the most reputable verifier among participants requesting to verify external requests. The choice of the most reputable verifier need to take into consideration giving a small advantage to verifiers of slightly lower average score but a much greater number of successfully performed verifications.

D. EMR STORAGE

Blockchain frameworks are incapable of storing large documents. Therefore, blockchain-based solutions typically resolve this weakness by storing such data off-chain and including a link to this data on-chain. By default, off-chain storage solutions do not encrypt the data, and therefore, there is a need to encrypt the data before uploading it to the EMR storage. Fully decentralized blockchain solutions optimally utilize decentralized storage systems, such as InterPlanetary File System (IPFS). IPFS stores files in a peer-to-peer network of public nodes, and provides content-based addressing based on the file's SHA-256 message digest, making it easy to establish connections between the blockchain and the storage [31]. Other solutions can adopt private storage systems, such as a private cloud-based database to store EMR data.

To achieve strong cross-chain interoperability levels, the blockchain networks should use a widely adopted and familiar standard for file formats. For example, a popular standard for healthcare and EMR-specific file format is Health Level Seven's Fast Healthcare Interoperability Resources (HL7 - FHIR), which provides a detailed description of how electronic healthcare data should be formatted like [32]. Such standardization makes EMR documents easily readable by any entity that gets access to the files.

E. CROSS-CHAIN HUB DAPP (CCHDA)

The primary interoperability hub within our solution is the DApp, which allows the exchange of transactions across blockchain networks through each network's default Application Programming Interfaces (API), in addition to a

Fusion Interface (FI) layer that integrates all communication across the different blockchain networks. The blockchain APIs enable communication between the DApp and the blockchain networks, individually. The FI component processes cross-chain transactions, acting as a translation layer between the two blockchain networks.

F. BLOCKCHAIN ENTITY DEFINITIONS

In the context of hospital-regulated EMR management blockchains, a simplified system design must involve two blockchain networks with their verifier nodes, a patient and a doctor, and the public Ethereum reputation system. The entities are defined as follows:

- **Hospital**: A regulatory entity that controls all patients, doctors, and verifiers registered to it. A hospital is responsible for deploying its smart contracts, which receive EMR data from patients and allows doctors to request the data. In our example, two hospital entities exist, such that **Hospital A** registers a patient and a doctor, and **Hospital B** only registers the patient with their data. All hospitals must register a set of verifier nodes that perform the communication verification tasks and other optional ones as explained earlier in the section. Moreover, the hospitals are envisaged to be responsible for assessing the performance of the verifiers and sharing the public reputation scores with the public Ethereum reputation system.
- **Patient**: A client entity that is registered in hospital blockchain networks. A patient is responsible for allowing or denying the doctor from accessing their data and making self-requests of the data. The patient may have additional responsibilities depending on the design of the healthcare network. For example, a patient-centered solution may require the patient to upload their data along with a token key on IPFS, and submitting the IPFS hash to the blockchain.
- **Doctor**: A client entity that is registered in hospital blockchain networks. A doctor is responsible for requesting EMR data from patients registered in the same network. Similar to the patient, the doctor may also have additional responsibilities, such as sharing their public key with a verifier for PRE. Requiring requests to be initiated by the doctor allows the system to operate under special circumstances, such as cases of emergency or incapacitated patients, where the decision on the request needs to be taken by a third-party entity [33].

All entities must be registered in their blockchain network and have identification addresses, such as Ethereum addresses and public-private key pairs.

G. SEQUENCE OF INTERACTIONS

We present the sequence of interactions for cross-chain interoperability in the context of healthcare from the perspective of both plaintext (unencrypted) and encrypted EMR data.

The unencrypted use case simulates a scenario where a doctor requests the patient EMR data through a patient-centered Blockchain Network A (BN_A), and the patient EMR data is managed by a private Blockchain Network B (BN_B) which stores the data (unencrypted) on a private cloud storage platform. On the other hand, the encrypted use case simulates a scenario where a doctor requests patient EMR data through a patient-centered BN_A, and the patient EMR data is managed by a public patient-centered BN_B which stores the encrypted data on public IPFS.

The sequence depicted in Figure 2 shows both unencrypted and encrypted interactions among the entities, which are mostly consistent in both use cases (additional interactions in the encrypted use case are highlighted in green). In both use cases, we assume that (BN_A) has its verifiers registered along with the patient and the doctor; whereas, (BN_B) has its verifiers and patients registered. Additionally, all verifiers are registered in the Public Ethereum network. The sequence assumes the patient has submitted an EMR bundle to BN_B, and in the encrypted use case, the EMR bundle contains the EMR document and an acquisition key, the hash of which the patient submits to the network.

Step 1) The doctor generates a token and makes a request to BN_A containing the hashed token key. The network returns a request identifier as an event for future reference, which the doctor sends to the patient in an off-chain manner, along with the token key.

Step 2) The patient makes a self-request of the EMR document on behalf of the doctor to BN_B. The patient specifies the range of acceptable verifiers. The network returns the patient's request identifier as an event for future reference. The network also broadcasts the patient's request identifier to its internal verifiers informing them of a new request ready for fetching.

Step 3) Verifiers of BN_B fetch the EMR document from the EMR storage and submits proof of document acquisition to BN_B. In the unencrypted use case, the proof is the EMR document's hash, whereas in the encrypted case the proof is the token located in the acquired EMR bundle. The network verifies the correctness of the proof which is stored privately on-chain and evaluates the performance rating of the verifiers based on their correctness and speed. The rating is used to update the network's internal reputation system and the public Ethereum reputation system. The network then sends tokens to the most reputable verifier and the patient.

Step 4) The patient responds to the doctor's original request by accepting it. Since the doctor's request does not directly specify the patient identifier, the patient must include the doctor's original request token key in the response, which prevents other patients from responding to the doctor's request. BN_A then informs the doctor of the response, and broadcasts the doctor's request identifier to its internal verifiers, informing them of a new request ready for fetching and verifying. In the encrypted use case, the doctor, upon receiving the response, sends their public key to the patient in an off-chain manner.
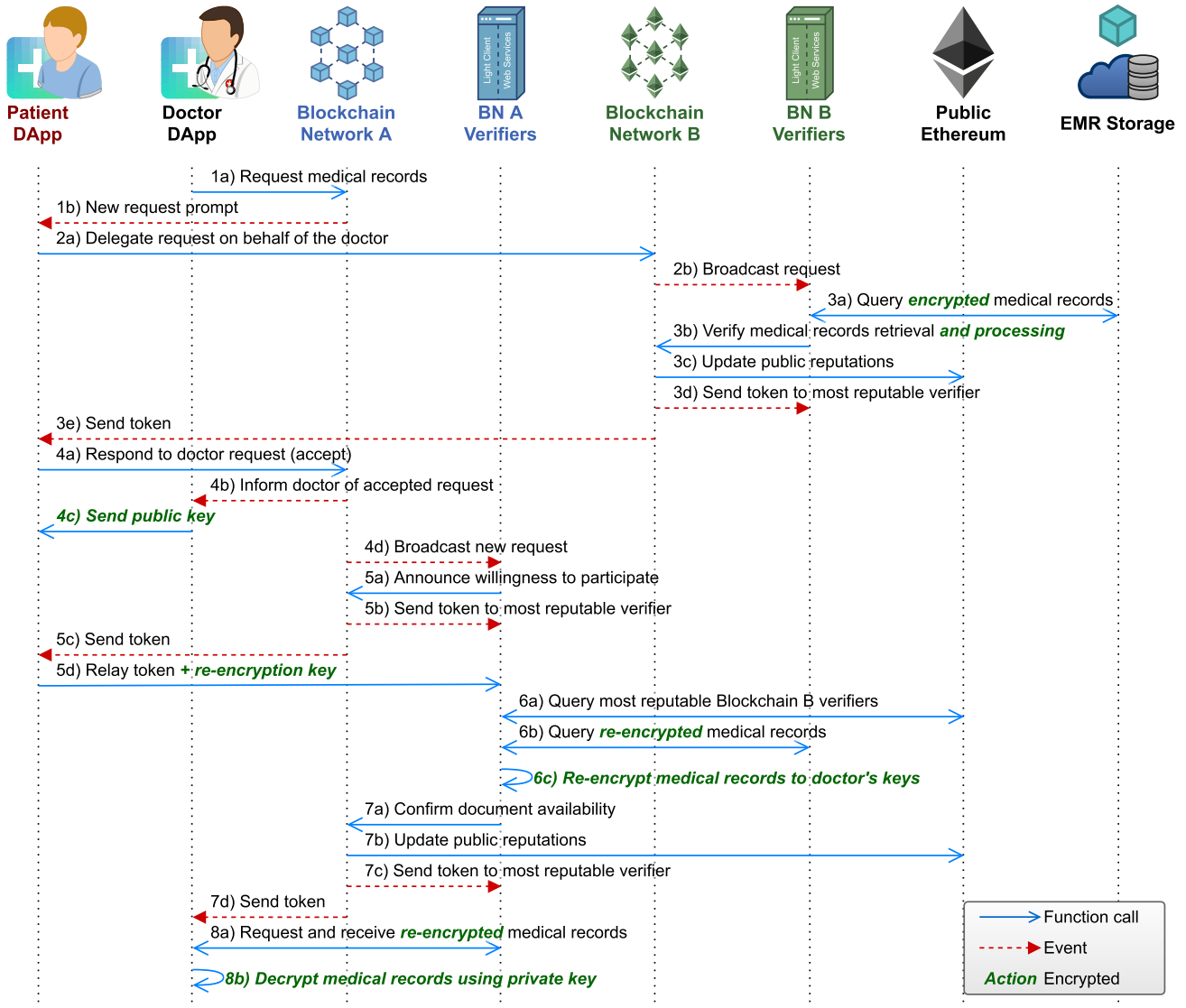
FIGURE 2. Sequence diagram of cross-chain interoperability for unencrypted and encrypted (green) EMR data sharing.

Step 5) Verifiers of BN_A announce their willingness to participate in fetching and verifying the EMR document from BN_B verifiers. After an adequate number of verifiers respond, BN_A sends tokens to the most reputable verifier and the patient. The patient at this stage communicates in an off-chain manner with the verifier, to pass on the token received from BN_B to BN_A verifier. In the encrypted use case, the patient uses their private key and the doctor's public key to create a re-encryption key, which is also sent to the BN_A verifier.

Step 6) Chosen verifier of BN_A queries the public Ethereum reputation system to ensure that the chosen verifier of BN_B is reputable. If so, the two verifiers establish a connection to send the EMR document from BN_B to BN_A. In the encrypted use case, upon completion of EMR document transfer, the verifier uses the re-encryption key to convert the

state of EMR document encryption from the patient to the doctor.

Step 7) The verifier of BN_A confirms to its network the availability of the document alongside a proof. In the unencrypted use case, the proof can be found in the form of the EMR document's hash, whereas in the encrypted use case the proof is the document's acquisition token, as described in Step 3. The network updates the reputation of the verifier in its internal records and the public Ethereum reputation system. The network then sends tokens to the verifier and the doctor for the final transfer of the EMR document.

Step 8) The doctor and the verifier use their tokens to establish a connection to transfer the EMR document from the verifier to the doctor. In the encrypted use case, the doctor decrypts the file using their private key.

TABLE 1. Definitions of notations used in the algorithms.

Notation	Definition
EA	Ethereum address
PEA	Ethereum address of the patient
DEA	Ethereum address of the doctor
VEA	Ethereum address of the verifier
EMR	EMR document
EMRL	List of EMR documents
EMRT	EMR document token
EMRTH	Hash of EMR document token
EMRA	EMR document IPFS address
R	Request
RL	List of requests
RC	Number of requests
RI	Request identifier
VL	List of verifiers of a request
RL	List of ratings of a R.LV
P	Request's number of verification participations
RTH	Hash of request token
minV	Chosen minimum number of verifiers
maxV	Chosen maximum number of verifiers
MINV	Network's minimum number of verifiers
MAXV	Network's maximum number of verifiers
L	Latency
VT	Validity of token
VR	Verification rating
VS	Verifier score
BVS	Best verifier score
VS	Verifier current score
VAS	Verifier average score
VSC	Verifier score count

IV. IMPLEMENTATION

Our implementation of a cross-chain interoperability system is based on the EMR document sharing across two hospitals. The implementation covers two smart contracts, one for each hospital. Each smart contract defines the characteristics of the main entities, which are the patient, doctor, and verifier. Additionally, the smart contracts control how the patient EMR documents, doctor requests, and verifier responses are processed. Definitions of all the notations used in the algorithms are presented in Table 1. Our algorithms are specific to the Ethereum framework as entities are assumed to have Ethereum addresses (EA). This aspect of the algorithms is generalizable to any other blockchain network that supports programmable smart contracts (e.g., Quorum, Hyperledger Besu, and Hyperledger Fabric).

Algorithm 1: Submitting an EMR Document, Performed by the Patient at Blockchain Network B

---

1 **Function** submitEMR(*DA, EMRTH*):
2   PEA ← Transaction initiator
3   Require that PEA is a patient
4   EMR ← New EMR document(PEA, EMRTH)
5   EMRL[DA] ← EMR

We designed algorithms for submitting an EMR document to the internal blockchain, requesting the document from an

external blockchain, and delegating the request by recreating the request. Algorithm 1 describes the submitEMR function, which allows a patient to enter the address of the IPFS bundle as an identifier of the EMR document, accompanied with the hash of the token placed inside the bundle.

Algorithm 2: Requesting an EMR From an External Blockchain, Performed by the Doctor at Blockchain Network a

---

1 **Function** requestExternalEMR(*RTH, minV, maxV*):
2   DEA ← Transaction initiator
3   Require that DEA is a doctor
4   minV ← max(minV, MINV)
5   maxV ← min(maxV, MAXV)
6   R ← New request(DEA, RTH, minV, maxV)
7   RL[RC] ← R
8   RC ← RC + 1

Algorithm 3: Self-Requesting an EMR Document From the Internal Blockchain, Performed by the Patient at Blockchain Network B

---

1 **Function** selfRequestEMR(*EMRA, minV, maxV*):
2   PEA ← Transaction initiator
3   EMR ← EMRL[EMRA]
4   Require that PEA = EMR.PEA
5   minV ← max(minV, MINV)
6   maxV ← min(maxV, MAXV)
7   R ← New request(PEA, EMRA, minV, maxV)
8   RL[RC] ← R
9   RC ← RC + 1
10   Emit an event to notify verifiers of the request

At a later time, a doctor can call the requestExternalEMR function defined in Algorithm 2 to request the EMR document from the patient, where the document is located in a different blockchain network. The function receives from the doctor a hashed token and the range of verifiers. The hashed token ensures that the request cannot be responded to unless the patient has a valid token, whereas the range of verifiers helps the doctor in controlling the confidence in the verification process and its cost. A range of a high number of verifiers would provide more confident responses, but for more expensive fees. As for the selfRequestEMR function explained in Algorithm 3, it performs almost the same actions as the requestExternalEMR function, with the difference being storing the IPFS address of the EMR document in the request and directly considering the request as granted and waiting to be verified.

When the EMR management functions are executed, the verifiers will have to process the requests at three steps as described in the set of algorithms executed by

---

Algorithm 4: Verifying a Request, Performed by Verifiers at Blockchain Network B

1 **Function** `verifyInternalRequest`(*RI, EMRT*):
2   VEA ← Transaction initiator
3   R ← RL[RI]
4   Require that R is granted by a patient, is internal, and not already verified
5   L ← Current time − R.time
6   **if** R can be verified **then**
7   VT ← keccak256(EMRT) == R.EMRTH
8   VR ← VT ×(map L from 0 to 1)
9   R.VL ← R.VL + VEA
10   R.RL ← R.RL + VR
11   **if** Sufficient verifications received **then**
12   BVS ← 0
13   **for** i ← 0 to length of R.VL **do**
14   VS ← R.VL[i] × (VAS[i]+1)^2
15   **if** VS > BVS **then**
16   BVS ← VS
17   VAS ← (VSC × VAS + VR)/(VSC +1)
18   Send token to the verifier of BVS and R.EA

Algorithm 5: Requesting to Participate in the External Verification Process, Performed by Verifiers at Blockchain Network a

1 **Function** `requestParticipation`(*RI*):
2   VEA ← Transaction initiator
3   R ← RL[RI]
4   Require that R is granted by a patient, is external, and not already verified
5   L ← Current time − R.time
6   **if** R can be verified **then**
7   **if** Length of R.VL = 0 **then**
8   R.VL ← Transaction initiator
9   **else**
10   BVS ← (VSC × VAS +1)/(VSC +2), where VAS and VSC are for the best verifier
11   VS ← (VSC × VAS +1)/(VSC +2), where VAS and VSC are for the new verifier
12   **if** VS > BVS **then**
13   R.VL[0] ← VEA
14   R.P ← R.P + 1
15   **if** Sufficient participations received **then**
16   Send token to the verifier of BVS and R.PEA

---

the verifier nodes only. The `verifyInternalRequest` detailed in Algorithm 4 is executed by various verifier nodes of Blockchain Network B, responding to the self-request of the patient made through `selfRequestEMR` function. The verifiers embed an identifier for the request and the original token key located inside the EMR document as part of their transaction. The function ensures that the response is acceptable by checking the validity of the token, and evaluates the performance of the verifier based on the latency and the correctness. In the algorithm, the ratings are mapped from 0 to 1, however, considering the high cost of floating-point operations, the scores were mapped from 0 to 65,535. Once the sufficient number of verification responses are received, which are evaluated based on the requester's minimum and maximum number of verifiers, the function updates the reputations of the verifiers and sends a token to the one with the highest final reputation score. The verifier's reputation is updated such that the new score is added to the old reputation average, and the reputation count is incremented by one. Updating the reputation average is computed dynamically, to minimize the stored data and processing time.

Algorithm 5 shows the `requestParticipation` function, which receives participation requests from verifiers to fetch and verify the EMR document from Blockchain Network B's verifiers. The function simply looks for the verifier of the highest reputation and sends to it and the patient tokens for them to establish a secure connection. In this function, the comparison of the reputations is not computed based on the average reputation alone, but it also combines the reputation count as given by Laplace's rule of succession, which assumes two more random ratings were given to the verifier. The two random ratings, assuming one is

---

Algorithm 6: Confirming Document Availability, Performed by Verifiers at Blockchain Network a

1 **Function** `documentAvailable`(*RI*):
2   VEA ← Transaction initiator
3   R ← RL[RI]
4   Require that R is granted by a patient, is external, and already verified
5   Require that R's external verifier = VEA
6   Send token to VEA and R.DEA

---

positive and the other is negative, result in an extra score of 1, whereas the reputation count is increased by 2. Finally, the `documentAvailable` function in Algorithm 6 is called by the verifier once it is ready to send to the doctor the re-encrypted EMR document.

V. EVALUATION

In this section, we evaluate the appXchain architecture and designed algorithms by implementing and deploying two Ethereum-based networks representing two different hospitals. Also, we verify the use case functionality and present the cost analysis.

A. QUALITATIVE FUNCTIONALITY EVALUATION

To evaluate our the appXchain system, we implemented each of the hospital blockchain network as a separate Ethereum network. The smart contracts for the two hospitals have been developed using Solidity language and Truffle framework for managing the compilation and deployment. The smart contracts, which we denote as HSC_A for hospital A and HSC_B for

hospital B were compiled using the standard Ethereum Solidity compiler version 0.7.0 with code optimization at 200 runs and deployed to local Ethereum test networks (testnets) created by Ganache from the Truffle Suite. The testnet for HSC$_A$ has 8 unique Ethereum addresses corresponding to the network admin, five verifiers, patient, and doctor. On the other hand, the testnet for HSC$_B$ contains 7 unique Ethereum addresses corresponding to the network admin, five verifiers, and patient. The testing of the code was automated using Truffle's JavaScript-based tests, allowing various use cases to be simulated with capabilities, such as time advancement and token generation.

1) Networks and Entities Preparation: Each of the administrations of the two hospitals deploy their blockchain networks and register their entities. The patient prepares a random token to be placed along with the EMR document and uploads it to IPFS. The address of the uploaded bundle is its hash. The bundle hash and the hash of the token are then submitted to HSC$_B$ by calling submitEMR.

2) Requesting EMR Document: The doctor generates a request token and gets its keccak256 hash. The hash and the range of verifiers [1, 4] are used in calling requestExternalEMR from HSC$_A$. The network processes the request to validate the doctor parameters, resulting in constructing the desired range of verifiers [2, 4] because the minimum number of acceptable verifiers by the network is 2. Upon successful processing of the request, the network returns the request identifier to the doctor as a broadcasted event. Figure 3 depicts the transaction inputs and outputs from our testing.

3) Delegating Request: The doctor informs the patient about the desired data and sends the request token to the patient. The patient then calls selfRequestEMR at HSC$_B$, specifying the EMR document IPFS address of and the desired range of verifiers, which in our tests is [2, 4]. The network processes the request and immediately broadcasts the IPFS address to its verifiers.

4) Verifying Request: HSC$_B$ verifiers obtain the EMR document files from the IPFS network. The files contain the encrypted EMR document and the original possession token identifier. The token is passed by the verifiers while calling verifyInternalRequest at HSC$_B$. In our tests, the first and third verifiers respond with invalid tokens, with the other two responding correctly. The function must additionally be executed by the fifth verifier, which acts as a time-based trigger. At this call, the function executes the *Sufficient verifications received* path, resulting in picking the verifier with the Ethereum address ending with A971 (second responding verifier) as the most reputable verifier, as shown in Figure 4.

5) Responding to Request: The patient calls respondRequest at HSC$_A$, passing the doctor's request identifier, request token, and true indicating granting the request. The network broadcasts an event of the request acceptance. Thus the doctor sends the public key to the patient. The patient uses the public key of the doctor along with his/her private key to generate a re-encryption key.
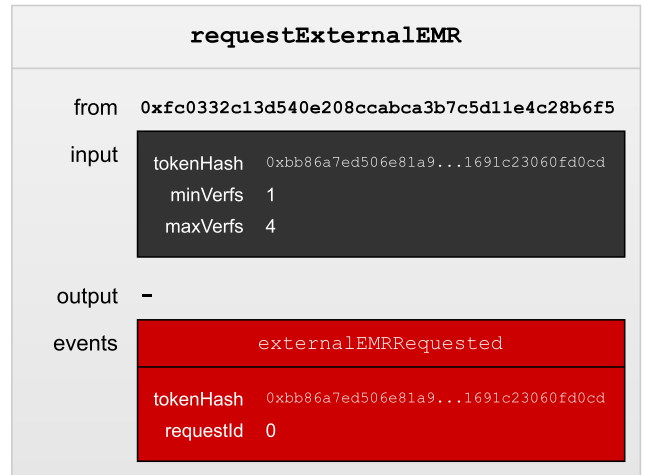

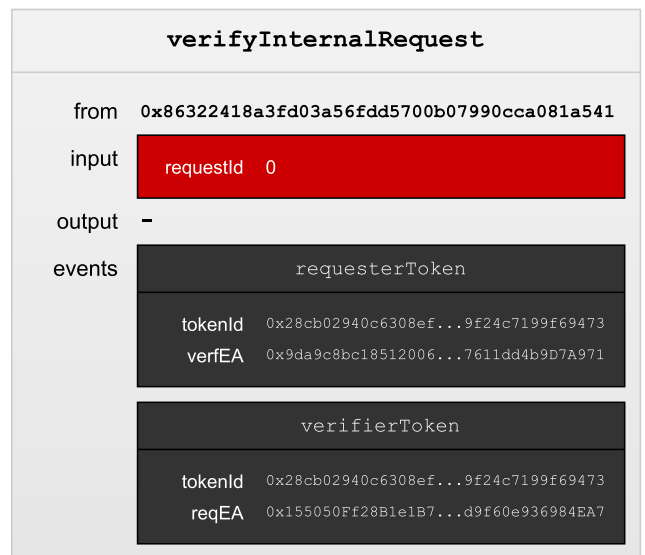**FIGURE 3.** The details of a doctor-initiated transaction to request an external EMR document.

FIGURE 4. The details of a verified internal EMR document self-request.

6) Announcing Willingness to Participate: 5 verifiers of HSC$_A$ execute requestParticipation. The fifth verifier triggers the *Sufficient participations received* path that helps to determine the most reputable verifier. In our test, all verifiers have matching reputations, and thus the network chooses the first responding verifier. The network broadcasts a communication token to the most reputable verifier and to the patient. Later, the HSC$_A$ verifier connects with the patient and receives the communication token with HSC$_B$ verifier, in addition to the re-encryption key.

7) Verifying External Interactions: The HSC$_A$ verifier queries the public Ethereum reputation system to verify the reputation of the HSC$_B$ verifier. After which the two verifiers establish a connection to transfer the EMR document. Then, the verifier of HSC$_A$ re-encrypts the EMR document to become readable by the doctor.

TABLE 2. Function caller, and gas and currency costs of HSC_A functions.

Function Caller	Function Name	Transaction Cost [Gas]	Cost [USD]
Admin A	HSC_A	1,801,931	172.99
Admin A	`registerPatient`	43,442	4.17
Admin A	`registerDoctor`	43,473	4.17
Admin A	`registerVerifier`	43,465	4.17
Doctor	`requestExternalEMR`	166,183	15.95
Patient	`respondRequest`	49,475	4.75
Verifier	`requestParticipation`	68,231	6.55
Verifier	`requestParticipation + evaluation`	81,152	7.79
Verifier	`documentAvailable`	31,177	2.99

TABLE 3. Function caller, and gas and currency costs of HSC_B functions.

Function Caller	Function Name	Transaction Cost [Gas]	Cost [USD]
Admin B	HSC_B	1,804,518	173.23
Admin B	`registerPatient`	43,442	4.17
Admin B	`registerVerifier`	43,460	4.17
Patient	`submitEMR`	64,423	6.18
Patient	`selfRequestEMR`	204,920	19.67
Verifier	`verifyInternalRequest`	99,629	9.56
Verifier	`verifyInternalRequest + evaluation`	194,788	18.70

8) Confirming Document Availability: The selected verifier of HSC_A informs the network that the verification process is complete. The network then sends a token to the verifier and the doctor.

9) Receiving and Decrypting EMR Document: The verifier node sends the EMR document to the doctor who uses the private key to decrypt the file.

B. COST ANALYSIS

Executing our cross-chain interoperability solution on local testnets enables the ability to analyze the cost of executing the smart contract functions on both networks individually. Our cost analysis is based on the scenario described in the functionality testing, at which there are 2 hospital networks, the first of which has a patient and a doctor, and the other one has only the patient with their EMR data. Throughout the evaluation, all smart contract transactions were recorded including their transaction costs. The transaction cost is measured in gas units, and it depends on the computation complexity of the function, alongside its transmission and deployment costs.

Table 2 and Table 3 present the cost of the prominent functions of HSC_A and HSC_B, respectively. The estimated costs in USD were based on the average gas price of 80 Gwei and the average Ether price of $1200, both recorded as of January 17, 2021. The functionality achieved by HSC_A includes registering entities, requesting documents, responding to requests, participating in the external verification process, and claiming that the document is available for the doctor. The function achieved by HSC_B also includes registering entities as well as submitting documents, initiating self-requests, and verifying the internal request.

A typical trend in both networks is the positive correlation between the number of modified state variables and the cost.

Besides the smart contract deployment, which is a one-time action, the functions that cost the most are the ones performed by the verifier nodes, especially when the final evaluation of all nodes is executed. The remaining functions, such as registering entities and responding to requests, have costs ranging from $3 to $7, which is reasonable considering the tenfold increase in Ethereum transaction costs over the last year.

VI. DISCUSSIONS

In this section, we discuss the appXchain in terms of its security features. Also, we compare the proposed approach with the existing solutions as well as outline some of its limitations.

A. SECURITY ASSESSMENT

Considering that appXchain is generic and adaptable to diverse blockchain systems, the security goals achieved are dependant on the security measures adopted by the underlying blockchain systems. For instance, a misconfigured and insecure blockchain system that adopts our solution would remain insecure. Therefore, as part of the analysis of the appXchain system, we investigated the security elements to determine if the support for cross-chain interoperability in the CCHDA and blockchain network would expose the systems to any security threat. In doing so, we assume that the individual blockchain is secure.

1) PRIVACY AND CONFIDENTIALITY

The appXchain system does not require participating blockchain entities to share private or sensitive information. As the data translations conducted as part of the

TABLE 4. Comparison of appXchain with existing solutions.

Study	Decentralized	Trustworthy	Integrity	Efficient	Scalable	Framework-independent	Domain-neutral
Notary scheme [8]	No	No	Yes	Yes	No	Yes	No
Relay (Sidechain) [8]	No	Yes	Yes	No	No	No	Yes
Hash-Locking + HTLC [8, 14]	Yes	Yes	Yes	Yes	No	Yes	Yes
Interledger Protocol [17]	No	No	Yes	Yes	No	Yes	No
Cosmos + Polkadot [20, 21]	No	Yes	Yes	No	Yes	No	Yes
Our Proposal (appXchain)	Yes	Yes	Yes	Yes	Yes	Yes	Yes

interoperability process are managed by the data owner entity, these do not present additional data privacy risks. Furthermore, confidential communication between the requesting and the translating entities relies on the off-chain communication link used by the two entities, such as private peer-to-peer messaging or a face-to-face exchange in real life. In the healthcare example as discussed earlier, cryptographic constructs are used to achieve appropriate protection for the patient's EMR.

2) INTEGRITY AND AUTHENTICITY
The appXchain solution does not introduce additional third parties to translate or relay information and transactions across the interoperating blockchain networks. For example, in the case of sharing an EMR document, the patient is responsible for translating the transaction and communicating with the verifier nodes to relay information, such as the re-encryption key and the token for connecting with the external verifier. Furthermore, communication between the entities involved is regulated by the blockchain network registering such entities that utilize the ability to generate tokens ensuring communication link security and authenticity of both parties.

3) AVAILABILITY AND RESILIENCY
As blockchain is an inherently peer-to-peer system, it facilitates protection against availability-focused attacks, such as Denial of Service (DoS). However, blockchain interoperability mandates trustworthy connections between the participating blockchain networks. Another major part of our design is the CCHDA whose availability is affected by remote servers, such as Infura that connect the DApp to the blockchain networks. This in return affects the availability of cross-chain communication in case the DApp was required for a translation of the interactions. In our architecture, we ensure that the entities that manage the CCHDA have an interest in a successful cross-chain interaction that facilitates achieving protection against adverse attempts.

4) NON-REPUDIATION AND PROVENANCE
Given the considerable off-chain interactions as part of the appXchain architecture, data provenance is significant to facilitate non-repudiation especially due to the decentralized nature of the setup. Within the healthcare use case, all entities, including the patient, doctor, and verifiers, are capable of sharing invalid information. For example, a patient can submit an invalid EMR document token hash to the healthcare causing any verifier response to be incorrect, thereby lowering their scores. To overcome the possibility of such attacks, our architecture uses internal and external reputation systems that allow entities to inspect the reputation of other entities within the same network, in addition to other public entities, such as the verifiers of an external network.

B. COMPARISON WITH EXISTING SOLUTIONS
Table 4 compares the appXchain approach with the current solutions investigated in Section II. The criteria for the comparison is based on our objectives for the optimal cross-chain interoperability standard. All the existing solutions, except those who use hash-locking, rely on a centralized entity for verifying the transactions and ensuring data and assets are transferred fairly. The notary scheme and the Interledger Protocol solutions have at least one blockchain network which blindly trusts another entity. Although all solutions maintain the integrity of the communicated messages across blockchains, notary-based solutions including Cosmos and Polkadot use complex mechanisms for integrity verification, thereby making them inefficient. Moreover, all solutions have limited scalability, except Cosmos and Polkadot, which directly tackle this issue with the concepts of parallel blockchains [34]. Framework-independent solutions would be able to provide cross-chain communication support for networks of different frameworks, such as Ethereum with Hyperledger Fabric. Theoretically, all solutions are independent, except notary-based solutions, because they rely on a specific block header structure. As for being domain-neutral, certain solutions cannot provide such flexibility as they only support token-based assets.

C. OPEN CHALLENGES
Herein, we outline and discuss the limitations of the appXchain in the form of open challenges.

1) INCREASING COST OF DEPLOYMENT
Although our evaluation and cost analysis has returned favorable results, it can benefit from further evaluation to simulate diverse real-world scenarios. For instance, the average gas price and the price of the Ether cryptocurrency are not stable

and continue to increase at a high rate. Additionally, our implementation is focused on a sample use case where not all entities are governed by the reputation systems, which can increase the complexity and cost of the solution.

2) LIMITED UPGRADABILITY

Our solution provides an adaptable and future-proof infrastructure, consisting mainly of the CCHDA and off-chain verifiers. However, the existing blockchain networks and their Ethereum-based smart contracts are not easily upgradable. Possible solutions to this problem have been proposed, such as using links as in Truffle Migrations and using proxies as in ZeppelinOS.

3) CROSS-INDUSTRY INTEROPERABILITY

Our solution provides a generalizable solution for cross-chain interoperability. Our implementation and testing of the solution in the healthcare industry further prove the adaptability of the solution. However, in the case of interoperating across heterogeneous industries, such as insurance-supply chain interoperability, further consideration of application-specific parameters is required including transaction formats, consensus algorithms, and blockchain configurations.

D. ADOPTION

The objectives intended to be achieved by the appXchain define the scope of the solution's adoption into different use case scenarios. Based on our comparisons in subsection VI-B, the appXchain is a solution targeted for situations where the system must be decentralized, where no entity blindly trusts another entity without verification, ensures the integrity of communication, efficient in computations and costs, scalable, does not rely on a specific framework, and not limited to coin or token asset transfer. Such properties of the appXchain make it a potential solution. On the other hand, the adoption of the appXchain can be held back by existing solutions for current blockchain networks. One area where interoperability solutions are established with widespread use is asset transfer. For example, notaries such as Binance and Coinbase enable interoperability between Bitcoin and Ethereum blockchain networks [35].

VII. CONCLUSION

In this paper, we have proposed an application-based cross-chain interoperability solution (appXchain) that is fully decentralized with the potential to be adapted to a wide range of use cases at low implementation and deployment cost. The appXchain architecture utilizes the fusion interface layer inside the Cross-Chain Hub DApp of critical entities in the network to translate cross-chain interactions and delegate requests from one blockchain network to another. Our system adopts the concept of verifier nodes that are internal computation nodes for each network with the capability of communicating with other verifier nodes of external networks to ensure the validity of responses and shared data. To govern

the off-chain interactions, our approach leads to a public reputation system for external verifiers, allowing nodes to query and learn the reputation of other nodes. To confirm the viability of the appXchain architecture, we designed a set of algorithms specific to the healthcare industry and implemented the smart contracts for sharing patient-centered EMR documents across separate blockchain networks. We evaluated our solution to verify its use case functionality in terms of operations and analyze the cost overheads. Furthermore, we analyzed the security implications of the appXchain solution highlighting that it does not introduce additional security threats.

REFERENCES

[1] I. Yaqoob, K. Salah, R. Jayaraman, and Y. Al-Hammadi, "Blockchain for healthcare data management: Opportunities, challenges, and future recommendations," *Neural Comput. Appl.*, Jan. 2021, doi: 10.1007/s00521-020-05519-w.

[2] M. Rauchs, A. Blandin, K. Bear, and S. B. McKeon, "2nd global enterprise blockchain benchmarking study," *SSRN Electron. J.*, Sep. 2019, doi: 10.2139/ssrn.3461765.

[3] Y. Pang, "A new consensus protocol for blockchain interoperability architecture," *IEEE Access*, vol. 8, pp. 153719–153730, 2020, doi: 10.1109/ACCESS.2020.3017549.

[4] T. Hardjono, A. Lipton, and A. Pentland, "Toward an interoperability architecture for blockchain autonomous systems," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1298–1309, Nov. 2020, doi: 10.1109/TEM.2019.2920154.

[5] S. Biswas, K. Sharif, F. Li, Z. Latif, S. S. Kanhere, and S. P. Mohanty, "Interoperability and synchronization management of blockchain-based decentralized e-Health systems," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1363–1376, Nov. 2020, doi: 10.1109/TEM.2020.2989779.

[6] C. Lima, "Developing open and interoperable DLTblockchain standards [standards]," *Computer*, vol. 51, no. 11, pp. 106–111, Nov. 2018, doi: 10.1109/MC.2018.2876184.

[7] M. Zarour, M. T. J. Ansari, M. Alenezi, A. K. Sarkar, M. Faizan, A. Agrawal, R. Kumar, and R. A. Khan, "Evaluating the impact of blockchain models for secure and trustworthy electronic healthcare records," *IEEE Access*, vol. 8, pp. 157959–157973, 2020, doi: 10.1109/ACCESS.2020.3019829.

[8] V. Buterin, "Chain interoperability," R3 Res., White Paper, 2016.

[9] E. Abebe, D. Behl, C. Govindarajan, Y. Hu, D. Karunamoorthy, P. Novotny, V. Pandit, V. Ramakrishna, and C. Vecchiola, "Enabling enterprise blockchain interoperability with trusted data transfer (industry track)," in *Proc. 20th Int. Middleware Conf. Ind. Track*, New York, NY, USA, Dec. 2019, pp. 29–35, doi: 10.1145/3366626.3368129.

[10] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, and Y.-C. Hu, "HyperService: Interoperability and programmability across heterogeneous blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, Nov. 2019, pp. 549–566, doi: 10.1145/3319535.3355503.

[11] Q. Lu, X. Xu, Y. Liu, I. Weber, L. Zhu, and W. Zhang, "uBaaS: A unified blockchain as a service platform," *Future Gener. Comput. Syst.*, vol. 101, pp. 564–575, Dec. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18319873

[12] S. Schulte, M. Sigwart, P. Frauenthaler, and M. Borkowski, "Towards blockchain interoperability," in *Business Process Management: Blockchain and Central and Eastern Europe Forum*, C. Di Ciccio, R. Gabryelczyk, L. García-Bañuelos, T. Hernaus, R. Hull, M. I. Štemberger, A. Kő, and M. Staples, Eds. Cham, Switzerland: Springer, 2019, pp. 3–10.

[13] P. Lafourcade and M. Lombard-Platet, "About blockchain interoperability," *Inf. Process. Lett.*, vol. 161, Sep. 2020, Art. no. 105976. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020019020300636

[14] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," Tech. Rep., 2016.

[15] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2019. [Online]. Available: https://par.nsf.gov/biblio/10168773

[16] D. Robinson, "HTLCs considered harmful," in *Proc. Stanford Blockchain Conf.*, 2019.

[17] A. Hope-Bailie and S. Thomas, "Interledger: Creating a standard for payments," in *Proc. 25th Int. Conf. Companion World Wide Web-(WWW) Companion*, 2016, pp. 281–282, doi: 10.1145/2872518.2889307.

[18] I. A. Qasse, M. A. Talib, and Q. Nasir, "Inter blockchain communication: A survey," in *Proc. ArabWIC 6th Annu. Int. Conf. Res. Track (ArabWIC)*, New York, NY, USA, 2019, pp. 1–6, doi: 10.1145/3333165.3333167.

[19] G. Sagirlar, J. D. Sheehan, and E. Ragnoli, "On the design of co-operating blockchains for IoT," in *Proc. 3rd Int. Conf. Inf. Comput. Technol. (ICICT)*, 2020, pp. 548–552, doi: 10.1109/ICICT50521.2020.00093.

[20] J. Kwon and E. Buchman. (2019). *Cosmos Whitepaper*. [Online]. Available: https://cosmos.network/cosmos-whitepaper.pdf

[21] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," White Paper, 2016. [Online]. Available: https://github.com/polkadot-io/polkadotpaper/raw/master/PolkaDotPaper.pdf

[22] P. Zhang, J. White, D. C. Schmidt, and G. Lenz, "Applying software patterns to address interoperability in blockchain-based healthcare apps," 2017, *arXiv:1706.03700*. [Online]. Available: https://arxiv.org/abs/1706.03700

[23] P. Zhang, M. A. Walker, J. White, D. C. Schmidt, and G. Lenz, "Metrics for assessing blockchain-based healthcare decentralized apps," in *Proc. IEEE 19th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Oct. 2017, pp. 1–4, doi: 10.1109/HealthCom.2017.8210842.

[24] G. Carter, B. Chevellereau, H. Shahriar, and S. Sneha, "OpenPharma blockchain on FHIR: An interoperable solution for read-only health records exchange through blockchain and biometrics," *Blockchain Healthcare Today*, Jun. 2020. [Online]. Available: https://blockchainhealthcaretoday.com/index.php/journal/article/view/120

[25] B. Chevallereau, G. Carter, and S. Sneha, "Voice biometrics and blockchain: Secure interoperable data exchange for healthcare," *Blockchain Healthcare Today*, vol. 2, Dec. 2019. [Online]. Available: https://blockchainhealthcaretoday.com/index.php/journal/article/view/119

[26] C. Chinchilla. (2019). *A Next-Generation Smart Contract and Decentralized Application Platform*. Accessed: Jan. 18, 2021. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper/

[27] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[28] (2018). *Quorum Whitepaper*. Accessed: Jan. 18, 2021. [Online]. Available: https://github.com/jpmorganchase/quorum/blob/master/docs/Quorum%20Whitepaper%20v0.2.pdf

[29] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf. (EuroSys)*, New York, NY, USA, 2018, pp. 1–15, doi: 10.1145/3190508.3190538.

[30] M. M. Madine, A. A. Battah, I. Yaqoob, K. Salah, R. Jayaraman, Y. Al-Hammadi, S. Pesic, and S. Ellahham, "Blockchain for giving patients control over their medical records," *IEEE Access*, vol. 8, pp. 193102–193115, 2020, doi: 10.1109/ACCESS.2020.3032553.

[31] J. Benet. (2014). *IPFS—Content Addressed, Versioned, P2P File System*. Accessed: Jan. 18, 2021. [Online]. Available: https://github.com/ipfs/ipfs/blob/master/papers/ipfs-cap2pfs/ipfs-p2p-file-system.pdf

[32] (2019). *HL7—Fast Healthcare Interoperability Resources Documentation*. Accessed: Jan. 18, 2021. [Online]. Available: https://hl7.org/implement/standards/fhir/

[33] M. M. Madine, K. Salah, R. Jayaraman, I. Yaqoob, Y. Al-Hammadi, S. Ellahham, and P. Calyam, "Fully decentralized multi-party consent management for secure sharing of patient health records," *IEEE Access*, vol. 8, pp. 225777–225791, 2020, doi: 10.1109/ACCESS.2020.3045048.

[34] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," Tech. Rep., 2021.

[35] N. Kannengießer, M. Pfister, M. Greulich, S. Lins, and A. Sunyaev, "Bridges between islands: Cross-chain technology for distributed ledger technology," in *Proc. Hawaii Int. Conf. Syst. Sci.*, Maui, HI, USA, Jan. 2020.

MOHAMMAD MADINE received the B.Sc. degree in computer engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2019, where he is currently pursuing the degree. He is also a Graduate Researcher and a Teaching Assistant. His research interests include blockchain solutions in healthcare, personal health records, and edge computing.
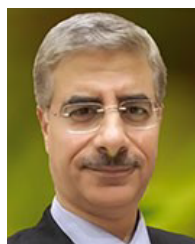
KHALED SALAH (Senior Member, IEEE) received the B.S. degree in computer engineering from Iowa State University, USA, in 1990, with a focus on computer science, and the M.S. degree in computer systems engineering and the Ph.D. degree in computer science from the Illinois Institute of Technology, USA, in 1994 and 2000, respectively. He is currently a Full Professor with the Department of Electrical and Computer Engineering, Khalifa University, United Arab Emirates. He has over 220 publications and three U.S. patents, has been giving a number of international keynote speeches, invited talks, tutorials, and research seminars on the subjects of blockchain, the IoT, fog and cloud computing, and cybersecurity. He is a member of the IEEE Blockchain Education Commitee. He served as the Chair of the Track Chair for IEEE Globecom 2018 on Cloud Computing. He is an Associate Editor of IEEE Blockchain Tech Briefs. He is also leading a number of projects on how to leverage blockchain for healthcare, 5G networks, combating deepfake videos, supply chain management, and AI.

RAJA JAYARAMAN received the bachelor's and master's degrees in mathematics in India, the M.Sc. degree in industrial engineering from New Mexico State University, and the Ph.D. degree in industrial engineering from Texas Tech University. He is currently an Associate Professor with the Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His expertise is in multi-criteria optimization techniques applied to diverse applications including supply chain and logistics, healthcare, energy, environment, and sustainability. His Postdoctoral Research was centered on technology adoption and implementation of innovative practices in the healthcare supply chains and service delivery. He has led several successful research projects and pilot implementations in the area of supply chain data standards adoption in the U.S. healthcare system. His research has appeared in top-rated journals including: *Annals of Operations Research*, *IISE Transactions*, *Energy Policy*, *Applied Energy*, *Knowledge Based Systems*, IEEE ACCESS, *Journal of Theoretical Biology*, and *Engineering Management Journal*. His research interests include blockchain technology, systems engineering and process optimization techniques to characterize, model and analyze complex systems with applications to supply chains, maintenance operations planning, and healthcare delivery.

YOUSOF AL-HAMMADI received the bachelor's degree in computer engineering from the Khalifa University of Science and Technology (previously known as Etisalat College of Engineering), United Arab Emirates, in 2000, the M.Sc. degree in telecommunications engineering from the University of Melbourne, Australia, in 2003, and the Ph.D. degree in computer science and information technology from the University of Nottingham, U.K., in 2009. He is currently an Acting Dean of Graduate Studies and an Assistant Professor with the Electrical and Computer Engineering Department, Khalifa University of Science and Technology. His research interests include the area of information security which include intrusion detection, botnet/bots detection, viruses/worms detection, machine learning and artificial intelligence, RFID security, and mobile security.

JUNAID ARSHAD received the Ph.D. degree from the University of Leeds, U.K. He investigated the challenge of effective intrusion severity analysis for clouds with the University of Leeds. He is currently an Associate Professor with the School of Computing and Digital Technology, Birmingham City University, U.K. His research interests include distributed computing and high-performance computing, such as grid and cloud computing, service-oriented computing, and the Internet of Things emphasizing security challenges including intrusion detection and response, trust establishment and management, and security event classification.

IBRAR YAQOOB (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Malaya, Malaysia, in 2017. He is currently working with the Department of Electrical Engineering and Computer Science, Khalifa University, United Arab Emirates. Previously, he worked as a Research Professor with the Department of Computer Science and Engineering, Kyung Hee University, South Korea, where he completed his Postdoctoral Fellowship under the prestigious grant of Brain Korea 21st Century Plus. He worked as a Researcher and a Developer with the Centre for Mobile Cloud Computing Research (C4MCCR), University of Malaya. His numerous research articles are very famous and among the most downloaded in top journals. He has been listed among top researchers by Thomson Reuters (Web of Science) based on the number of citations earned in six categories of computer science. He is also serving/served as a guest/an associate editor in various journals. He has been involved in a number of conferences and workshops in various capacities. His research interests include big data, blockchain, edge computing, mobile cloud computing, the Internet of Things, and computer networks.

● ● ●