Cluster Computing (2025) 28:191
https://doi.org/10.1007/s10586-024-04928-z

PriCollabAnalysis: privacy-preserving healthcare collaborative analysis on blockchain using homomorphic encryption and secure multiparty computation

Ahmed M. Tawfik[1] · Ayman Al-Ahwal[2] · Adly S. Tag Eldien[3] · Hala H. Zayed[1,4]

Received: 14 April 2024 / Revised: 3 October 2024 / Accepted: 14 November 2024 / Published online: 21 January 2025

Abstract

Advances in blockchain technology offer a decentralized ledger with transformative potential for healthcare data management, facilitating secure transactions and transparent record-keeping. Nevertheless, the sensitive nature of patient data requires enhanced privacy measures. This paper introduces a comprehensive framework enabling researchers to conduct collaborative statistical analysis on health records while preserving privacy and ensuring security. Statistics are invaluable across various disciplines, guiding consequential decisions based on such analysis. The framework integrates privacy-preserving techniques, including secret-sharing, secure multiparty computation (SMPC), and homomorphic encryption, within a blockchain-based healthcare ecosystem. Patient data is divided using secret-sharing, enabling controlled access. Furthermore, SMPC allows secure data aggregation without revealing individual records, while homomorphic encryption supports computation on encrypted data within smart contracts. Through a series of controlled experiments, we assess the framework's effectiveness in maintaining data privacy, facilitating secure collaboration, and conducting statistical data analysis. The results demonstrate successful preservation of data privacy and secure analysis on a permissioned blockchain using the Hyperledger Fabric platform. Our framework showcases efficient performance while effectively utilizing system resources. This research contributes to the evolution of secure and privacy-conscious healthcare data analysis, paving the way for practical applications and future advancements.

Keywords Blockchain · Privacy · Healthcare · Homomorphic encryption · Secure multiparty computation

1 Introduction

In recent years, the convergence of blockchain technology and healthcare has shown immense potential to revolutionize how patient electronic health records (EHRs) are managed, shared, and utilized [1]. The decentralized and transparent nature of blockchain promises enhanced data integrity, secure transactions, and improved interoperability across healthcare systems [2]. However, the sensitive and private nature of healthcare data demands sophisticated solutions that go beyond traditional blockchain mechanisms to ensure confidentiality, integrity, and patient-centric control.

An inherent challenge lies in the fact that all blockchain participants view an identical ledger, complicating transactions involving confidential data. Typically, access control mechanisms address privacy needs in decentralized networks like blockchain. For instance, the Hyperledger Fabric channel restricts data access to enhance privacy [3]. However, a persistent issue arises as nodes within the same channel handle identical transactions. An alternative approach involves employing public key cryptography. Here, participants encrypt messages using their public keys before submitting them to the ledger. Nonetheless, the

✉ Ahmed M. Tawfik
ahmed.tawfek@fci.bu.edu.eg

[1] Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha, Egypt

[2] Communication and Electronics Department, Pyramids-Institute for Engineering and Technology, Giza, Egypt

[3] Electrical Engineering Department, Faculty of Engineering at Shoubra, Benha, Egypt

[4] Faculty of Engineering, Egypt University of Informatics (EUI), Cairo, Egypt

resulting ciphertexts under different public keys lack collaborative analyzation capabilities.

The limitations related to privacy significantly impede the widespread adoption of blockchain, particularly due to regulations like HIPAA [4] that restrict healthcare providers from sharing patients' data. Consequently, there's a crucial need for a secure system that allows research institutions to analyze private health data accurately while preserving the privacy of patients. For example, geneticists seek to analyze genetic data across a distributed network of patient genomes for personalized medicine research such as exploring individualized treatment options based on genetic information [5]. Fortunately, privacy-preserving cryptographic techniques offer a promising solution to this privacy issue. Therefore, geneticists could perform analyses on encrypted data, ensuring patient privacy. Insights are obtained without exposing sensitive genetic information, supporting advancements in personalized medicine.

In this scenario, there's a necessity for sophisticated privacy-preserving cryptographic techniques that maintain data encryption throughout processing. SMPC [6] distributes computation across multiple parties, ensuring that no individual party can access other parties' data. Similarly, homomorphic encryption [7] enables the execution of analytical functions directly on encrypted data, producing encrypted results identical to those generated from plaintext operations. Each technology possesses its distinct applications. In this study, homomorphic encryption is considered favorable due to its capacity to execute aggregation operations directly, preserving privacy for all participants if required, thereby broadening the range of potential applications. Furthermore, homomorphic encryption ensures confidentiality not only during data computation but also during transmission and storage.

This research delves into integrating advanced privacy-preserving techniques with blockchain technology to establish a robust and privacy-conscious healthcare ecosystem. While storing EHRs directly in the blockchain might not be suitable for large data, the InterPlanetary File System (IPFS) [8] offers a decentralized and robust storage solution that significantly benefits healthcare data management. IPFS, when integrated with blockchain technology and privacy-preserving techniques, distributes data storage across a network of nodes, ensuring high availability and resilience against single-point failures. It operates by using content-based addressing, facilitating efficient retrieval of information based on content rather than location. This approach enables secure and rapid access to healthcare records. Through integration with privacy-preserving techniques such as SMPC and homomorphic encryption, IPFS allows the storage and retrieval of encrypted health data while maintaining patient confidentiality. This integration protects sensitive healthcare data, ensuring encryption and tamper resistance on IPFS, thereby maintaining patient privacy and enhancing data security within the healthcare landscape.

This research aims to make substantial contributions to the field of blockchain technology in healthcare by focusing on the integration of privacy-preserving techniques. The primary contributions of this paper are as follows:

- Integrating Privacy-Preserving Techniques: We integrate privacy-preserving cryptographic techniques, such as secret-sharing, homomorphic encryption, and secure multiparty computation, into blockchain-based healthcare systems to provide a decentralization feature in data processing and sharing among stakeholders, making these techniques applicable in practical real-world systems.
- Facilitating Secure Data Analysis Collaboration: We introduce novel approaches that enable secure collaboration among healthcare providers and researchers, enabling joint data analysis while maintaining individual patient privacy.
- Ensuring Secure Computation: We employ smart contracts (chaincodes) to execute secure multiparty computation among healthcare providers, ensuring decentralization, immutability, and trust within the system, thus eliminating the need for a third party.
- Presenting Statistical Healthcare Data Access Scenarios: We showcase the real-world applicability of our framework by illustrating its effectiveness in scenarios involving collaborative and secure healthcare data analysis through a case study.

The remaining sections of this paper are organized as follows: Sect. 2 presents an overview of related work in the domain encompassing blockchain, privacy, and healthcare. The architecture of our proposed framework is detailed in Sect. 3. The statistical healthcare data access scenario is outlined in Sect. 4. Section 5 focuses on the security and privacy analysis of the proposed framework. The evaluation and experimental results of our solutions are discussed in Sect. 6. Lastly, Sect. 7 summarizes the paper and delineates potential avenues for future research.

2 Related work

The convergence of blockchain technology, privacy-preserving techniques, and healthcare data management has garnered significant attention. Researchers are exploring innovative approaches to preserve patient privacy, ensure data security, and enable secure collaboration in healthcare. Here, we present an overview of relevant work in this field.

Blockchain technology's potential to revolutionize healthcare data management has spurred numerous investigations. Health information exchange, data interoperability, and secure patient data sharing have been focal points of research. The utilization of privacy-preserving techniques within healthcare data management is necessary to protect sensitive patient information. Secret-sharing [9], SMPC [6], and homomorphic encryption [7] have emerged as promising solutions. Secret-sharing [9] is a fundamental technique in cryptography that divides a confidential data point into multiple shares, distributed among different participants. The technique provides an added layer of security, particularly when sharing critical patient information within a distributed network. Yang et al. [10] presented a privacy-preserving EHR system that combines secret-sharing and blockchain for secure data sharing.

The field of SMPC [6] has been an active area of research for several decades. Introduced by Andrew Yao in 1982, the concept emerged through the formulation of what is now known as Yao's Millionaires' problem [11]. The problem elucidates a scenario involving two millionaires who aim to discern the wealthier individual without divulging their actual fortunes. The resolution involves leveraging one-way functions within interactive communications between the involved parties. In a subsequent work [12] in 1986, Yao presented a more comprehensive solution, discussing the generation of a random integer $N = p \cdot q$. The integer's secret components (p, q) are concealed from each party individually but are recoverable jointly when necessary. Yao also proposed workarounds for enabling secure computations between two parties. Building on Yao's pioneering works, Goldreich, Micali, and Wigderson introduced two secure multi-party computation methods in 1987 [13].

The concept of SMPC has gained prominence as a means of facilitating collaborative data analysis without revealing raw data [14–17]. The approach empowers multiple healthcare providers to contribute their data for joint analysis without exposing sensitive patient information. Authorized researchers can collectively compute aggregated statistics or perform analyses without exposing the raw data. SMPC protocols ensure that no party gains insights into others' contributions, thereby reinforcing data confidentiality in collaborative scenarios. Researchers increasingly recognize the necessity of integrating privacy-preserving techniques with blockchain to enhance healthcare data management. A recent study by Li et al. [14] proposed a survey on blockchain-based methods employing SMPC for collaborative medical research while safeguarding individual data privacy.

Homomorphic encryption [7] is leveraged to facilitate computation on encrypted data within blockchain-based healthcare systems. The technique is particularly valuable for preserving patient privacy during data-driven analyses and computations [18–21]. Zhang et al. [18] explored the use of homomorphic encryption for secure computation over encrypted medical data, enabling private analysis and diagnosis.

Some researchers have focused on Hyperledger Fabric [3], an idealistic permissioned blockchain maintained by the Linux Foundation. Ghadamyari et al. [22] introduced a framework for conducting statistical analysis on private health data using blockchain technology and the Paillier encryption algorithm [23]. The method operates within a blockchain network involving data custodians and researchers, employing smart contracts to perform computations while maintaining data privacy. The proposed approach showcases computations for count and mean statistical functions, ensuring that only the researcher can decrypt the encrypted data stored on the ledger. While the approach used the Paillier cryptosystem to achieve data privacy, it encrypted data owned by different participants without segmenting them by the requester's public key. The practice could lead to the exposure of parties' privacy in scenarios where the owner of the private key and the invoker of the smart contract are compromised or if the requester employs a man-in-the-middle attack.

Zhou et al. [16] cover various aspects of SMPC protocols integrated into the Hyperledger Fabric blockchain platform [3]. They emphasize the utilization of homomorphic encryption, secret sharing, and zero-knowledge proofs to safeguard privacy and guarantee the accuracy and verifiability of computations. These computations involve off-chain preprocessing to generate quadruples and on-chain computation utilizing a hierarchical multiplication gate. However, the method exhibits limited scalability in larger networks, with numerous computations that could be streamlined. Furthermore, it does not address practical challenges, such as enabling statistical analysis on healthcare data with assured privacy.

The authors in [15–17, 19–22] have introduced privacy-preserving cryptographic techniques and protocols aimed at safeguarding data privacy. However, their proposed protocols often necessitate extensive interaction and numerous data exchanges among the involved parties. Building upon this foundational research, our work presents a holistic framework that integrates additive homomorphic encryption [23], secret-sharing [9], and SMPC [6] techniques with blockchain technology to establish a privacy-conscious healthcare framework. We emphasize the importance of efficient implementations and regulatory compliance. This innovative integration of techniques addresses the identified gaps in existing research, such as vulnerabilities to man-in-the-middle attacks and the need for enabling statistical analysis of healthcare data with assured privacy. It offers a comprehensive solution for secure and private healthcare data management.

The subsequent section will elaborate on the architecture of our proposed framework, named “PriCollabAnalysis”, demonstrating how these techniques are employed to achieve a privacy-enhanced and secure healthcare ecosystem.

3 PriCollabAnalysis: proposed framework

Our proposed framework leverages blockchain technology to enhance data accessibility, integrity, and decentralized trust through smart contracts. We implement additive homomorphic encryption [23] for secure computation on encrypted data and employ the SMPC protocol to enable collaborative data analysis without exposing raw data. This framework is tailored for a network involving multiple healthcare providers and researchers.

Healthcare providers are assumed to operate within a blockchain network channel, contributing to a shared and distributed ledger. Concurrently, researchers utilize front-end applications integrated with network-provided APIs for communication. Researchers can send specific query requests to healthcare providers. Each healthcare provider computes query results and encrypts them using additive homomorphic encryption [23]. Smart contracts are utilized to compute outcomes while preserving data privacy. Participants in the blockchain network, such as healthcare providers and insurance companies, serve as validators of the channel, enabling access to data stored on the ledger. In contrast, researchers, considered end-users of the network, have restricted authentication access.

3.1 The proposed framework overview

In this section, we have outlined the primary structure of the proposed framework. Figure 1 illustrates the suggested architecture for facilitating collaborative healthcare analysis. Utilizing a permissioned blockchain, we maintain the framework using a Hyperledger Fabric platform [3]. We elaborate on the privacy-preserving cryptographic algorithms employed within our protocol, including secret-sharing [9], SMPC [6], and additive homomorphic encryption [23].

The proposed framework comprises eight sequential steps, delineated as follows:

1. Initially, the requester’s identity is verified at one of the participating hospitals to initiate collaborative healthcare analysis.
2. The selected participating hospital submits a proposal to the blockchain network for the SMPC process using the smart contract (chaincode).
3. Other participating hospitals join in the SMPC process by enrolling in the chaincode’s SMPC request.
4. The participating hospitals compute the query’s result, divide it into shares using the secret-sharing technique according to the number of participating hospitals, encrypt these shares using the public keys of other participating hospitals, and subsequently submit the encrypted shares through the chaincode.
5. After the chaincode transmits the encrypted shares, each hospital receives its required shares, decrypts them using its secret key, and then accumulates the decrypted shares.
6. Using the additive homomorphic encryption, each hospital encrypts the accumulated value with the generated chaincode’s public key HE_P_k and then sends the encrypted result to the chaincode.
7. After receiving the encrypted values from all participating hospitals, the chaincode aggregates these encrypted values using the additive homomorphic encryption. Then, it decrypts the aggregated outcome using its secret key HE_S_k.
8. Finally, upon the chaincode’s dispatch of the finalized result, the requester receives the final result.

3.2 Privacy-preserving techniques

In this subsection, we elaborate on the privacy-preserving cryptographic techniques employed within our protocol, including secret-sharing [9], additive homomorphic encryption [23], and SMPC [6].

3.2.1 Secret-sharing

Shamir’s Secret Sharing scheme [9] divides a secret into multiple shares, distributing them among participants in such a way that reconstructing the secret requires a minimum threshold of shares. This threshold determines the minimum number of shares necessary to reconstruct the secret, while any subset below this threshold remains insufficient to unveil the secret information.

Additive Secret Sharing Equation:

$$y_i = (a_0 + a_1 \cdot x_i + a_2 \cdot x_i^2 + \ldots + a_{t-1} \cdot x_i^{t-1}) \bmod p \tag{1}$$

Equation for Reconstructing the Secret:

$$a_0 = \sum_{i=1}^{t} y_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^{t} \frac{x - x_j}{x_i - x_j} \bmod p \tag{2}$$

In these equations:

- y_i represents the share given to participant i.
- a_0 is the secret to be shared.
- $a_1, a_2, \ldots, a_{t-1}$ are randomly chosen coefficients from a finite field.
- x_i is the unique value assigned to participant i.
- t indicates the threshold required to reconstruct the secret.
- p is the modulus, often a prime number.
- x is the variable used in the reconstruction equation.

This formula calculates the secret a_0 by summing up a weighted combination of the shares y_i, where each term in the sum consists of the share y_i multiplied by a product of x terms that depend on the other x-values. When t shares are combined, the secret a_0 can be accurately reconstructed.

3.2.2 Additive homomorphic encryption

Additive homomorphic encryption [23] is a type of homomorphic encryption scheme where mathematical operations performed on the encrypted data yield an encrypted result that, when decrypted, corresponds to the result of performing the same operations on the plaintext. The additive homomorphic encryption scheme used in our framework is the Paillier cryptosystem [23], which is partially homomorphic encryption and supports only the addition of encrypted values.

Key Generation Parameters:

- Randomly choose two large prime numbers p and q.
- Compute $n = p \times q$ (public key component).

Compute Other Key Components:

- $\lambda = \mathrm{lcm}(p-1, q-1)$ (used for decryption), where lcm means the least common multiple.
- Choose a random number g such that $\gcd(L(g^{\lambda} \bmod n^2), n) = 1$ (public key component), where gcd means the greatest common divisor and $L(x) = \frac{x-1}{n}$.
- Calculate the private key μ such that $\mu = L(g^{\lambda} \bmod n^2)^{-1} \bmod n$ (private key component).

Encryption:

$$c = g^m \cdot r^n \bmod n^2 \tag{3}$$

Decryption:

$$m = L(c^{\lambda} \bmod n^2) \cdot \mu \bmod n \tag{4}$$

Addition of Encrypted Values:

Let c_1 and c_2 be the ciphertext of m_1 and m_2, respectively.

$$c_3 = c_1 \cdot c_2 \bmod n^2 \tag{5}$$

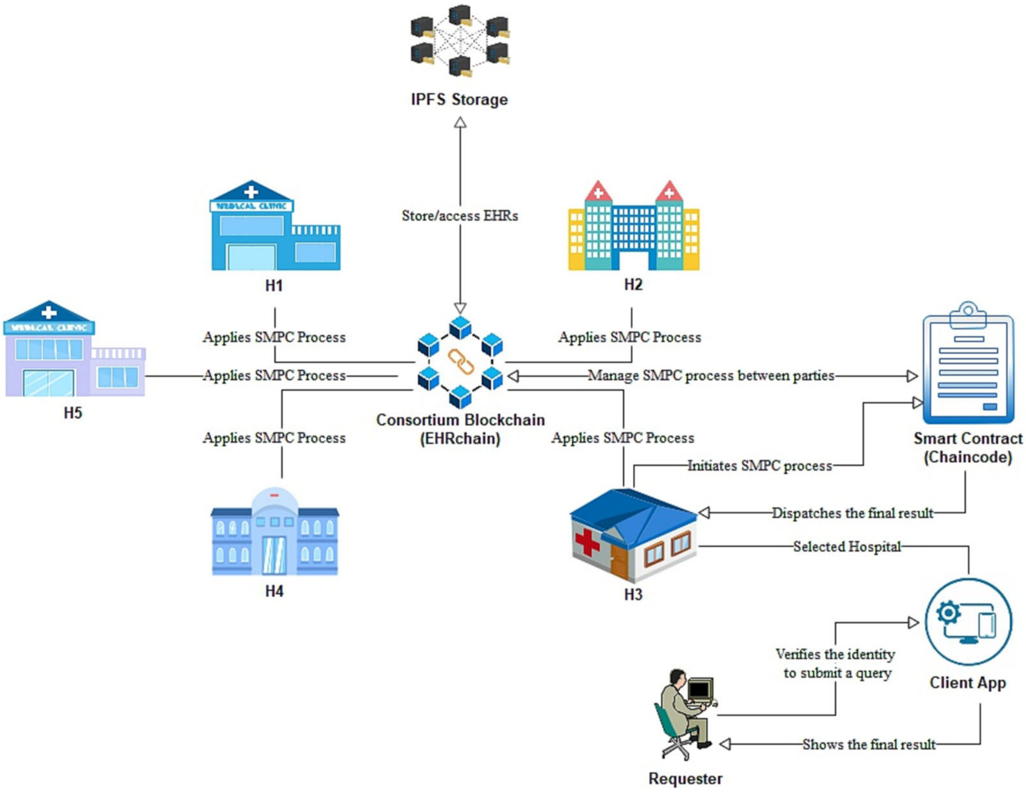


Fig. 1 The architecture of the proposed framework

The addition of these ciphertexts yields a new ciphertext c_3 representing the sum m_1 and m_2 without decrypting the individual ciphertexts

3.2.3 Secure multiparty computation

Our framework enables secure collaboration among healthcare providers and researchers. Secure Multiparty Computation (SMPC) [6] involves multiple participants, each holding their private data, and collectively computing a function over these private inputs without directly disclosing them. The SMPC protocol enables computations while maintaining the privacy of each party's data.

The steps involved in an SMPC protocol typically include:

1. Input Sharing: Each party privately splits its input data S into n shares or parts $S_1, S_2, ..., S_n$, distributing them among the participating parties. In a (k, n)-threshold scheme, any subset of k shares is sufficient to reconstruct the secret, while fewer than k shares reveal partial information about S:

$$S = S_1 \oplus S_2 \oplus \ldots \oplus S_n \tag{6}$$

2. Secure Computation: Parties collaboratively perform computations on their respective shares S_i to collectively compute the desired function S without revealing any individual party's input.
3. Result Reconstruction: After the computation, the parties combine their computed shares S_i to obtain the result S. In a (k, n)-threshold scheme, any set of k shares or more, $S_1, S_2, ..., S_k$, can be used to reconstruct the original secret:

$$S = S_1 \oplus S_2 \oplus \ldots \oplus S_k \tag{7}$$

Our proposed framework employs cryptographic techniques to facilitate secure computations over distributed data. In the next section, we will delve into a real-world scenario to illustrate the practical applicability of the proposed framework.

4 Statistical healthcare data access scenario

In this section, we present case studies that highlight the practical application and effectiveness of our proposed framework in addressing real-world healthcare challenges while ensuring data privacy, patient control, and secure collaboration.

4.1 Scenario

The requester, such as the Ministry of Health, leverages the blockchain framework to securely analyze medical data, gaining valuable insights for healthcare decision-making. We simulate the participation of five healthcare provider organizations, although real-world scenarios may involve more, all built on the Hyperledger Fabric platform. Within each organization, patient health records are partitioned using secret-sharing, and collaborative analysis are performed using SMPC facilitated through the smart contract (chaincode) within Hyperledger Fabric. Homomorphic encryption is employed to perform computations on encrypted data.

4.2 Framework implementation

In this section, we illustrate the procedure followed for secure computation using the SMPC technique within the blockchain framework. The smart contract (chaincode) coordinates computations among participants without relying on a TTP or exposing raw data. The results are obtained through the following steps, as depicted in Fig. 2:

- **Step 01:** The requester's identity is verified on the blockchain network through the authentication process.
- **Step 02:** The requester, denoted as R, is seeking specific statistical information about patients infected with diseases such as COVID-19, diabetes, cancer, or others.
- **Step 03:** Once the requester R is authenticated through one of the participating hospitals P_i, the selected hospital initiates the SMPC process via a chaincode on the blockchain network.
- **Step 04:** The chaincode invites the other hospitals to enroll in the SMPC process.
- **Step 05:** The chaincode prompts each participating hospital P_i to retrieve the patients' EHRs from IPFS storage and decrypt them using its private key $Priv_{ki}$ of the Advanced Encryption Standard (AES) [24]. $\mathrm{EHR}'_i \leftarrow \mathrm{IPFS}(\mathrm{hash})$ $\mathrm{EHR}_i \leftarrow \mathrm{Dec}(\mathrm{EHR}'_i, Priv_{ki}), \quad i \in \{1, \ldots, N\}$
- **Step 06:** The chaincode instructs each participating hospital P_i to divide its healthcare data result S_i associated with the specified disease into N shares S_{ij}, where $j = 1, 2, ..., N$, determined by the total number of enrolled participating hospitals N. $S_i = S_{iN} + \sum_{j=1}^{N-1} S_{ij} (\mathrm{mod\ p})$
- **Step 07:** Each participating hospital P_i encrypts $(N - 1)$ shares by utilizing the public keys pk_j, where j ranges from 1 to $N - 1$, belonging to the other participating hospitals $(N - 1)$. $S'_{ij} \leftarrow \mathrm{Enc}(S_{ij}, \mathrm{pk}_j), \quad j = 1, 2, \ldots, N - 1$

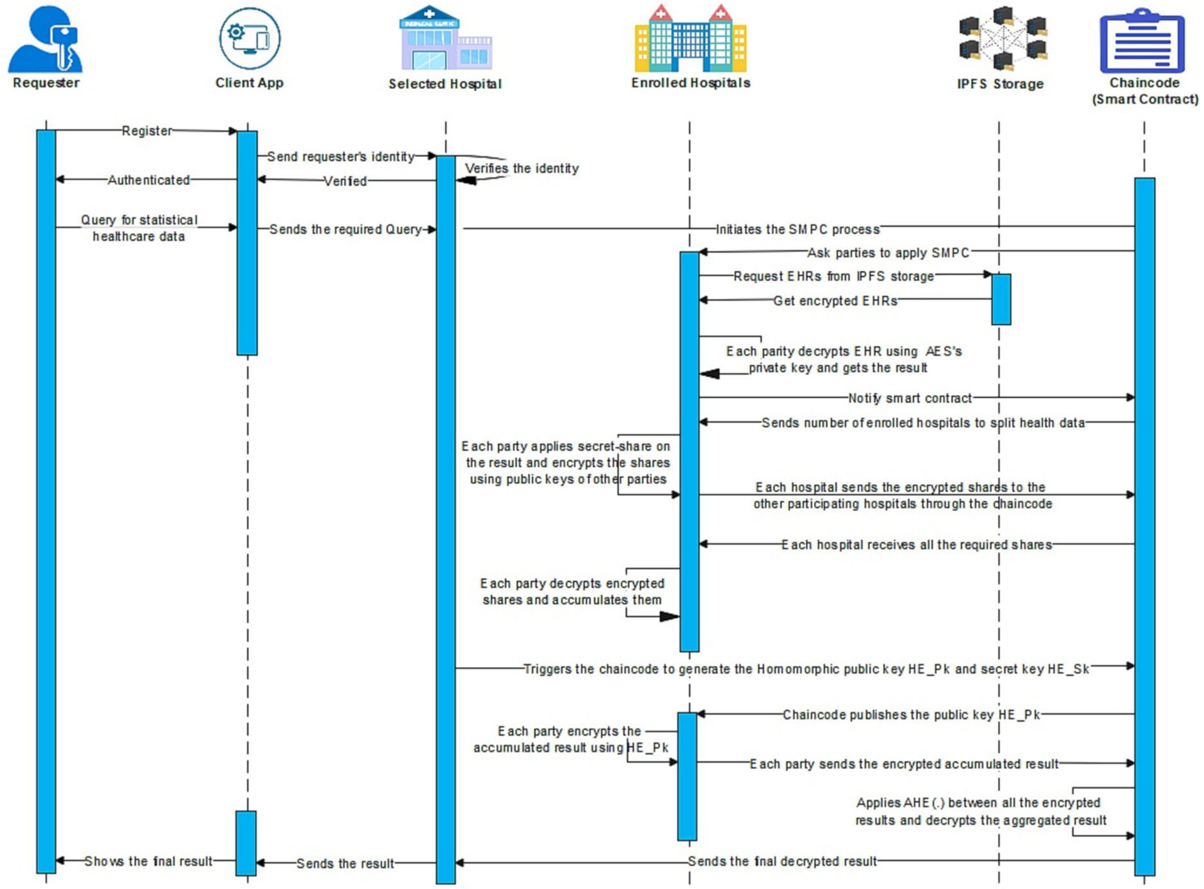

Requester
Client App
Selected Hospital
Enrolled Hospitals
IPFS Storage
Chaincode
(Smart Contract)
Register
Send requester's identity
Verifies the identity
Authenticated
Verifed
Query for statistical
healthcare data
Sends the required Query
Initiates the SMPC process
Ask parties to apply SMPC
Request EHRs from IPFS storage
Get encrypted EHRs
Each parity decrypts EHR using AES's
private key and gets the result
Notify smart contract
Sends number of enrolled hospitals to split health data
Each party applies secret-share on
the result and encrypts the shares
using public keys of other parties
Each hospital sends the encrypted shards to the
other participating hospitals through the chaincode
Each hospital receives all the required shares
Each party decrypts encrypted
shares and accumulates them
Triggers the chaincode to generate the Homomorphic public key HE_Pk and secret key HE_Sk
Chaincode publishes the public key HE_Pk
Each party encrypts the
accumulated result using HE_Pk
Each party sends the encrypted accumulated result
Applies AHE (.) between all the encrypted
results and decrypts the aggregated result
Shows the final result
Sends the result
Sends the final decrypted result

Fig. 2 Sequence diagram of statistical healthcare data access scenario

- **Step 08:** Each participating hospital P_i forwards the encrypted shares S'_{ij}, where j ranges from 1 to $N-1$, to the corresponding participating hospital P_j following the guidelines outlined in Algorithm 1. $P_j \leftarrow \{S'_{ij}\}, \quad j = 1, 2, \ldots, N-1$
- **Step 09:** When each participating hospital P_i has received all the necessary shares S'_{ij}, it proceeds to decrypt them using its respective secret key sk_i. $S_{ij} \leftarrow \mathrm{Dec}(S'_{ij}, sk_i), \quad j \in \{1, \ldots, N-1\}$
- **Step 10:** After each participating hospital P_i decrypts all the received shares S_{ij} (where $j = 1, 2, \ldots, N-1$), it accumulates these decrypted shares with its first share to construct an anonymized accumulated value S_i using a threshold of N. $S_i \leftarrow S_{i1} \oplus S_{i2} \oplus \ldots \oplus S_{iN}$
- **Step 11:** The invoking hospital triggers the chaincode to generate the public key ($\mathrm{HE\_}P_K$) and secret key ($\mathrm{HE\_}S_k$) for additive homomorphic encryption, which are required to proceed with the SMPC process on healthcare data.
- **Step 12:** The chaincode notifies each participating hospital P_i to encrypt their accumulated values S_i using the public key ($\mathrm{HE\_}P_K$) generated by the chaincode. Subsequently, the encrypted value S''_i is sent to the chaincode to apply the additive homomorphic encryption. $S''_i \leftarrow \mathrm{Enc}(S_i, \mathrm{HE\_}P_K), \quad i \in \{1, \ldots, N\}$
- **Step 13:** When the chaincode receives the encrypted values S''_i from all participating hospitals (N), it performs the additive homomorphic encryption $(\cdot)$ operation on them. $S'' = S''_1 \cdot S''_2 \cdot \ldots \cdot S''_N$
- **Step 14:** After applying the additive homomorphic encryption $(\cdot)$ and obtaining the aggregated encrypted result S'', the chaincode proceeds to decrypt the final result before sending it to the invoker hospital. $S \leftarrow \mathrm{Dec}(S'', \mathrm{HE\_}S_K)$
- **Step 15:** The chaincode forwards the decrypted final result S to the invoking hospital, which subsequently transmits it to the requester R. $R \leftarrow S$

Requesters analyze aggregated data without compromising patient confidentiality, ensuring meticulous data privacy. The results, shared in encrypted form, safeguard individual confidentiality while enhancing collaborative insights and minimizing the risk of unauthorized access. This case study underscores the practical viability of our proposed framework in addressing critical healthcare scenarios. By seamlessly integrating blockchain technology with privacy-

Algorithm 1 Encrypt shares and send

```
function ENCRYPTSHAREANDSEND($shares$, $hospitalId$)
    $hospitalsNum \leftarrow$ length(getRegisterHospitals())
    $index \leftarrow 1$
    $hId \leftarrow$ parseInt($hospitalId$)
    while index $\neq$ (hospitalsNum $-$ 1) do
        if hId == 0 then
            hId $\leftarrow$ hospitalsNum $-$ 1
        end if
        hId $\leftarrow$ hId $-$ 1
        print "Share Index: " + index + " Sent to hospital: " + hId
        ENCRYPTSHARE($shares$[index], index, hospitalsPK[hId], hId)
        index $\leftarrow$ index + 1
    end while
end function
function ENCRYPTSHARE($share$, $shareIndex$, $hospitalPublicKey$, $hospitalId$)
    if parseInt($share$) exists then
        $shareBI \leftarrow$ convertToBigInteger(parseInt($share$))
        $encryptedShare \leftarrow$ PaillierCipher.encrypt($shareBI$, hospitalPublicKey)
        print "Sending share to the hospitalId: " + hospitalId
        SENDSHARETOHOSPITAL(hospitalId, shareIndex, encryptedShare)
    else
        print "Exception: Share is invalid."
    end if
end function
function SENDSHARETOHOSPITAL($hospitalId$, $shareIndex$, $encryptedShare$)
    if contractEHR.submitTransaction exists then
        contractEHR.submitTransaction("sendHospitalShare", hospitalId, shareIndex, encryptedShare)
    else
        print "Exception: Transaction submission failed."
    end if
end function
```

preserving techniques, it empowers patients, facilitates secure collaboration, and enables data-driven research, all while maintaining strict data privacy and security standards.

5 Security and privacy analysis

In this section, we present a formal analysis of the security and privacy aspects of the PriCollabAnalysis framework. We define several scenarios, along with corresponding theorems and proofs, to demonstrate the robustness of our approach in safeguarding healthcare data privacy and ensuring secure collaboration.

Scenario 1: confidentiality preservation Patients or healthcare providers encrypt their data using the AES cryptography algorithm before storing it on IPFS. They use individually generated AES symmetric keys for encryption and decryption.

Theorem 1 *Utilizing AES encryption ensures data confidentiality within the framework.*

Proof Let K represent the set of all possible AES symmetric keys. For any given key $k \in K$, the encryption function $E(k, m)$ encrypts a message m using key k, yielding ciphertext c. Similarly, the decryption function $D(k, c)$ decrypts ciphertext c using key k, producing the original message m. Therefore, only authorized entities possessing the correct key can decrypt the data, ensuring confidentiality.

Scenario 2: integrity verification

The framework compares the hash of encrypted data stored on the ledger with the hash derived from the encrypted data obtained from IPFS storage to verify data integrity.

Theorem 2 *Comparing hashes ensures data integrity within the framework.*

Proof Let H denote the set of all possible hash values. For any encrypted data d, let $h_1 = H(E(d))$ represent the hash generated from the encrypted data stored on the ledger, and $h_2 = H(E'(d))$ represent the hash derived from the encrypted data obtained from IPFS storage. If $h_1 = h_2$, the data remains unchanged and retains its integrity. Conversely, if $h_1 \neq h_2$, indicating a mismatch, potential data corruption is detected.

Scenario 3: availability assurance

Let D_{IPFS} represent the decentralized storage provided by IPFS.

Theorem 3 *The availability of healthcare data is ensured through blockchain decentralization and IPFS decentralized storage.*

Proof Blockchain decentralization, exemplified by Hyperledger Fabric, mitigates the risk of a single point of failure, thereby enhancing overall system availability. Additionally, IPFS decentralized storage (D_{IPFS}) further ensures data availability by distributing data across multiple nodes, reducing reliance on centralized servers.

Scenario 4: data privacy preservation

The framework employs secret-sharing and homomorphic encryption techniques to preserve data privacy.

Theorem 4 *Secret-sharing and homomorphic encryption techniques preserve data privacy within the framework.*

Proof Let P represent the set of all participating entities. Each entity i divides its data into shares based on the number of registered parties, denoted as N, encrypting the shares using other parties' public keys. This process can be represented as:

$$S_i = S_{iN} + \sum_{j=1}^{N-1} S_{ij} (\text{mod p})$$

where S_{ij} represents the share of entity i encrypted using the public key of entity j, denoted as pk_j. Each entity holds its share S_{ij} along with shares from other entities, making individual shares meaningless without integration through SMPC via the chaincode. Additionally, the integration of additive homomorphic encryption enables computations on encrypted shares without revealing sensitive information. Reconstruction of the secret requires a threshold number of shares, maintaining data privacy.

Scenario 5: eliminating trusted third parties

The framework eliminates the need for a TTP by employing chaincode within the blockchain network.

Theorem 5 *The use of chaincode ensures decentralized trust establishment within the framework.*

Proof Let C represent the set of chaincode instances. Chaincode orchestrates the execution of SMPC processes among multiple participating entities, facilitating decentralized and autonomous participation, eliminating reliance on a central entity. Each chaincode instance c operates independently, contributing to the decentralized nature of the blockchain. The replication of chaincode across multiple network nodes ensures fault tolerance and reliability, enhancing the framework's overall trustworthiness. Additionally, the execution of chaincode by multiple nodes in the network ensures consensus-driven collaboration, promoting regulatory compliance and secure data analysis while upholding privacy and confidentiality.

After establishing the framework's security and privacy measures, we now evaluate its real-world performance. In the following section, we present our comprehensive evaluation and insights on scalability, efficiency, and practical applicability in healthcare data management.

6 Evaluation and experimental results

In this section, we assess the performance, efficiency, and effectiveness of our proposed framework. To evaluate its effectiveness, we developed and implemented the PriCollabAnalysis framework, specifically designed for collaborative healthcare data analysis among providers. Deployed on Hyperledger Fabric [3], an open-source blockchain platform known for its adaptable architecture, our framework emphasizes privacy. It achieves this through tailored cryptographic privacy-preserving techniques embedded within a dedicated smart contract (chaincode) that meets precise business requirements.

The integration of the IPFS storage system ensures decentralization and scalability within the framework. To achieve consensus among nodes, our framework utilizes the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm [25]. This algorithm guarantees transaction integrity and consensus among network participants, complemented by the Raft ordering service [26]. It enables multiple nodes to collaboratively manage blockchain transactions, establishing consensus on their order. The simplified deployment and operation of ordering service nodes in a Fabric network make Raft particularly suitable for production environments.

Following the framework's implementation, performance tests were conducted to validate its practical applicability. These tests involved measuring time consumption, reflecting the duration of each operation's execution. This metric accounts for the computational cost required for transaction execution on the blockchain network. Each test was iterated 10 times, and the average value was considered the representative result.

Table 1 Test environment for experiments

Component	Configuration
Distribution	Single machine with multiple dockers
Machine operating system	Ubuntu Linux 22.04 LTS
CPU	Intel Core i7, 2.8 GHz
Storage	16 GB memory, 256 GB SSD
Platform	Hyperledger Fabric 2.4.7
Size	5 Organizations
Orderer Service	Raft-based ordering service
On-chain	CouchDB
Off-chain	IPFS distributed storage

6.1 Experimental setup

The experiments were conducted in a test environment detailed in Table 1. In this setup, individual peers functioned within separate Docker containers, while the chaincode operated within signed Docker containers on the same machine. Interaction with the blockchain network occurred through the Hyperledger Fabric SDK for Java. Throughout the experiments, each organization operated with a single peer, with these peers associated with various participants. The setup designated all organizations as part of the endorsement policy, aligning the count of participants with the number of computing nodes.

The performance evaluation involved a diverse range of healthcare providers, researchers, and healthcare data records to ensure representative results. Researchers necessitated specific collaborative healthcare data analysis from healthcare providers. The experimental setup comprised the following components:

1. IPFS Off-chain Distributed Storage System: This system was utilized to securely store encrypted healthcare data.
2. Node Configuration: The system involved five nodes - N1, N2, N3, N4, and N5. Each node, representing a hospital, consisted of:
    - One peer (Peer0): Responsible for maintaining a copy of the blockchain ledger and participating in the consensus process.
    - One Certificate Authority: Handling the issuance and management of digital certificates for secure network communication and identity verification.
    - On-chain database (CouchDB): Storing healthcare data records and offering efficient querying capabilities.
    - Clients: Each node had at least two types of clients:
        - Healthcare provider: Representing healthcare professionals within the hospital and interacting with the system to access and update patient data.
        - Requester: Representing individuals requiring healthcare data analysis.
3. Orderer Service: This component facilitated consensus among network participants and ordered transactions into a consistent ledger.
4. Dedicated Chain for EHR Sharing: A single channel hosted one permissioned chain - EHRChain - with its blocks and real state (CouchDB). This chain aimed to securely share EHRs between patients and healthcare providers. It was associated with a chaincode providing functionality and logic for handling EHR-related operations, such as secure storage, retrieval, and updating.

6.2 Experimental phases

To facilitate performance experiments using Hyperledger Caliper [27] and our tailored Fabric Client SDK interface, the experimental settings are defined across the following phases:

(1) **Performance metrics in collaborative analysis:** This experiment aims to assess different performance indicators of our network, specifically focusing on resource consumption such as CPU and memory usage. The evaluation is based on the network configuration presented in Table 1.

(2) **Different homomorphic encryption key sizes:** The key size in cryptosystems plays a crucial role in determining system performance. A trade-off exists between key size and security: smaller keys enable faster operations but increase vulnerability to brute-force attacks. In this experiment, we evaluated how varying key sizes impact the running time of additive homomorphic encryption using the Paillier Cryptosystem. Our results were compared to the method proposed by Zhou et al. [28]. Four distinct key sizes-512, 1024, 2048, and 4096 bits-were tested within our framework, following the approach of Zhou et al. [28], to assess their effect on performance. These keys were generated and applied in 10 rounds of secure healthcare data analysis, with the overall average performance recorded for each key size.

(3) **Response times with increasing participating nodes:** The proposed framework operates in a decentralized environment where each organization contributes to the computations. As the number of participating nodes increases from 2 to 20, the computational load naturally grows, resulting in an expected increase in overall execution time. In this experiment, we compared our measured response times with the methods proposed by Zhou et al. [28] and Ghadamyari et al. [22], evaluating the performance of secure collaborative data analysis as the number of nodes progressively increased.

6.3 Results and discussions

The experimental results indicate the following:

(1) **Evaluation of performance metrics:** The performance analysis, as illustrated in Fig. 3, reveals the CPU usage and memory consumption across different components of the framework, with a maximum

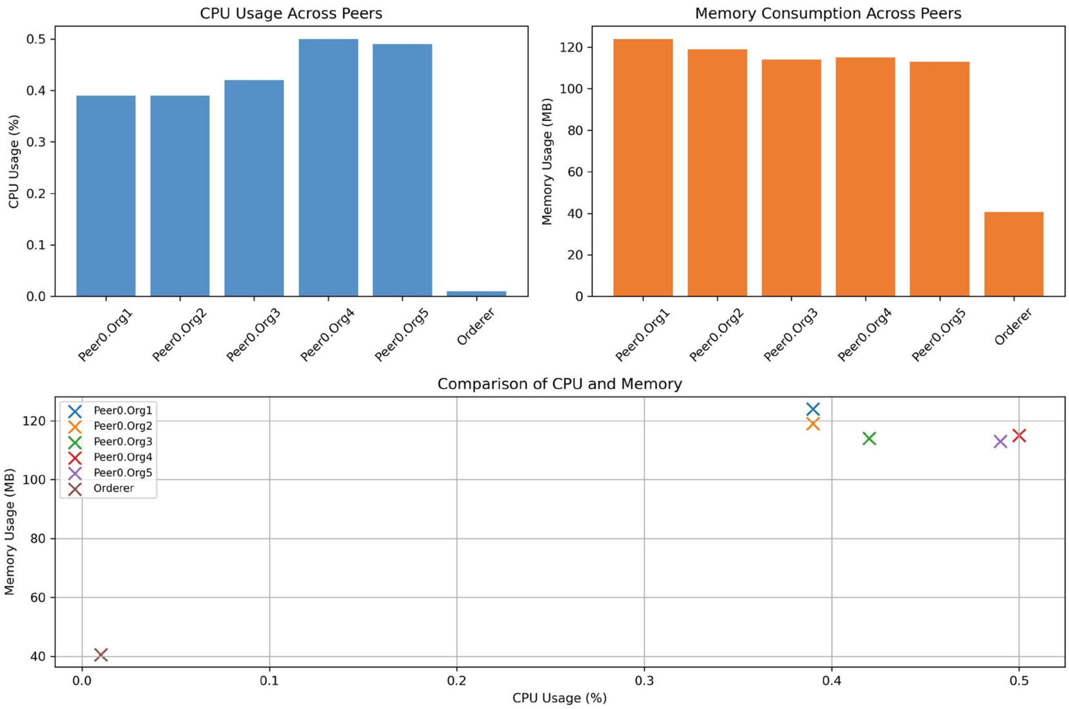
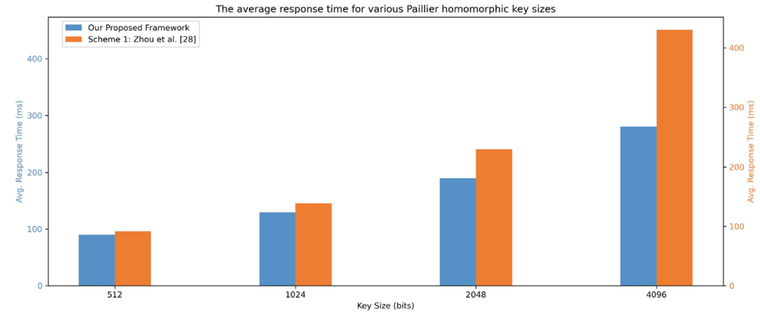


Fig. 3 Analysis of resource consumption in the proposed framework

Fig. 4 The average response time for various additive homomorphic key sizes in comparison to the scheme proposed by Zhou et al. [28]



CPU usage of 0.49% for ’Peer0.Org5’ and peak memory consumption of 124 MB for ’Peer0.Org1’. The framework demonstrates efficient resource utilization, with consistently low memory and CPU usage across all peers and the orderer component. These findings highlight the framework’s effectiveness in resource management, making it well-suited for practical deployment in healthcare data analysis scenarios.

(2) **Computation time evaluation using various homomorphic key sizes:** Fig. 4 presents an in-depth comparison of response times for securely executing the designated function using our SMPC protocol in a test environment detailed in Table 1. The comparison includes the scheme proposed by Zhou et al. [28] across different additive homomorphic key sizes. As the key size increases from 512 bits to 4096 bits, the total response time also increases, demonstrating the expected trade-off between security and computational efficiency. Importantly, while our framework shows longer response times with larger key sizes, it consistently outperforms the method proposed by Zhou et al. [28]. This marginal difference underscores our framework’s ability to maintain competitive efficiency even as security demands increase. These results confirm the optimization of our computation protocol, ensuring both result accuracy and the preservation of health data privacy. Based on the National Institute of Standards and Technology (NIST) [29] recommendation, a 2048-bit key is adopted for subsequent experiments to strike a balance between security and performance.

(3) **The evaluation of response times across different computing nodes:** The experiment analyzing response times across different computing nodes

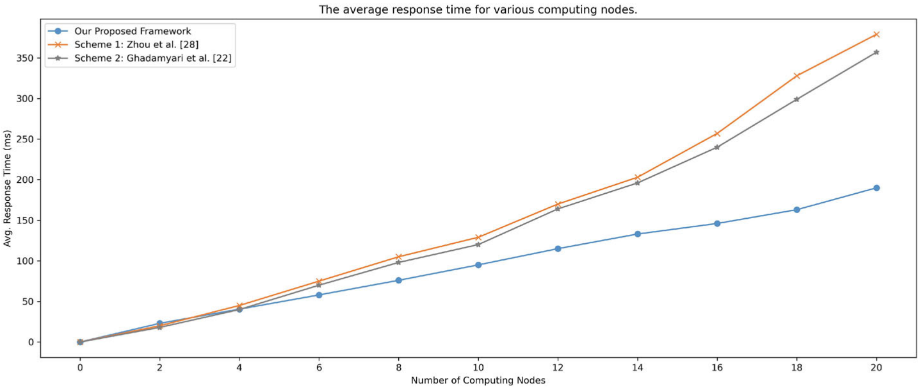


Fig. 5 The average response time for various computing nodes in comparison to the schemes proposed by Zhou et al. [28] and Ghadamyari et al. [22]

offers valuable insights into the scalability and performance of our proposed framework compared to schemes by Zhou et al. [28] and Ghadamyari et al. [22]. The number of participating nodes, each representing a distinct hospital, was varied to examine the impact on collaborative data analysis, as shown in Fig. 5. Our framework consistently delivered competitive average response times, ranging from 23 milliseconds for 2 nodes to 190 milliseconds for 20 nodes. As the number of nodes increased from 2 to 20, our framework provided better response times than the other schemes, showcasing its ability to handle increased computational loads effectively. This performance is attributed to the optimization of the computational protocol and the limited use of additive homomorphic encryption, efficiently integrated with the smart contract.

The experimental results confirm the viability and effectiveness of our proposed framework. Despite the inherent trade-off between data privacy and computational efficiency, our optimization efforts have successfully balanced these aspects to ensure acceptable performance for practical healthcare applications.

7 Conclusion

The integration of blockchain technology into healthcare data management introduces a transformative framework aimed at safeguarding data accessibility, integrity, and patient privacy. By leveraging advanced cryptographic techniques such as additive homomorphic encryption, secret-sharing, and SMPC, this framework ensures data confidentiality throughout collaborative analysis among healthcare providers and researchers. The decentralized ledger structure of blockchain, combined with sophisticated encryption methods, ensures secure computation on encrypted data, preserving confidentiality throughout processing. The meticulous orchestration of an eight-step process, from requester verification to result dispatch, guarantees controlled access to sensitive health records. This framework ensures secure computation on encrypted data while providing controlled access for healthcare providers and researchers. Through the utilization of additive homomorphic encryption and SMPC protocols, this innovative approach fosters a trustworthy environment for collaborative research while adhering to regulatory standards and addressing patient privacy concerns, marking a significant stride in the privacy-conscious evolution of healthcare data management. Future research could explore the integration of fully homomorphic encryption to enable more complex computations while preserving data privacy, though this will require addressing the computational overhead associated with such techniques. The convergence of blockchain and artificial intelligence (AI) presents opportunities for predictive healthcare analytics through federated learning, allowing AI models to train on decentralized data without compromising patient privacy. Additionally, advancing interoperability by developing standardized protocols for data exchange between distinct blockchain platforms will be essential to fully realize the potential of blockchain technology in healthcare.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Ahmed Tawfik, the first draft of the manuscript was written by Ahmed Tawfik, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). No funding was received to assist with the preparation of this manuscript. The authors have no relevant financial or non-financial interests to disclose.

Data availibility Datasets generated during and /or analyzed during the current study are available from the first author upon reasonable request.

Declarations

Conflict of interest The Author declares that there is no Conflict of interest.

Ethical approval Not applicable.

Consent to publish Not applicable.

References

1. Mayer, A.H., André, C., da Costa, R., da Rosa, R.: Electronic health records in a blockchain: a systematic review. Health Inf. J. **26**(2), 1273–1288 (2020). https://doi.org/10.1177/1460458219866350. **(PMID: 31566472)**
2. Batubara, F.R., Ubacht, J., Janssen, M.: Unraveling transparency and accountability in blockchain. In Proceedings of the 20th Annual International Conference on Digital Government Research, pp 204–213 (2019)
3. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S.W., Yellick, J.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, New York, NY, USA, (2018). Association for Computing Machinery ISBN 9781450355841. https://doi.org/10.1145/3190508.3190538
4. HIPAA Compliance Assistance. Summary of the hipaa privacy rule. Office for Civil Rights (2003)
5. Kuo, T.-T., Bath, T., Ma, S., Pattengale, N., Yang, M., Cao, Y., Hudson, C.M., Kim, J., Post, K., Xiong, L., Ohno-Machado, L.: Benchmarking blockchain-based gene-drug interaction data sharing methods: A case study from the idash 2019 secure genome analysis competition blockchain track. Int. J. Med. Inf. **154**, 104559 (2021) https://doi.org/10.1016/j.ijmedinf.2021.104559. https://www.sciencedirect.com/science/article/pii/S1386505621001854
6. Cramer, R., Damgård, I.B., Nielsen, J.B.: Secure multiparty computation and secret sharing. Secure Multiparty Computation and Secret Sharing, pp 1–373 (2015). https://doi.org/10.1017/CBO9781107337756
7. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Foundations Secure Comput. **4**(11), 169–180 (1978)
8. Zheng, Q., Li, Y., Chen, P., Dong, X.: An innovative IPFS-based storage model for blockchain. Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2018, pp 704–708 (2019). https://doi.org/10.1109/WI.2018.000-8
9. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979). https://doi.org/10.1145/359168.359176
10. Yang, C.-N., Li, P., Cheng, H.-H., Kuo, H.-C., Lu, M.-C., Xiong, L.: A security model of multihospital FHIR database authorization based on secret sharing and blockchain. IEEE IoT J (2023). https://doi.org/10.1109/JIOT.2023.3328989
11. Yao, A.C.: Protocols for secure computations. In 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982), pp 160–164 (1982). https://doi.org/10.1109/SFCS.1982.38
12. Yao, A.C.-C.: How to generate and exchange secrets. In 27th Annual Symposium on Foundations of Computer Science (SFCS 1986), pp 162–167 (1986). https://doi.org/10.1109/SFCS.1986.25
13. Micali, S., Goldreich, O., Wigderson, A.: How to play any mental game. In Proceedings of the Nineteenth ACM Symposium on Theory of Computing, STOC, pp 218–229. ACM, New York (1987)
14. Wu, S., Li, J., Duan, F., Lu, Y., Zhang, X., Gan, J.: The survey on the development of secure multi-party computing in the blockchain. In 2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC), pp 1–7 (2021). https://doi.org/10.1109/DSC53577.2021.00008
15. Li, D., Liao, X., Xiang, T., Wu, J., Le, J.: Privacy-preserving self-serviced medical diagnosis scheme based on secure multi-party computation. Comput. Secur. **90**, 101701 (2020). https://doi.org/10.1016/j.cose.2019.101701
16. Zhou, J., Feng, Y., Wang, Z., Guo, D.: Using secure multi-party computation to protect privacy on a permissioned blockchain. Sensors (2021). https://doi.org/10.3390/s21041540
17. Yang, Y., Wei, L., Wu, J., Long, C.: Block-SMPC: a blockchain-based secure multi-party computation for privacy-protected data sharing. In Proceedings of the 2020 The 2nd International Conference on Blockchain Technology, ICBCT'20, pp 46–51. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3390566.3391664
18. Zhang, M., Zhang, Y., Shen, G.: Ppdds: a privacy-preserving disease diagnosis scheme based on the secure Mahalanobis distance evaluation model. IEEE Syst. J. **16**(3), 4552–4562 (2022). https://doi.org/10.1109/JSYST.2021.3093415
19. Vanin, F.N.d.S., Policarpo, L.M., Righi, R.d.R., Heck, S.M., da Silva, V.F., Goldim, J., da Costa, C.A.: A blockchain-based end-to-end data protection model for personal health records sharing: a fully homomorphic encryption approach. Sensors, **23**(1), 2023. https://doi.org/10.3390/s23010014. https://www.mdpi.com/1424-8220/23/1/14
20. Wibawa, F., Catak, F.O., Kuzlu, M., Sarp, S., Cali, U.: Homomorphic encryption and federated learning based privacy-preserving cnn training: Covid-19 detection use-case. In Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference, EICC '22, pp 85–90. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3528580.3532845
21. Ali, A., Al-rimy, B.A.S., Alsubaei, F.S., Almazroi, A.A., Almazroi, A.A.: Healthlock: Blockchain-based privacy preservation using homomorphic encryption in internet of things healthcare applications. Sensors, **23**(15), (2023). https://doi.org/10.3390/s23156762
22. Ghadamyari, M., Samet, S.: Privacy-preserving statistical analysis of health data using paillier homomorphic encryption and permissioned blockchain. In 2019 IEEE International Conference on Big Data (Big Data), pp 5474–5479 (2019). https://doi.org/10.1109/BigData47090.2019.9006231
23. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In Stern, J (ed.), Advances in Cryptology—EUROCRYPT '99, pp 223–238. Springer, Berlin, Heidelberg (1999)
24. Paar, C., Pelzl, J.: The Advanced Encryption Standard (AES), pp. 87–121. Springer, Berlin, Heidelberg (2010)

25. Castro, M., Liskov, B.: Practical byzantine fault tolerance. OSDI, pp 173–186, (1999). cited By 99
26. Alexandridis, A., Al-Sumaidaee, G., Alkhudary, R., Zilic, Z.: Making case for using raft in healthcare through hyperledger fabric. In 2021 IEEE International Conference on Big Data (Big Data), pp 2185–2191. IEEE (2021)
27. HYPERLEDGER. Measuring blockchain performance with hyperledger caliper—hyperledger foundation , Mar 19 (2018)
28. Zhou, J., Feng, Y., Wang, Z., Guo, D.: Using secure multi-party computation to protect privacy on a permissioned blockchain. Sensors, **21**(4), (2021). https://doi.org/10.3390/s21041540
29. Barker, E., Burr, W., Jones, A., Polk, T., Rose, S., Smid, M., Dang, Q., et al.: Recommendation for key management part 3: application-specific key management guidance. NIST Spec. Publ. **800**, 57 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ahmed M. Tawfik holds a B.Sc. in Computer Science and an M.Sc. in Privacy and Security of EHRs from Benha University. He is currently pursuing a Ph.D. in Computer Science and serves as an Assistant Lecturer at the Faculty of Computers and Artificial Intelligence, Benha University. His research interests include privacy preservation in healthcare, security, and blockchain-based access control methods.

Ayman Al-Ahwal earned a B.Sc. in Electronics and Communication Engineering from Zagazig University (Benha Branch) in 1997, an M.Sc. from the same institution in 2003, and a Ph.D. in Electronics (Computer) from Benha University in 2008. His research focuses on computer network security, information security, IoT, VANET routing protocols, and cryptography.

Adly S. Tag Eldien received a B.Sc. in Electronics and Communication Engineering in 1984, an M.Sc. in 1989, and a Ph.D. in Electrical Engineering in 1993 from Zagazig University (Benha Branch). He is currently a professor in the Department of Electrical Engineering at Benha University. His research interests include communication networks, robotics, network security, privacy, and cryptography.

Hala H. Zayed earned a B.Sc. in Electronics and Communication Engineering from Zagazig University (Benha Branch) in 1985, an M.Sc. in 1989, and a Ph.D. in Electrical Engineering in 1995. She became an Associate Professor in Computer Engineering in 2005 and was promoted to Professor of Computer Science in 2012. Previously, she served as the Dean of the Faculty of Computers and Artificial Intelligence at Benha University. She is currently a Professor at the Faculty of Engineering, Egypt University of Informatics (EUI), Cairo, Egypt. Her research interests include artificial intelligence, neural networks, computer vision, feature extraction, pattern recognition, security, and healthcare.