Frontiers in Health Informatics ISSN-Online:2676-7104
www.healthinformaticsjournal.com
2024; Vol 13: Issue 3
Open Access

Facilitating the Secured Transfer of EHRs through the Implementation of Hybrid Encryption Methods over a Solid Blockchain-Based Framework

Destin N Joy[1], Chandra Shekhar Yadav[2]
[1]School Of Computer Science Singhania University Pacheri Bari, Jhunjhunu (Raj.), India.
Destin.Joy@hotmail.com
[2]Professor and Dean, School of Computer Applications, Affiliation - Noida Institute of Engineering and Technology Greater Noida, India. csyadavrp@gmail.com; drcsyadav@niet.co.in

Cite this paper as: Destin N Joy, Chandra Shekhar Yadav (2024) Facilitating the Secured Transfer of EHRs through the Implementation of Hybrid Encryption Methods over a Solid Blockchain-Based Framework. *Frontiers in Health Informatics, 13* (3),8796-8811

Abstract:

There are serious issues with security, confidentiality, and compatibility with the "Electronic-Health Records (EHRs)" information that is kept and exchanged in distributed computing environments. A significantly more protected, confidential, and compatible system with emerging and distributed technology, brings us to the rise of "Block-Chain (BC)" innovation as an alternative. By allowing the Patient "Data-Owner (DO)" to exchange their secure information through the network, data encryption in a BC-based distributed storage framework assures security. The existing research provides several systems, including "Digital-Signature," "Homomorphic-Encryption," and multiple encryption algorithms, allowing the safe transmission of EHRs across a BC context. When it comes to the secure transmission of data, the "Symmetric-Encryption (SE)" and "Asymmetric-Encryption (AE)" methods are crucial. Intending to keep EHR data secure and user anonymity unaffected, these methods are vital in the healthcare BC network. A "Privacy-Preserving (PP)" based "Hybrid-Encryption (HE)" which incorporates both SE and AE has been proposed in the present research for use in BC-based healthcare networks for encryption of EHR data and restricting anonymity reading and retrieving at the same time. The suggested BC-HE architecture employs two methods for encrypting and decrypting EHR information and keys: SE utilizing the AES method and AE utilizing the RSA method. This allows for the smooth PP operation of the system in a public domain. All of the nodes within the network will get the newly created keys when the proposed model generates the Symmetric-Key and Public-Key pairs. In addition to protecting the confidentiality of BC's DOs, this BC-HE method will make it easier for authorized clients to share data between departments and organizations. After comparing the proposed BC-HE framework with the existing BC-LS framework utilizing the measurements of "Security Ratio," "Encryption Time," and "Decryption Time," the results demonstrate that the proposed BC-HE offers a higher level of security.

Keywords: *EHR, Block-Chain, Security, AES, RSA*

1. INTRODUCTION

Numerous countries' recent healthcare reform programs as well as the fast expansion of the global web have led to a huge growth in the quantity of EHRs. EHR enables the continuous gathering of data from several channels and provides patients with the knowledge they need to take an active role in their healthcare, with a focus on their well-being throughout their lives. The domains involving the health of patients, research in clinical sciences, medical care, and developing hospitals could reap enormous

advantages from EHRs. By providing several regulated formats and supporting resources, EHRs allow practitioners to devote more time to patient care and less on making medical records [1].

When patients visit multiple healthcare centers for various medical exams, an immeasurable amount of EHRs is created. Therefore, an issue known as an information island arises for EHRs. Whenever patients and physicians want access to medical records, it requires a significant investment of time and energy to move EHRs across various healthcare organization's databases. However, these issues may be resolved by the exchange of EHRs across healthcare facilities. It has the potential to increase the accuracy level of patient illness diagnoses while offering specialists additional history information to assist in their ability to decide [2].

However, there are a lot of issues with EHR transmission and communication. Firstly, there is no universally accepted template for EHRs, and the models utilized among various healthcare organizations are very distinct. Secondly, there is a long-term access period since patients have to confirm their identification and check their privileges of access before they are allowed to use EHRs. Thirdly, issues including inadequate storage protection, leakage of information, and records manipulation originate from DO's concerns about their privacy in the massive EHR network. Thus, finding solutions to the aforementioned challenges is a major focus of the medical community [3].

The last few decades have seen enormous development in EHRs, owing to the proliferation of cloud-based storage. An increasing number of individuals are moving their EHR storage to "Cloud Servers (CSs)" under the "Cloud Computing (CC)" paradigm because of the low cost of computing and the enormous capacity available there. Leveraging CS technologies for storage, an area-wide medicinal data exchange system is constructed to enhance EHR capabilities sharing. This framework thoroughly combines the systems of several healthcare organizations [4].

Still, a recurring issue with these systems is their reliance on "Cloud-Service Providers (CSPs)". Data leaking is an imminent threat if CSPs are the targets of malicious attacks. For instance, attackers obtain critical health information using technological advances and perform illicit activities to earn big profits since most networking equipment is immediately approved for accessing the public networks beneath the motive of profits. Furthermore, the fact that CSs would conspire with physicians to tweak the outsourced EHRs is not taken into account by any of the current approaches. This can be challenging to identify such conduct once it happens [5].

The issues of EHR manipulation, counterfeiting, and leaking may be resolved by using BC technology because it offers decentralized management, permanence, and tracking. The security and confidentiality of personal information may also be enhanced, and data exchange can be encouraged in the healthcare industry's manufacturing processes. By eliminating the requirement for an outside party to conduct accreditation inspections and data examination, such BC systems can successfully tackle the issues caused by the CC's storing model's emphasis on health data [6].

Data exchange is more efficient, and EHR data becomes more secure and private. To obtain an agreement within the BC system, every single transaction is required to be shown to the nodes, this could lead the transaction's details to be revealed. Intending to encourage the growth of BC advances in technology, the question concerning how to secure transaction data has emerged as a critical one [7]. Cryptographic methods could potentially utilized for safeguarding data stored in BC, according to specialists, which would eliminate the issue of privacy leaking of data related to transactions. The

application of cryptographic technology associated with BC allows for the implementation of real-time monitoring of the EHR authentication procedure, and EHRs may be seen as a valuable privacy resource as a whole [8].

Problem Statement: Conventional methods of controlling who may access what when exchanging EHRs presume that CSs are entirely trustworthy by DOs and provide CSs complete authority over all data accessibility and authorization. Considering that the CS is sincere yet inquisitive, this presumption seems no longer valid in CC. Although the CS promises to behave honestly when processing data inquiries, they could gather sensitive information despite the knowledge or permission of DOs. This might cause major problems with the leakage of data as well as the security of the network [9]. Distributed, traceable, and programmable, the BC seems a crucial part of EHR and may prevent data corruption and forgery. In some situations, the BC may be utilized to provide secure and reliable EHR administration while simultaneously obscuring the data. BC's storing and data-sharing performance is low, and EHRs often include massive, cross-media clinical information like CT and various other medical imaging data [10]. To take advantage of their mutual benefits, CC and BC systems require being integrated promptly through some security approaches.

Paper Contribution: In existing works, few proposals exploit key policy methods to protect the privacy of users and many proposals utilize single encryption mechanisms and access control policies for securing the EHR data. Nevertheless, none of the proposals combine the necessity of SE and AE for end-to-end encryption to ensure authenticity and acknowledgment. Hence, there is a need to adapt both the SE and AE techniques in the e-Health BC system to enhance security and privacy. In this research work, a PP technique for the patient-centric e-health system is presented to enhance the privacy of patients and their EHR data. In this proposed system, PP is ensured in the CC-based BC network through confidentiality. This technique aims to facilitate data sharing across organizational boundaries with authorized users and provide security to stored EHR data besides improving the privacy of BC users. To achieve this, the proposed system introduces a HE (SE & AE) technique for content encryption in the BC network. Shielded data is communicated between the nodes/parties in the BC network. For this process, it generates the Symmetric-Key and Public-Key for encrypting and decrypting the content. It then performs a key exchange process based on the user's request. By utilizing the proposed BC-HE technique it enhances the privacy of e-Health users.

Paper Organization: Section 2 of the research article provides a literature review on EHR security over CC and BC networks, focussing on articles that discuss the transmission of sensitive data, Section 3 explores further the methodologies of the suggested BC-HE framework, outlining the current BC-LS frameworks for EHR confidentiality, Section 4 presents a comparison of the findings from implementing both of these frameworks, and Section 5 concludes the research and suggests areas that require additional research.

2. RELATED WORKS

A new certificate-less "Provable Data-Possession (PDP)" system for the safe management of EHRs on CSs was suggested by the researchers in [11]. Sharing several backups over different CSs improves data recovery, which is a major concern when it comes to EHR preservation and security. Approved physicians will be enabled to access EHRs from earlier periods owing to a novel data format called the Map-Version Marker-Table, which allows for block-level tracking and dynamic functions. This study proves that the system is secure for CC-based EHR and bases its assertion on the fact that the Diffie-

Hellman computing issue is intractable.

Concerns regarding security associated with healthcare IoT, especially for EHRs built with CC, are discussed by the researchers in [12]. As a way to improve the system's capacity for expansion and utility, they suggest implementing restricted access and flexible grouping of users. An effective revoking strategy is included in the system, guaranteeing both backward and forward confidentiality and reversible storing to avoid illegal accessibility to data. This technique tackles the urgent demand for strong PP for medical data administration while being safe from contemporary risks.

An encrypted method for cloud-based EHR administration was suggested by the researchers in [13]. It uses a combination of RSA's "Improved Key-Generation Scheme" and the Blowfish technique for encryption of data. Also, for safe key exchanges, it incorporates steganography-oriented accessibility management. Enhanced EHR confidentiality and fast retrieval of data on the BC infrastructure are two benefits of the system.

A PP-based Federated-Learning Framework featuring Homomorphic-Encryption was developed by the researchers in [14] to address the security and privacy concerns related to medical information to guarantee data confidentiality. This methodology uses Accessibility-Control techniques to confirm the identity and reliability of users, together with encryption from the DO end to secure the privacy of distributed models used for training. To manage users quickly, reduce transmission costs, and solve client disengagement while training, an acknowledgment system was developed on the server end. When it comes to medical applications built on BC, this two-pronged approach improves Federated-Learning and enhances safeguarding personal information.

To enhance healthcare system protection, the researchers of [15] created a technique called "BtRaI" that integrates technology with trustworthy reputation evaluation. Improving medical facilities, including time-sensitive observation and remote illness detection, is the primary objective of this concept. It does this by offering a thorough reputation evaluation system that promotes involvement with the agreement-building process, which in turn discourages detrimental conduct. A token-oriented incentive and penalty structure, a PBFT technique for better effectiveness in BC collaboration, and components for a multifaceted reputation evaluation are notable characteristics of this framework.

3. METHODOLOGIES

The permissioned BC is private and only authorized parties can enter into the network. Since BC is a transparent network, the permissioned BC ensures the availability and transparency of stored data; all participants of the permissioned BC network can view the entire BC content. This leads to a privacy issue among the participants. To solve this problem, a new BC-HE mechanism is proposed to maintain the confidentiality of stored data and provide privacy to DOs. The suggested BC-HE methodology's architecture is illustrated in Figure 1.

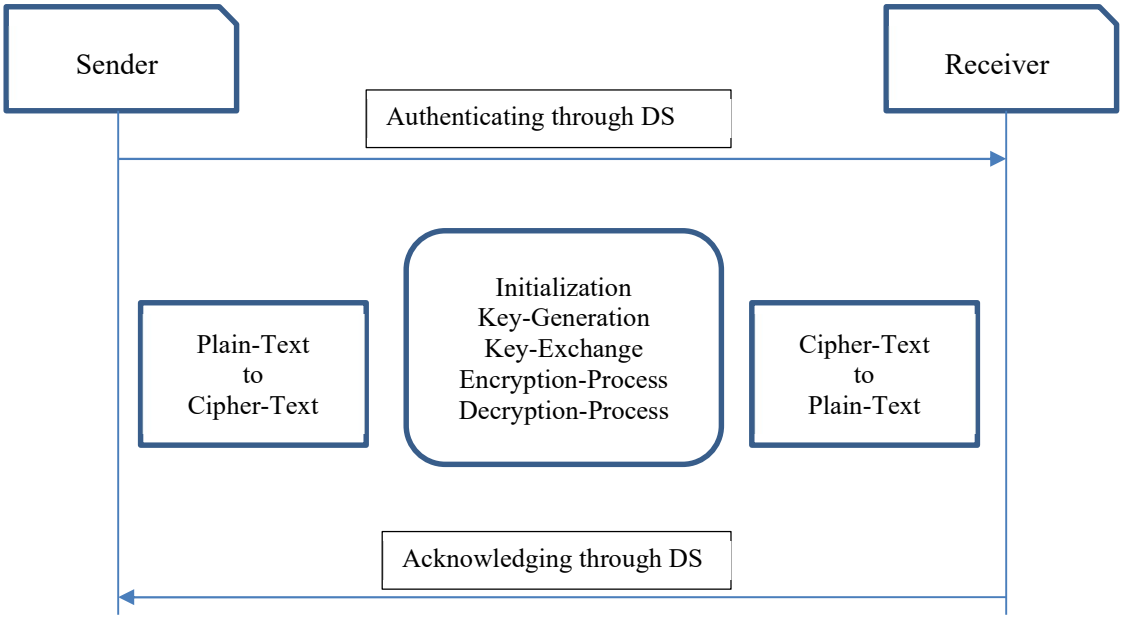

Figure 1: Proposed BC-HE Methodology

The working principle of the proposed BC-HE architecture is as follows:

- It utilizes both Symmetric and Asymmetric-Keys for Content Encryption/Decryption. It creates the Symmetric-key and Public-key pairs and then shares the generated keys to all the participating nodes in the BC network. Once the keys reach all the participating BC nodes, then it aims to encrypt the content with the Symmetric-Key.
- The system permits the BC users to initiate a content request to any participant in the network. Thereby, the client generates a request to the DO for retrieving the content along with their Public-key. Then the proposed BC-HE framework stimulates the DO to receive the request from the client to retrieve and view a particular content with a Public-Key.
- Then it performs an encryption process at the sender/DO end and the Symmetric-Key gets encrypted with the Public-Key of the receiver at the sender/DO side. With this end, the proposed system sends the Encrypted-Key to the receiver for decryption. To ensure data authenticity, the proposed BC-HE framework also generates and sends a "Digital-Signature (DS)" along with the Encrypted-key to the receiver. The Digitally-Signed Encrypted-key has been transmitted to the requested client over the network for the decryption process.
- Once the receiver receives the Encrypted-key, it decrypts the key with the receiver's Private-Key, and the decryption process is carried out by the system at the receiver's side. Once the key decryption process is completed then it permits the decryption of content with the help of the key on the client/receiver side. After decryption, the system allows the receiver to view the original content.
- For acknowledging the key, it immediately generates a DS and sends the same to the sender. The receiver ensures the receipt of such a DS by sending the acknowledgment through an encrypted DS. The sender can now use the Private-Key for decrypting the DS.

3.1 BC-LS (Existing Framework)

To solve the problem of insecure EHR data transmission, a novel framework "BC-Lightweight Signcryption (BC-LS)" has been developed earlier which integrates LS techniques with innovative BC

solutions. According to the envisioned BC-LS framework, it contains 2 stages throughout the whole process. During the first stage, "Neighborhood Sensor-Nodes (NSNs)" are deployed to establish "Wireless Body-Area Networks (WBANs)" where patients may interact. Proximity NSNs could establish a "Cluster-Head (CH)" that can exchange EHR information with the WBAN's "Gateway SNs (GSNs)" network. The GSN integrates encrypted data and signing with an LS method for authorized participants during the next stage. The framework uses the "Inter-Planetary File-System (IPFS)" for secured access through keys and data exchange through public networks. The overall objective of this research initiative is to find ways to make EHR access possible from any location while also protecting patients' personal information. Guaranteeing the continued availability of encrypted EHR is the responsibility of the BC. A BC functions as a decentralized digital ledger that records each transaction. The cryptographic keys interconnect blocks of the BC transactions using a hashing method. All of those Secret-Keys have been stored in shared ledgers which are connected by SN. Each of the SNs updates and synchronizes the BC's redundant data.

3.2 BC-HE (Proposed Framework)

The proposed BC-HE system has the following phases i) Initialization, ii) Key-Generation, iii) Key-Exchange, iv) Encryption-Process, and v) Decryption-Process. The following sections will describe each phase of the proposed BC-HE system:

3.2.1 Initialization

The system elements and parameters of the BC-HE are presented in Table 1.

Table 1: Elements and Parameters

S.No	Elements	Parameters	Description
1.	Participants	$P = \{ P_1, P_2, P_3, \ldots P_n \}$	Sender and Receiver
2.	File Path	FP = File Location	Local File Storage, Inter Planetary File System (IPFS)
3.	Block	$B = \{ B_1, B_2, B_3, \ldots B_n \}$	Blockchain Blocks
4.	File Content	$D = \{ D_1, D_2, D_3, \ldots D_n \}$	Original Text / PlainText
5.	Cipher Text	C	Encrypted Text
6.	Symmetric Key	S_Key	Symmetric Key for Content Encryption
7.	Asymmetric Key	Private_key, Public_Key	Public key Cryptography
8.	Asymmetric Key	SSK, SPK	Sender Initiated
9.	Asymmetric Key	RSK, RPK	Receiver Initiated
10.	Digital Signature	Digi_Sign	Authentication and Acknowledgement

The initialization phase of the proposed BC-HE framework identifies the elements and constructs the necessary parameters for the elements for the ultimate execution of the framework. The elements of the proposed BC-HE framework for content encryption/decryption are BC Participants, Input-File

location, File-Content, Cipher-Text, Symmetric-Keys and Asymmetric-keys (both Private-Keys and Public-keys) for sender and receivers. Once the elements are identified, the system creates the appropriate parameters.

3.2.2 Key-Generation

In this section, the functional components of the Key-Generation method of the proposed BC-HE system are presented. The BC-HE generates the necessary keys to each node after completion of the key initialization process and identification of the BC users. To enhance security and privacy, it performs Double-Encryption and Double-Decryption processes. To achieve this, the BC-HE framework generates three keys for each participant of the network. The three keys are one Symmetric-Key and a pair of Asymmetric-Keys, containing a Private-Key and a Public-Key. The same is true for all the participants of the network. The generation of Symmetric-Key and Asymmetric-Key and the procedure involved in this phase are described in the following subsections.

3.2.2.1 Symmetric-Key Generation

This process aims to facilitate the encryption of the EHR file content present in the blocks of the BC network. The S_Key is the "Private-Key", both parties can utilize the same key for Encryption and Decryption processes. The proposed BC-HE system adapts the "Advanced-Encryption Standard (AES)" algorithm for generating the S_Key. The generated S_key is 256-bits long. Out of the 256-bits, the first 128-bits are used as a Signing-Key; the remaining 128-bits are used as an Encryption-Key. The 128-bits Signing-Key represents the identification of the node and its content, each having 64-bits. The remaining 128-bits aim to encrypt the identified File-Content. Once the proposed BC-HE framework generates the S_Key, it then encrypts the file content whenever a block is created in the BC network. The proposed framework stores the appropriate S_Key in each block after the encryption process for further usage in decryption. After the generation of S_Key, the HE technique introduces Asymmetric-Key pairs to encrypt the S_Key.

3.2.2.2 Asymmetric-Key Generation

This process aims to facilitate the encryption of the generated S_Key. The purpose of encrypting the S_Key is to avoid the view of the content under analysis by the participants present in different nodes of the BC network. Even though the encrypted content is available to all the participants, the system ensures security by not providing the S_Key. By introducing another layer of security, the Asymmetric-Key can encrypt the S_Key. Thus, the content view is prohibited for conventional users of the BC nodes, as they may not be the actual recipients of the content. If any one of the BC nodes is interested in viewing the content, then the node should apply a request to the DO along with their Public-Key. For S_Key Encryption, the proposed framework generates Asymmetric-Key to all BC nodes. Asymmetric-Key is a Public-Key Cryptography; thus BC-HE generates a set of key pairs namely (Private-Key, Public-Key). The key pair shall then be utilized for encryption and decryption processes. The first element of the key pair namely Private-Key referred to as Pr_Key. It facilitates the encryption of S_Key. The other element of this key pair namely Public-key referred to as Pub_key, facilitates the decryption of the corresponding encrypted S_Key.

The proposed BC-HE system adapts the "Rivest-Shamir Adleman (RSA)" algorithm for generating Asymmetric-Keys. By utilizing the RSA algorithm, the proposed system generates

Asymmetric-Key pairs namely Pr_Key and Pub_Key. In the Asymmetric-Key generation, the proposed system considers the two largest "Prime-Numbers (K, Q)" and calculates "N = K*Q". It then constructs the Public-Key element of the "Asymmetric-Key". For the construction of the Public-Key, the system finds the "Public-Exponent (e)" by utilizing "Greatest-Common Division (GCD ((e, φ (N)) == 1))" and generates the "Pub_Key (e, N)". Once the system generates the Public-Key then it constructs the Private-Key element. For constructing the Private-Key, the proposed system finds the Private-Exponent or Secret-Exponent (d) for generating the Private-Key. The proposed system utilizes "d = (2 + φ (N)) / e" for finding the Private-Exponent (d) and generates the Pr_Key (d, N). In the Private-Key and Public-Key constructions, 'e' is the Public-Exponent", 'd' is the Private-Exponent, and 'N' is the link between the Pr_Key and Pub_Key. Thus, the generated Pr_Key and Pub_Key are Asymmetric-Keys. The proposed system utilizes Asymmetric-Key pairs for encrypting the generated S_Key.

3.2.3 Key Exchange

The process of exchanging the keys between the sender and receiver is presented in this module. It allows two or more participants to share their keys after making a request. This module permits the BC nodes to exchange their keys over the network after the creation of Symmetric-Key and Asymmetric-Key pairs. In this, the sender is permitted to perform a Double-Encryption process and the receiver is authorized to perform a Double-Decryption process. To perform these encryption and decryption processes, both the sender and the receiver are required to exchange their Public-Keys. The proposed BC-HE initiates the key exchange from the receiver's side. Once the receiver keys are exchanged properly, then the response key exchange takes place on the sender side. The key exchange process of the proposed system facilitates the exchange of the keys in two stages, as described below:

Stage 1: The receiver exchanges the Public-Key with Sender

In this stage, the receiver establishes a request for the content along with the Receiver's Public-Key (RPK) to the sender.

Stage 2: Sender exchanging the Public-Key with Encrypted S_Key to Receiver

In this stage, the sender sends an encrypted S_Key to the receiver for decryption purposes.

3.2.4 Encryption

The Encryption process begins at the sender side. To maintain the privacy of the BC users, the proposed PP system performs a Double-Encryption process. The first stage of the encryption process takes place after the completion of the key generation process. The second stage of the encryption process takes place once the key exchange process is completed. Once the complete key set (S_Key, Pr_Key, and Pub_Key) is generated at each participant, the nodes are waiting for file content to perform the first stage of the encryption process. In this proposed BC-HE system, whenever a participant adds data or files on the block in the BC network, the contents are immediately encrypted with the help of S_Key as described in subsection 3.2.2.1. This Content-Encryption is performed with the S_Key, before adding data into the block in the BC network. The first stage of the encryption process performs on all "File-Content (D_i)" present in each "Block (B_i)" with the S_Key_i; the protocol then generates a "Cipher-Text (Ci)" for all of the content present in "Block (Bi)" over the BC network. The proposed BC-HE technique executing the Content-Encryption process is described as "C1 = ES (D, S_Key)". All participants perform the first stage

of Content-Encryption with their own S_Key_i. Once the Content-Encryption is completed with S_key then the protocol completes the first phase of encryption.

After completion of the content-encryption, the proposed framework shall wait for the second stage of encryption, encrypting the S_Key. To encrypt the S_Key, the sender needs the receiver's Public-Key from the receiver. If any one of the clients/participants in the network wishes to view the encrypted content, then the client has to generate a request along with his Public-Key. After receiving the content request from the client along with the receiver's Public-Key, the proposed framework performs the second stage of encryption, encrypting the S_Key with the Receiver's Public-Key, "C2 = EA (S_Key, RPK)". The proposed framework sends the "Encrypted-Key (C2)" to the receiver for decryption. In this work, the process of Content-Encryption and S_Key encryption is turned into Double-Encryption. The Double-Encryption process of the proposed system is illustrated in Figure 2.

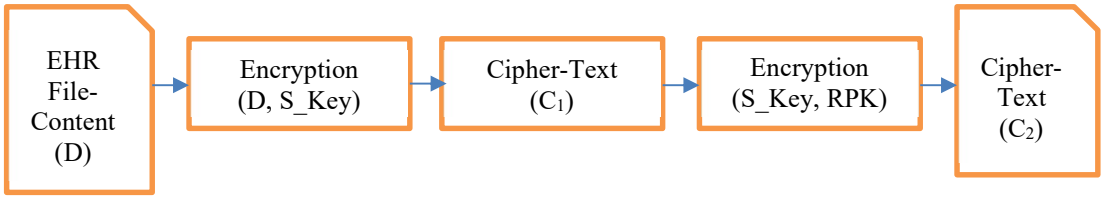

Figure 2: Proposed Double-Encryption Process at Sender Side

In the first phase of encryption, the sender encrypts the "Content (Di)" with his own S_Key, "C1 = ES (D, S_Key)" where 'D' is the sender's content and performs Symmetric-Encryption with S_Key. After the completion of the first phase of encryption, the sender is ready to perform the second phase of encryption. The sender encrypts his own S_Key with the receiver's Public-key (RPK) and obtains "C2 = EA (S_Key, RPK)".

3.2.5 Decryption

The decryption process starts at the receiver side. To view the content, the proposed system performs a Double-Decryption process at the receiver side. It initiates after receiving the decryption key from the DO of the content. Once the key exchange is properly taken place between the sender and the receiver, it decrypts the Encrypted-Key with the receiver's Private-Key. In the first stage of the decryption process, the receiver decrypts the "Cipher-Text (C_2)" with his own "Secret-Key (RSK), $S\_Key=DecryptA(C_2, RSK)$". After the first stage of the decryption process, it generates the original S_Key at the receiver side. After the successful decryption process, it utilizes the S_Key for decrypting the "Encrypted-Content (C_1)" and restores the "Original-Content (D) = DecryptS (C_1, S_Key)". After the Double-Decryption processes, the receiver can view the "Original-Content (D)". The steps involved in the two-stage decryption process of the proposed BC-HE system are presented in Figure 3.

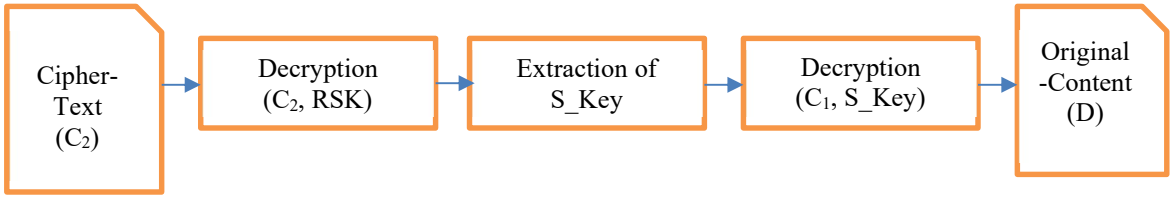

Figure 3: Proposed Double-Decryption Process on the Receiver Side

In the first phase of decryption, the receiver decrypts the "Cipher-Text (C2)" with Receiver's Secret-Key, and decrypts, "DecryptA (C_2, RSK) = S_Key". After the completion of the first phase of decryption, the receiver gets the sender's original S_Key for Content-Decryption. In the second phase of decryption, the receiver decrypts the "Cipher-Text (C_1)" with the help of the sender's S_Key, which is already decrypted in the first phase of decryption, "DecryptS (C_1, S_Key) = D" and obtains the "Content (D)". The receiver is now permitted to view the original content after the completion of the second phase of the decryption process.

3.2.6 Generation of Digital-Signature (DS) for Authentication and Acknowledgement

The strength of the proposed PP-based BC-HE technique is its ability to combine the Symmetric-Key and Asymmetric-Keys for Encryption/Decryption processes. This Dual-Key system performs Double-Encryption and Double-Decryption processes. To ensure authenticity and acknowledgment, the proposed BC-HE system introduces a DS for both the sender and receiver. The capacity of generating a DS permits strong authentication in terms of a message for both the sender and the receiver. With this feature, it is possible to confirm the origin and recipient of the message. The step-by-step process of generating the DS that ensures authentication is presented in Figure 4. In this system, the sender initiates a DS to ensure authenticity. The receiver sends the DS to ensure the acknowledgment. With this end in view, the proposed system authorizes the original DO of the data and acknowledges the original recipient of the data. To execute the scenario, it allows the sender to attach a DS, before sending the encrypted S_Key to the receiver. This process of attaching the DS takes place after the completion of the S_Key encryption process. The process of attaching a DS is illustrated in Figure 4.

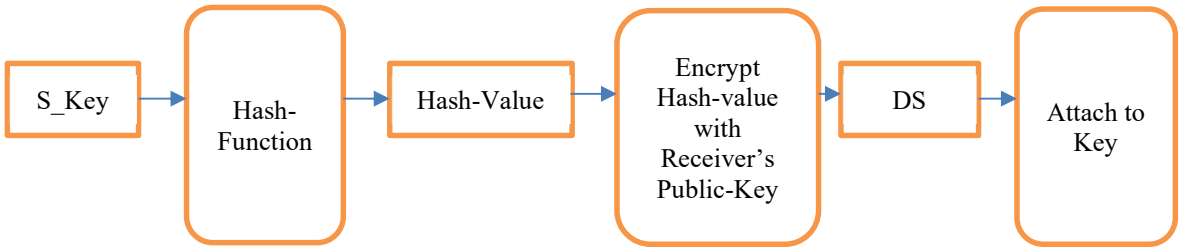

Figure 4: Generation of DS for Authentication

Making a hash-value corresponding to S_Key is the primary step in making a DS. This suggested BC-HE framework employs the widely-used Hash-Function (SHA-256) method to generate hash-values. The result of using this method is a 256-bit integer describing the message. The hash-value is then encrypted using the Public-Key of the recipient. The Encrypted-Hash is the DS and the same is attached along with the encrypted S_Key.

The steps involved in signing and verifying the DS are carried out on the receiver side. After a successful key exchange, the recipient must validate its source by comparing the DS with their public key. After the decryption of the DS, the receiver matches the signature with the hashed S_Key; if both the hashes are the same, the receiver confirms the origin of the message. Once the origin of the message is verified, the receiver then sends an acknowledgment to the sender by sending a hashed S_Key. The process of generating acknowledgment to the sender is presented in Figure 5.

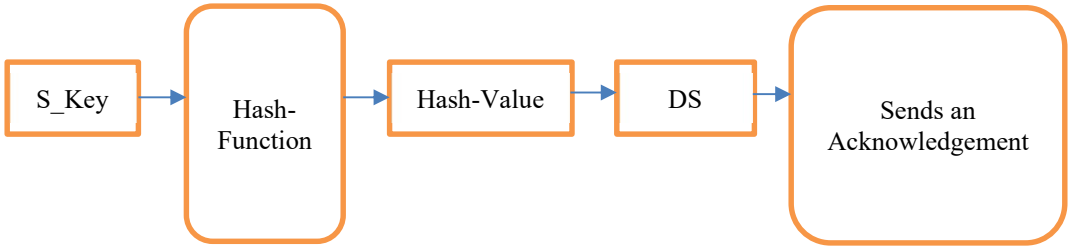

Figure 5: Generation of DS for Acknowledgment

In the proposed BC-HE framework, the receiver generates the DS for acknowledgment. The process of attaching a DS for acknowledgment at the receiver side is illustrated in Figure 5. The first step of the creation of a DS is the creation of a hash of the S_Key. The SHA-256 is utilized for generating the hash-value. After obtaining the hash-value, it is encrypted with the sender's Public-Key. The Encrypted-Hash is the DS and the same is automatically sent back to the sender for acknowledgment.

4. RESULTS AND DISCUSSIONS

The construction of the proposed BC-HE methodology operation as follows as Creation of Blocks, Construction of BC, Content-Encryption, EHR storage, and storing of the Symmetric-Key and Asymmetric-Keys with the respective blocks for key exchange in the encryption/decryption process. To perform these tasks, the proposed BC-HE framework executes the following modules Key-Initialization, Key-Generation, Key-Exchange, Encryption, and Decryption processes. These research experiments were carried out with a huge number of transactions. However, only a few transaction results are presented here due to space constraints. The Java code is developed to generate the above modules of the proposed BC-HE system. The metrics for performance, including "Security Ratio," "Encryption Time," and "Decryption Time" are used to assess both the proposed BC-HE and the current BC-LS framework.

(i) SR "Security Ratio":

An SR is a measure to quantify the fraction of the aggregate EHRs that were transmitted safely relative to the total quantity of EHRs. Based on statistical modeling using Equation (1), the SR has been depicted in the form of "Percentage (%)":

$$SR = \frac{Number\ of\ Patient's\ EHR\ Securly\ \ transfered\ to\ Destination}{Total\ number\ of\ EHR} \qquad \text{Eq}\rightarrow 1$$

Comparisons between the suggested BC-HE framework and the existing BC-LS framework across different numbers of EHRs at the same time are shown in Table 2 about the SR metric. Maintaining a higher level of SR while the EHR extends is indicative of an efficient framework. There are approximately an aggregate of "50" to "250" EHRs. The "X-axis" in Figure 6 shows the "%" of SR obtained, while the "Y-axis" shows the total quantity of EHRs analyzed. The SR has been discovered to range between "89% and 98%" according to Table 2 and Figure 6. The existing BC-LS framework achieves 97% SR with at least "50 EHR," whereas the suggested BC-HE framework achieves 98%. Results show that the existing BC-LS framework achieves 89% with an upper limit of "250 EHR," whereas the proposed BC-HE framework achieves 91%. Therefore, compared to the current BC-LS framework, the suggested BC-HE framework yields better SR for the maximum quantity of EHR across patients.

Table 2: SR Comparison

Total Patient's EHR	BC-LS	BC-HE
50	97	98
100	95	97
150	93	95
200	91	93
250	89	91

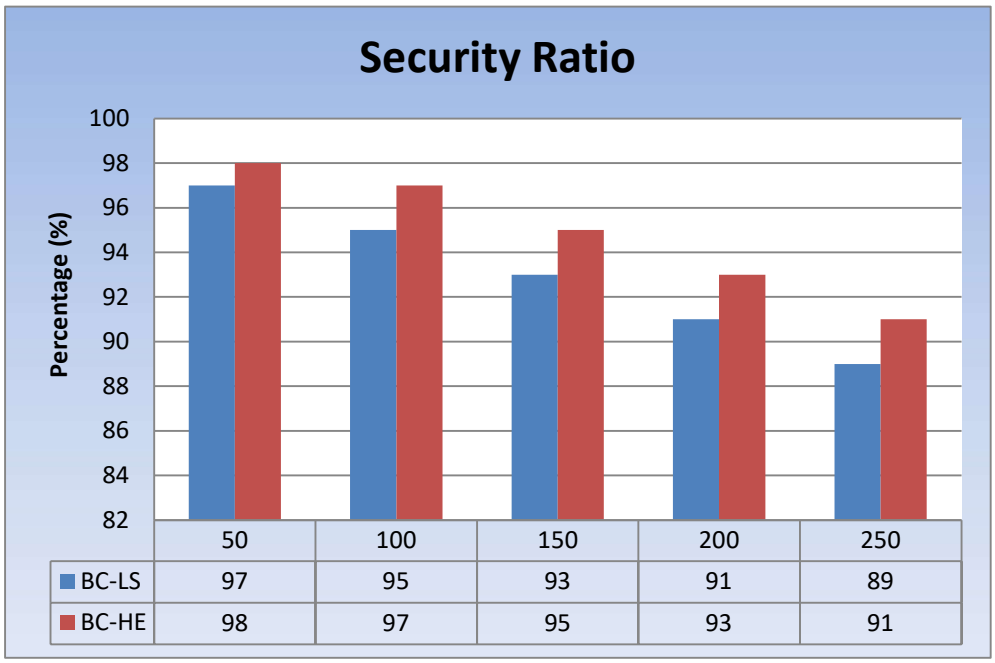

Figure 6: SR Comparison

(ii) Encryption-Time (ET):

The time length it requires for a framework for encryption of EHR through a certain S_Key is measured in "seconds" and is designated as an ET. This does not impart the amount of time devoted to EHR "Input/Output (IO)" processing. The statistical computation that was done to determine ET is illustrated in Equation (2):

$$ET = Number\ of\ EHRs * S\_Key \qquad \text{Eq}\rightarrow 2$$

Table 3 and Figure 7 demonstrate the results when evaluating the ET computations conducted by the currently used BC-LS framework in comparison to the suggested BC-HE framework. Every possible permutation of the patient's EHR throughout the ET is considered in this study. An assortment of "50 to 250" individuals contributed to the EHR compilation. The "X-axis" in Figure 7 shows the time measured in "seconds" that the entire encryption procedure was carried out, while the "Y-axis" shows the average range of EHRs. With a minimum of "50 EHR" in the sample, the current BC-LS framework was found to execute in "1.2 seconds," whereas the suggested BC-HE framework takes "0.9 seconds" for execution. The current BC-LS framework required "19.2 seconds" for a maximum capacity of "250 EHR," whereas the suggested BC-HE framework required "14.4 seconds" for execution. Increasing the number of EHRs

causes the ET to rise steadily. When it comes to ET, the suggested BC-HE framework is far more impactful than the current BC-LS framework.

Table 3: ET Comparison

Average Patient's HER	BC-LS	BC-HE
50	1.2	0.9
100	2.4	1.8
150	4.8	3.6
200	9.6	7.2
250	19.2	14.4

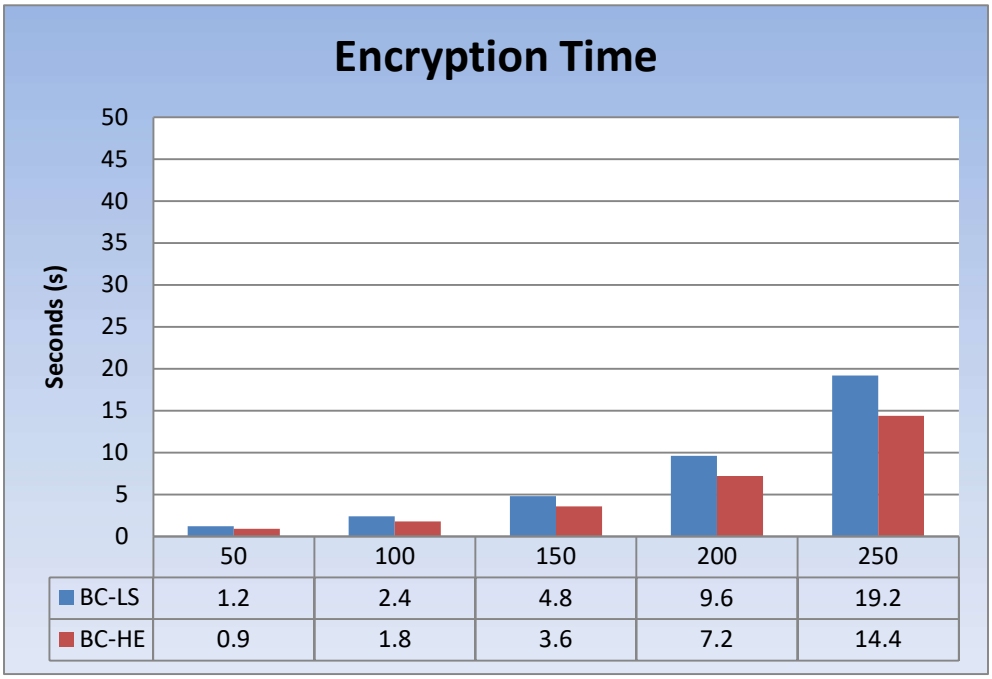

Figure 7: ET Comparison

(iii) Decryption-Time (DT):

The time length it requires for a framework for decryption of EHR through a certain S_Key is measured in "seconds" and is designated as a DT. This does not impart the amount of time devoted to EHR "Input/Output (IO)" processing. The statistical computation that was done to determine DT is illustrated in Equation (3):

$$DT = Number\ of\ EHRs * S\_Key \qquad \text{Eq}\rightarrow 3$$

Table 4: DT Comparison

Average Patient's HER	BC-LS	BC-HE
50	1.3	1.1
100	2.6	2.2
150	5.2	4.4
200	10.4	8.8
250	20.8	17.6

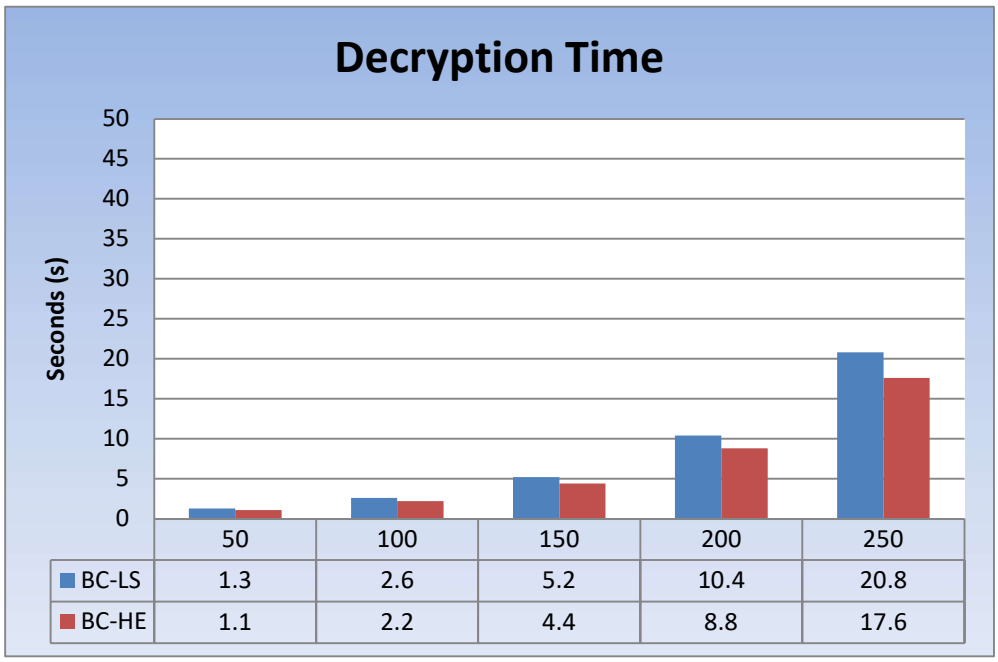

Figure 8: DT Comparison

Table 4 and Figure 8 demonstrate the results when evaluating the DT computations conducted by the currently used BC-LS framework in comparison to the suggested BC-HE framework. Every possible permutation of the patient's EHR throughout the DT is considered in this study. An assortment of "50 to 250" individuals contributed to the EHR compilation. The "X-axis" in Figure 8 shows the time measured in "seconds" that the entire decryption procedure was carried out, while the "Y-axis" shows the average range of EHRs. With a minimum of "50 EHR" in the sample, the current BC-LS framework was found to execute in "1.3 seconds," whereas the suggested BC-HE framework takes "1.1 seconds" for execution. The current BC-LS framework required "20.8 seconds" for a maximum capacity of "250 EHR," whereas the suggested BC-HE framework required "17.6 seconds" for execution. Increasing the number of EHRs causes the DT to rise steadily. When it comes to DT, the suggested BC-HE framework is far more impactful than the current BC-LS framework.

5. CONCLUSION

In this research work, a PP-based BC-HE technique based on a permissioned e-Health BC system has been designed and implemented for a secure EHR storage and access scenario. Since the permissioned BC provides a transparent, immutable, and tamper-resistant environment, the transparency property fails

to provide privacy to DOs. Hence, a mechanism for ensuring the privacy of users besides providing efficient access between patients and other health participants via encryption techniques has been established in this work. The suggested BC-HE framework was deployed and evaluated with Double-Encryption and Double-Decryption on the EHR scenario health data network. The findings from the experimentation prove that the suggested BC-HE framework was highly secure and private to every individual in the e-health system. This framework encrypts all EHRs and stores the hash-values of every EHR transaction in each block of the BC. The BC-HE framework has the potential for ensuring the security, confidentiality, and privacy of the permissioned e-health BC system. To ensure authenticity, ownership rights, and access permission, a Patient-based Access-Control Mechanism will be proposed in future work.

REFERENCES:

[1]. A. Ullah, M. Azeem, H. Ashraf, A. A. Alaboudi, M. Humayun, and N. Jhanjhi, ‘‘Secure healthcare data aggregation and transmission in IoT—A survey,’’ IEEE Access, vol. 9, pp. 16849–16865, 2021, doi:10.1109/ACCESS.2021.3052850.

[2]. S. Qi, Y. Lu, W. Wei, and X. Chen, ‘‘Efficient data access control with fine-grained data protection in cloud-assisted IIoT,’’ IEEE Internet Things J., vol. 8, no. 4, pp. 2886–2899, Feb. 2021, doi:10.1109/JIOT.2020.3020979.

[3]. L. Liu, H. Wang, and Y. Zhang, ‘‘Secure IoT data outsourcing with aggregate statistics and fine-grained access control,’’ IEEE Access, vol. 8, pp. 95057–95067, 2020, doi:10.1109/ACCESS.2019.2961413.

[4]. C. Hahn, J. Kim, H. Kwon, and J. Hur, ‘‘Efficient IoT management with resilience to unauthorized access to cloud storage,’’ IEEE Trans. Cloud Comput., vol. 10, no. 2, pp. 1008–1020, Apr. 2022, doi:10.1109/TCC.2020.2985046.

[5]. J. Zhang, J. Ma, Y. Yang, X. Liu, and N. N. Xiong, ‘‘Revocable and privacy-preserving decentralized data sharing framework for fog-assisted Internet of Things,’’ IEEE Internet Things J., vol. 9, no. 13, pp. 10446–10463, Jul. 2022.

[6]. S. Fugkeaw, L. Wirz, and L. Hak, ‘‘Secure and lightweight blockchain-enabled access control for fog-assisted IoT cloud-based electronic medical records sharing,’’ IEEE Access, vol. 11, pp. 62998–63012, 2023, doi:10.1109/ACCESS.2023.3288332.

[7]. S. Fugkeaw, ‘‘Secure data sharing with efficient key update for industrial cloud-based access control,’’ IEEE Trans. Services Comput., vol. 16, no. 1, pp. 575–587, Jan. 2023, doi:10.1109/TSC.2021.3110828.

[8]. S. Fugkeaw, L. Wirz, and L. Hak, ‘‘An efficient medical records access control with auditable outsourced encryption and decryption,’’ in Proc. 15th Int. Conf. Knowl. Smart Technol. (KST), Feb. 2023, pp. 1–6, doi:10.1109/KST57286.2023.10086904.

[9]. Destin, N. Joy., & Chandra Shekhar Yadav. (2024). A Survey on Secure Framework for Privacy-Preserving Over EHR in Cloud Environment. Nanotechnology Perceptions, 20(S2), 377–394. https://doi.org/10.62441/nano-ntp.v20is2.28.

[10]. J. Zhang, Y. Yang, X. Liu, and J. Ma, ‘‘An efficient blockchain-based hierarchical data sharing for healthcare Internet of Things,’’ IEEE Trans. Ind. Informat., vol. 18, no. 10, pp. 7139–7150, Oct. 2022, doi:10.1109/TII.2022.3145851.

[11]. J. Shen, P. Zeng, K. R. Choo, and C. Li, ‘‘A certificateless provable data possession scheme for cloud-based EHRs,’’ IEEE Trans. Inf. Forensics Security, vol. 18, pp. 1156–1168, 2023, doi:10.1109/TIFS.2023.3236451.

[12]. S.Xu, J.Ning, X.Huang, Y.Li, and G.Xu, ‘‘Untouchable once revoking: A practical and secure dynamic EHR sharing system via the cloud,’’ IEEE Trans. Dependable Secure Comput., vol. 19, no. 6, pp. 3759–3773, Nov. 2022, doi:10.1109/TDSC.2021.3106393.

[13]. P. Chinnasamy and P. Deepalakshmi, ‘‘HCAC-EHR: Hybrid cryptographic access control for secure EHR retrieval in healthcare cloud,’’ J. Ambient Intell. Humanized Comput., vol. 13, no. 2, pp. 1001–1019, Feb. 2022, doi:10.1007/s12652-021-02942-2.

[14]. B. Wang, H. Li, Y. Guo, and J. Wang, ‘‘PPFLHE: A Privacy-Preserving Federated Learning Scheme with Homomorphic Encryption for Healthcare data,’’ Appl. Soft Comput., vol. 146, Oct. 2023, Art. no. 110677, doi:10.1016/j.asoc.2023.110677.

[15]. Y. Liu, Z. Liu, Q. Zhang, J. Su, Z. Cai, and X. Li, ‘‘Blockchain and trusted reputation assessment-based incentive mechanism for healthcare services,’’ Future Gener. Comput. Syst., vol. 154, pp. 59–71, May 2024, doi:10.1016/j.future.2023.12.023.