Authentication and Access Control-Based Data Security in EHR: A Blockchain-Based System for Mobile Cloud Computing

B. Prema Sindhuri
Department of CSE, K L University, India
premasindhuri13@gmail.com (corresponding author)

Kameswara M. Rao
Department of ECSE, K L University, India
kamesh.manchiraju@kluniversity.in

Received: 15 November 2025 | Revised: 21 December 2025 and 6 January 2026 | Accepted: 9 January 2026

 DOI: https://doi.org/10.48084/etasr.16290

ABSTRACT

The storage of Electronic Health Records (EHRs) in mobile cloud platforms has evolved over the past couple of decades as mobile devices and cloud computing have been linked to facilitate the exchange of medical data between patients and medical professionals. Moreover, this modern approach offers medical institutions enhanced flexibility, reduced operational costs, and improved accessibility to EHRs. This innovative method raises concerns about network security and data privacy in e-health systems. Distributing EHRs to mobile users while maintaining confidentiality standards in the mobile cloud is difficult. This manuscript presents an integrated blockchain-based security framework for EHR sharing in Mobile Cloud Computing (MCC) environments. The proposed framework operates through four phases: registration, authentication, contract agreement, and data uploading and encryption. Blockchain technology is employed as a decentralized trust layer to provide transparency, immutability, and auditable access control, whereas smart contracts are used to enforce authorization policies among healthcare participants. Additionally, this work uses a dependable access control strategy in line with smart contracts to ensure secure transmission of EHRs between patients and medical professionals. The suggested Integrated Lightweight Key Management Mechanism (ILWKM) ensures authentication with secure transactions by generating a symmetric encryption key and a session key. The session key is encrypted using a modified cubic map along with data upload parameters. The new data encryption standard, Improved Elliptic Curve Cryptography (IECC) mechanism, is suggested to encrypt the data with a high level of security during the data uploading and encryption phase. The experimental results demonstrate that the proposed framework achieves improved security and computational efficiency compared to conventional approaches, indicating its suitability for secure EHR sharing in MCC environments.

Keywords-Electronic Health Record (EHR); blockchain; authentication; Integrated Lightweight Key Management Mechanism (ILWKM); Improved Elliptic Curve Cryptography (IECC)

I. INTRODUCTION

The application of blockchain technology to e-health services has gained significant traction recently. Due to its decentralized and tamper-resistant characteristics, blockchain has demonstrated significant potential in secure Electronic Health Records (EHRs) sharing and access control across distributed healthcare institutions. The integration of Mobile Cloud Computing (MCC) and the Internet of Medical Things (IoMT) has further transformed healthcare delivery by enabling continuous health monitoring and remote medical services [1, 2]. Therefore, the implementation of blockchain technology has the potential to introduce novel solutions to facilitate healthcare delivery and, consequently, transform the healthcare industry [3]. The introduction of novel technologies such as MCC and IoMT has resulted in substantial modifications in e-health operations in the healthcare sector. Patients can currently gather health data at home using mobile devices (including wearable sensors and smartphones) and distribute them to cloud settings, where clinicians have immediate access to examine health records and provide urgent medical attention [4, 5]. Moreover, this smart e-health service facilitates medical experts to monitor patients and provide continuous treatment at home, which not only improves healthcare delivery but also reduces costs for patients. In addition, the accessibility of entire EHRs in the cloud assists medical professionals in tracking patient health and providing appropriate medical services during evaluation and treatment [6, 7].

Despite these benefits, the growing prevalence of EHR storage on clouds raises security problems that slow the adoption of e-health applications in cloud environments. Secure EHR sharing among medical professionals and patients in mobile cloud platforms is one of these key challenges [8, 9]. Unregistered individuals may acquire unauthorized access to EHRs without patients' consent, compromising privacy, security, and data integrity of cloud e-health services. Furthermore, consumers may struggle to monitor and manage their health information shared among healthcare providers on clouds [10, 11]. As a result, reliable security measures for mobile cloud EHR sharing systems are required [12].

Conventional methods for EHR sharing access control presume that cloud servers are completely trusted by the data owners, and that the servers can execute all access management and authorization rights over the usage of data. This assumption, however, no longer holds in mobile clouds because the cloud server has emerged as honest-but-curious [13, 14]. The cloud server will genuinely execute data requests, but in the meantime, it may gather private data from users without their knowledge, resulting in major information leakage concerns and network security vulnerabilities. Additionally, traditional access control solutions focus primarily on a specified point of access, which might create a central point of failure for e-health networks [15, 16].

In this context, blockchain-related access control delivers several innovative security improvements for e-health that outperform traditional access control systems. Initially, the blockchain creates irreversible transaction ledgers for collaborating on data systems [17, 18]. Furthermore, access control utilizing blockchain may accomplish transparency, while also successfully addressing the issue of data leakage resulting from inquisitive servers. Third, this approach employs blockchain-based smart contracts to achieve user verification and authentication. Eventually, blockchain combined with smart contract software negates the need for central servers to assure transaction parties' fairness [19]. Because smart contracts are public on the blockchain, all linked entities in the blockchain network have a copy of them, granting equal power over the regulation of each contract's operations [20].

Unlike Hyperledger Fabric–based EHR systems that focus on enterprise-level deployment and permissioned blockchain infrastructure [21], this work emphasizes lightweight cryptographic security and authentication mechanisms suitable for MCC environments. The proposed framework is complementary to Fabric-based systems and can be integrated as a cryptographic enhancement layer within permissioned healthcare blockchains. Motivated by these challenges, this paper proposes a blockchain-enabled EHR security framework that integrates smart contract-based access control with lightweight key management and Improved Elliptic Curve Cryptography (IECC). The primary objective of this work is to enhance authentication efficiency and data confidentiality while maintaining compatibility with MCC environments.

A. Related Work

In 2021, authors in [4] developed the IKGSR approach, utilizing a hybrid cryptography strategy that encrypts data and employs a Blowfish algorithm for the encryption key. Additionally, stenography-related access control is used to retrieve the encrypted material, distributing the key in accordance with substring indexing and keyword search techniques. The performance and security of the proposed method were compared with those of conventional methods. Although the model demonstrates improved security and retrieval performance, it does not address blockchain-based access enforcement or mobile cloud scalability constraints.

In 2023, authors in [3] developed the MA-RABE technique to exchange encrypted attribute-based schemes based on multiple authorities in blockchain. By transmitting and sharing secret data, the authority communities carried out key generation, sharing of user attributes, and user access control. Furthermore, a shared single-way anonymous key contract was created, which helped secure data against unauthenticated users. While maintaining minimal user revocation overhead, the method achieved robust security.

In 2022, authors in [6] proposed the CP2EH mechanism to address both attribute security for doctors and unauthorized access to patient details. Zero-Knowledge Proof (ZKP) and Oblivious Transfer (OT) protocols were combined, with the OT protocol protecting doctor's secret keys and privacy features, and ZKP safeguarding patient personal information from untrusted medical professionals. As a result, the model reduced both communication and computing costs.

In 2021, authors in [1] developed the SAS mechanism for privacy preservation. The model gathered information from each EHR system without data loss, ensuring privacy and confidentiality. The data were gathered using various data mining approaches to verify the central data mining server. Experimental and theoretical assessments demonstrated improvements in communication and computational costs.

In 2016, authors in [5] proposed MedRec, a blockchain-based system for managing access permissions to EHRs. The scheme employed smart contracts to control authentication and data access among healthcare participants. Although MedRec improves the transparency and integrity of medical data sharing, it mainly concentrates on permission management rather than on lightweight cryptographic efficiency for mobile cloud environments.

In 2020, authors in [8] proposed xDBAuth, an authorized delegation and access control system for Internet of Things (IoT) based on decentralized blockchain. The work presented an organizational framework for both global and local smart contracts for authorized delegation and access control of internal and external IoT devices. Furthermore, the recommended structure protected the privacy of an external user through enabling them to acquire authorization in their corresponding parent IoT networks.

In 2023, authors in [7] proposed the IPRS mechanism to preserve Patients' Medical Health Records (PMHRs) information while reducing computational difficulties. The mechanism was implemented in Solidity to validate functions. Security against Man-in-the-Middle (MITM) attacks was verified through assessment.

In 2019, authors in [2] developed an Ethereum blockchain system to secure the distribution of EHRs in MCC environments according to the e-health model. The system secured original health details from potential attacks and achieved resilience in data transmission on mobile clouds. The suggested approach evaluated system security and demonstrated performance improvements, including lower network latency, enhanced data privacy, and implemented lightweight access control. Table I summarizes the key features and challenges of the discussed conventional techniques.

TABLE I. FEATURES AND CHALLENGES OF CONVENTIONAL TECHNIQUES RELEVANT TO DATA SECURITY IN EHR

Ref.	Techniques	Features	Challenges
[4]	IKGSR	Secure data extraction	Requires blockchain-based sharing and secure storage for healthcare information
[3]	MA-RABE	Minimized user revocation and robust security	Requires trusted outsourced decryption to secure CSPs against dishonest protocols
[6]	CP2EH	Minimal communication and computation cost	Does not include value-added services
[1]	SAS	Collusion resistance against intermediate data mining	Needs dynamic joining/leaving of EHR schemes
[5]	MedRec	Blockchain-based access control for EHRs	Limited focus on lightweight encryption and MCC efficiency
[8]	xDBAuth	Low computational overhead	Needs formal validation and modeling
[7]	IPRS	Reduced computational complexity	Needs consideration of collusion attacks in cloud
[2]	Ethereum blockchain	Low network latency and high security	Requires effective control of EHR access in mobile cloud settings

Although existing studies address secure EHR sharing using blockchain and cryptographic techniques, several limitations remain. Many approaches focus exclusively on access control without integrating lightweight authentication suitable for MCC environments. Others rely on computationally intensive encryption schemes, increasing latency. Limited attention has also been given to combining authentication, access control, and encryption within a unified blockchain-enabled workflow.

The contributions of this work are summarized as follows:

- An integrated blockchain-enabled EHR security framework for MCC is presented, combining authentication, access control, and encrypted storage into a unified workflow.
- An Integrated Lightweight Key Management Mechanism (ILWKM) is employed to reduce authentication overhead by combining pre-shared symmetric keys with session key generation.
- An improved ECC-based encryption process (IECC) is implemented to enhance data protection during EHR storage and transmission, focusing on performance improvements rather than redefining core ECC principles.

II. METHODOLOGY

A. Architecture of the Proposed Blockchain-Based EHR System

The proposed framework adopts a conceptual permissioned blockchain model to manage EHRs access transactions. The focus of this work is on access control logic and cryptographic security rather than blockchain protocol optimization. Therefore, parameters such as block size, consensus throughput, and mining latency are not fixed to a specific blockchain implementation (e.g., Ethereum or Hyperledger) and are abstracted at the functional level. The proposed system comprises four main entities: patients, healthcare providers, cloud servers, and a blockchain network. Patients generate and own EHR data, healthcare providers request access to medical records, and cloud servers store encrypted EHR data. The blockchain network maintains access policies and transaction records through smart contracts. The security objectives of the proposed system include:

- Confidentiality of EHR data during storage and transmission.
- Secure user authentication.
- Fine-grained access control.
- Resistance to common cryptographic attacks.

The system assumes a semi-trusted cloud environment and the presence of adversaries capable of eavesdropping, replaying messages, or attempting unauthorized access. In Figure 1, patients' details are categorized by patient ID and area ID, representing their current residence.

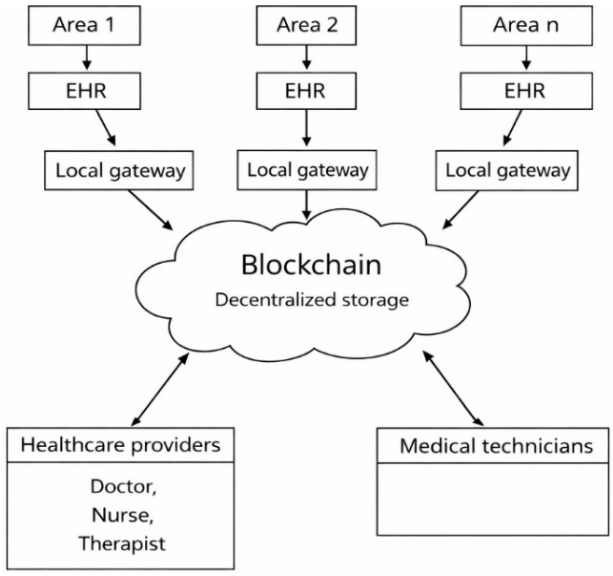


Fig. 1. Architectural framework of the proposed blockchain-based EHR system.

Let P^{id} and A^{id} denote patient ID and area ID, respectively. The wearable sensor network can be considered as private and administrated by its local user. The patients'

details are gathered from wearable sensors like smartwatches through a mobile application incorporated in patients' smartphones. As stated before, patient records stored on the blockchain correspond to their addresses, represented as $Adr = \{A^{id}, P^{id}\}$, whereas the bulk of medical records are stored in decentralized cloud storage due to blockchain storage limitations. The cloud EHR manager is responsible for administering these medical records. The participating entities must know the patient address to retrieve a specific health record from the cloud. The proposed mobile cloud blockchain system is depicted in Figure 2.

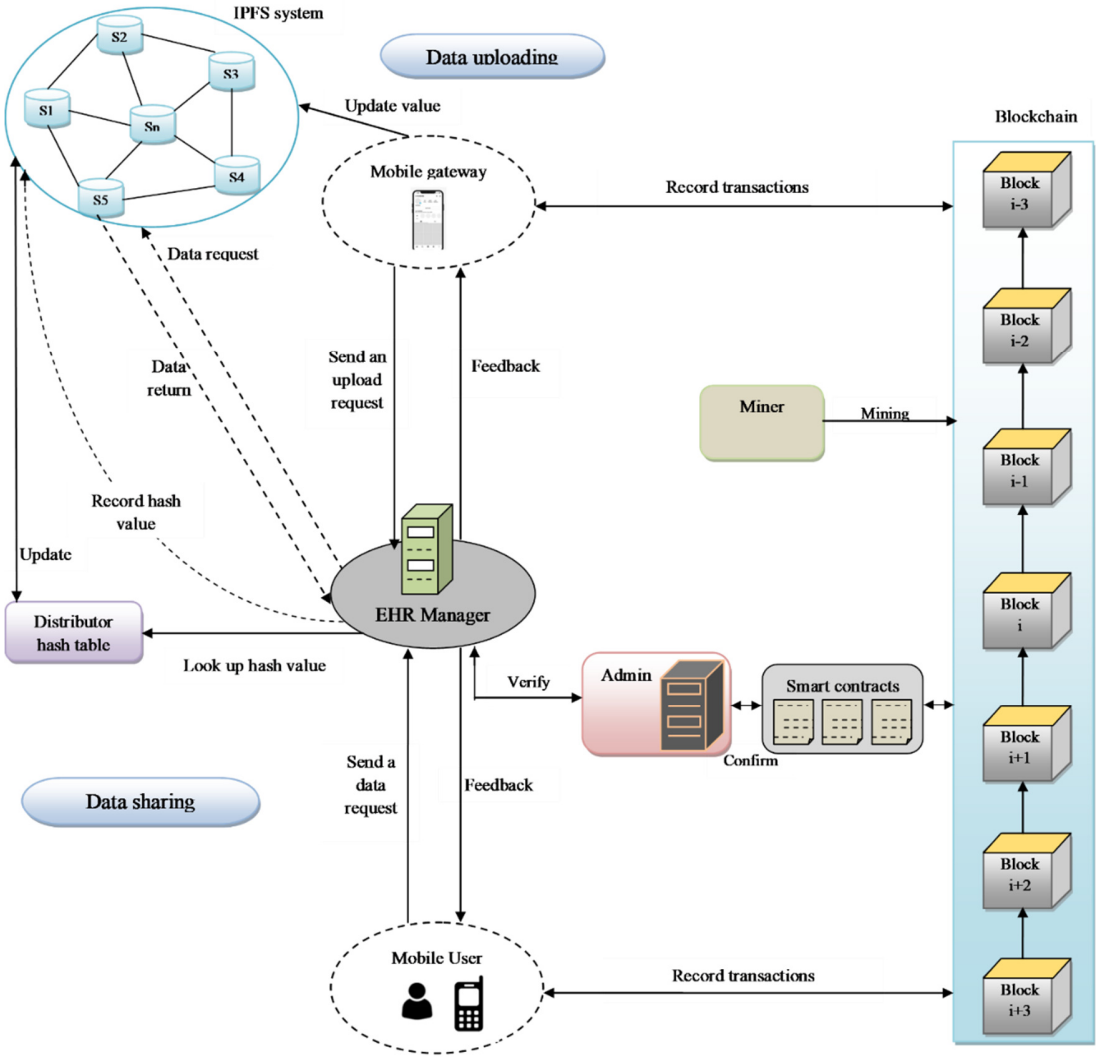


Fig. 2. Proposed model of the mobile cloud blockchain system.

Consider that each healthcare provider is assigned an ID, represented as Hp^{id}, which they use to access patient records on the cloud via mobile devices. For any medical assessment, the health provider obtains the patient's medical history from cloud storage. Clinician access patients' EHRs to investigate medical records and provide appropriate healthcare services. The major entities of the cloud blockchain network are as follows:

- EHR manager: The EHR manager plays a vital role in regulating data access and storage transactions within the blockchain network. User transactions, including mobile user data access and mobile gateway storage processes, are regulated by EHR using smart contracts and authorization policies.
- Smart contracts: All access control operations are defined and enforced by smart contracts. Users communicate with smart contracts via the contract address and the Application Binary Interface (ABI). Moreover, smart contracts identify and grant access permissions, and validate requests from medical users by triggering messages or transactions.
- Admin: The admin manages all operations and transactions on cloud, including modifying, adding, or cancelling access permissions. Moreover, the admin is the only entity capable

of updating or modifying the rules defined in smart contracts.

Decentralized storage: Because storing large amounts of data directly on the blockchain is infeasible, the system employs InterPlanetary File System (IPFS) as a decentralized, end-to-end file storage framework. IPFS is implemented on top of Kademlia DHT and BitTorrent protocol. Also, the users can verify and access files in IPFS by relying on cryptographic hashes.

B. Structure of Data Block

As depicted in Figure 3, the data block structure is implemented with two major elements, including the block header and transaction records. The description of these two elements is elaborated as follows.

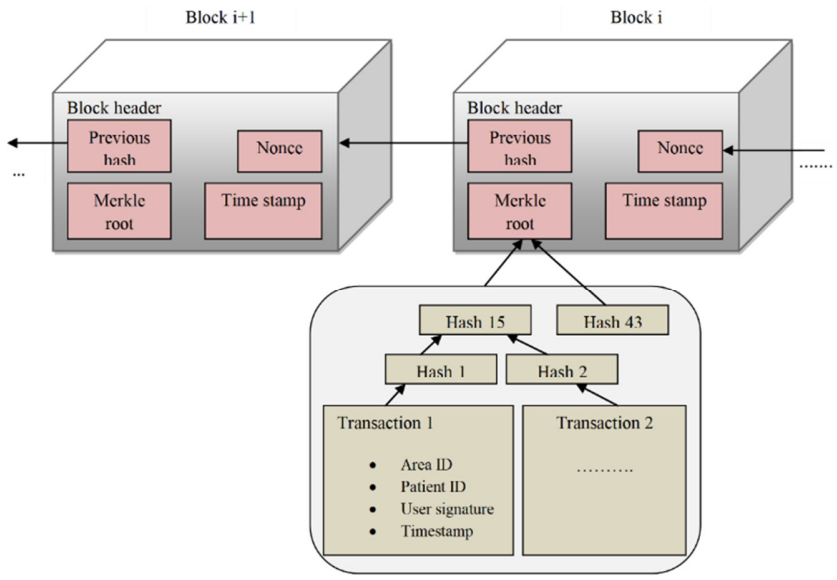


Fig. 3. Data block framework of the proposed model.

1) Transaction Records

According to the Merkle tree infrastructure, the transactions in each block are systematized, in which the leaf nodes represent mobile users' data access transactions. The transaction process supports data requests in which the mobile user provides the details of the patient with $\left(A^{id}, P^{id}\right)$, and these requests are signed and verified using the user's private key at a specific time (timestamp). Thus, trust is established between the data block and the user through this digital signature [2].

2) Block Header

The following metadata are included in the block header to validate the data block:

- Merkle root: The Merkle root is a framework used to accumulate a set of transactions in a block.
- Hash: The block's SHA-256 hash is employed in this work, and the value of the hash is defined as in (1). Here, *Hs* signifies the hash function, and *Tr* signifies transaction data:

$$Hs_{12} = Hs\left(Hs_1 + Hs_2\right) = Hs\left[\left(Tr_1.Hs\right) + \left(Tr_2.Hs\right)\right] \quad (1)$$

- Nonce: To create a hash value below the target difficulty, the nonce generates a number through the proof of work operation performed by miner nodes.
- Previous hash: The block's previous hash is employed to validate the block.
- Timestamp: The generated block's time is termed as timestamp, and it is the timestamp of the last transaction in the block.

C. Registration Phase

During the registration phase, patients and healthcare providers submit their identity credentials to the system. Unique identifiers and cryptographic parameters are generated and securely stored. Registration details are recorded on the blockchain using smart contracts to ensure transparency and immutability. Through strict user constraints, smart contracts provide the EHR manager with management authority [2].

D. Contract Agreement Phase

Smart contracts define access control policies and permissions for EHR sharing. The ABI, along with the contract address, is used to interface with smart contracts. Smart contracts can recognize, verify, and grant access permissions to medical users by triggering transactions. Once deployed, the contracts automatically enforce access rules without reliance on centralized authorities. Therefore, they are regarded as critical software in the proposed e-health infrastructure [2].

1) Smart Contract Design

To track transaction activities on the blockchain network and establish a data-sharing agreement, a smart contract is

administered by the administrator. Let the user role, the user public key, and the patient's blockchain address be denoted by U^{Role}, U^{PK}, and Adr, respectively. The following five functions are mainly involved in the contract:

- $AddUser\left(U^{PK}, U^{Role}\right)$: A new user is added to the main contract through this function, which is executed by the administrator. According to the request, the user is identified by the public key and appended to the contract with the corresponding role.
- $RetrieveEHRs\left(U^{PK}, Adr\right)$: Medical records are extracted from cloud storage through this function, which is executed by the EHR manager. The patients' details can be retrieved by any network participant by providing the ID $\left(A^{id}, P^{id}\right)$ to the smart contract. Then, the smart contract validates the request and transmits a message to the EHR manager to retrieve and return the required data to the requester.
- $DeleteUser\left(U^{PK}, U^{Role}\right)$: A user can be removed from the network according to the corresponding public key, and this function is executed by the administrator. In addition, all related patient data can be removed from cloud storage.
- $Penalty\left(U^{PK}, action\right)$: When a request to the EHR system is detected as unlawful, the EHR manager notifies the smart contract to impose a penalty on the requester, such as issuing a warning message to an unauthenticated mobile entity.
- $PolicyList\left(U^{PK}\right)$: Executed by the administrator, this function defines the policy agreement for medical services as an end-to-end agreement between medical staff and the patient, for example, allowing a specific clinician to access the records of a particular patient.

2) Data Sharing Process

In the main cloud, data sharing with access control is implemented to allow authorized mobile users to access data. The procedure consists of three phases: creation of request information, access validation and data extraction, and adding the data-sharing transaction to the blockchain [2]. The data uploading and sharing procedure using IECC is depicted in Figure 4.

- Creation of request information: To access the data, a mobile user generates a user transaction Tr, which represents a data request for cloud-stored records. The user creates a blockchain account comprising a private key and a public key for identification and transaction signing. The patient's address is included in the request as metadata. The transaction data Tr are then transmitted to the EHR manager for access verification on the cloud platform.
- Access validation and data extraction: This phase is executed by the EHR manager, who forwards the user transaction to the smart contract for validation and auditing. Based on the public keys listed in policy list, the smart contract authenticates the request and returns the result to the EHR manager. If the user public key is included in the policy list, access permission is granted and the requester obtains the required information. Otherwise, the smart contract imposes a penalty and blocks further requests from that user. The request index contains the patient's storage address, enabling the EHR manager to locate the requested data in the storage node in the decentralized cloud. The EHR manager verifies the hash value of the data file and retrieves the encrypted file from the IPFS storage system, which is then returned to the requester.
- Adding the data-sharing transaction to blockchain: The validated transaction is clustered into a block, added to the transaction pool, and processed by the mining network. All confirmed transactions are recorded and appended to the blockchain for distribution to all mobile users. Finally, end users update their transaction data through the blockchain client integrated into their mobile devices.

E. Authentication Phase: Proposed Integrated Lightweight Key Management Mechanism Scheme

In this phase, authentication is performed between the patient and the administrator. For authentication and auditing purposes, the user transaction is acquired by the EHR manager. The administrator subsequently transmits a message to the smart contract, urging it to confirm the request. In accordance with the policy list, the smart contract validates the request and returns a message to the EHR administrator that includes the user's public key. When the user's public key is identified in the policy list, the requester's access permission is authorized, allowing them to view the requested data. On the contrary, if the key is not recognized, the smart contract penalizes the requester and negates future requests from this user.

It is also necessary to minimize transaction delay and power consumption for this authenticated transaction. The ILWKM is employed to authenticate users efficiently in MCC environments. ILWKM generates symmetric encryption keys and session keys using a pre-shared key strategy combined with a modified cubic map. This approach reduces computational and communication overhead while maintaining secure session establishment [22]. ILWKM does not alter blockchain consensus or block formation; instead, it ensures that only verified users can initiate valid blockchain transactions. ILWKM is evaluated using performance metrics such as latency, key sensitivity, and attack resistance rather than being proposed as a formally new cryptographic primitive. The mathematical notations used in the ILWKM process are summarized in Table II.

TABLE II. MATHEMATICAL NOTATIONS OF ILWKM PROCESS

Notation	Description
Ky^{pool}	Subset of pool keys
Ky	Symmetric encryption key
Ky_{Enc}	Encryption key
g	Control parameter
Sn_n^{Ky}	Session key for a particular sensor device
$Mstr^{Ky}$	Master session key for encryption

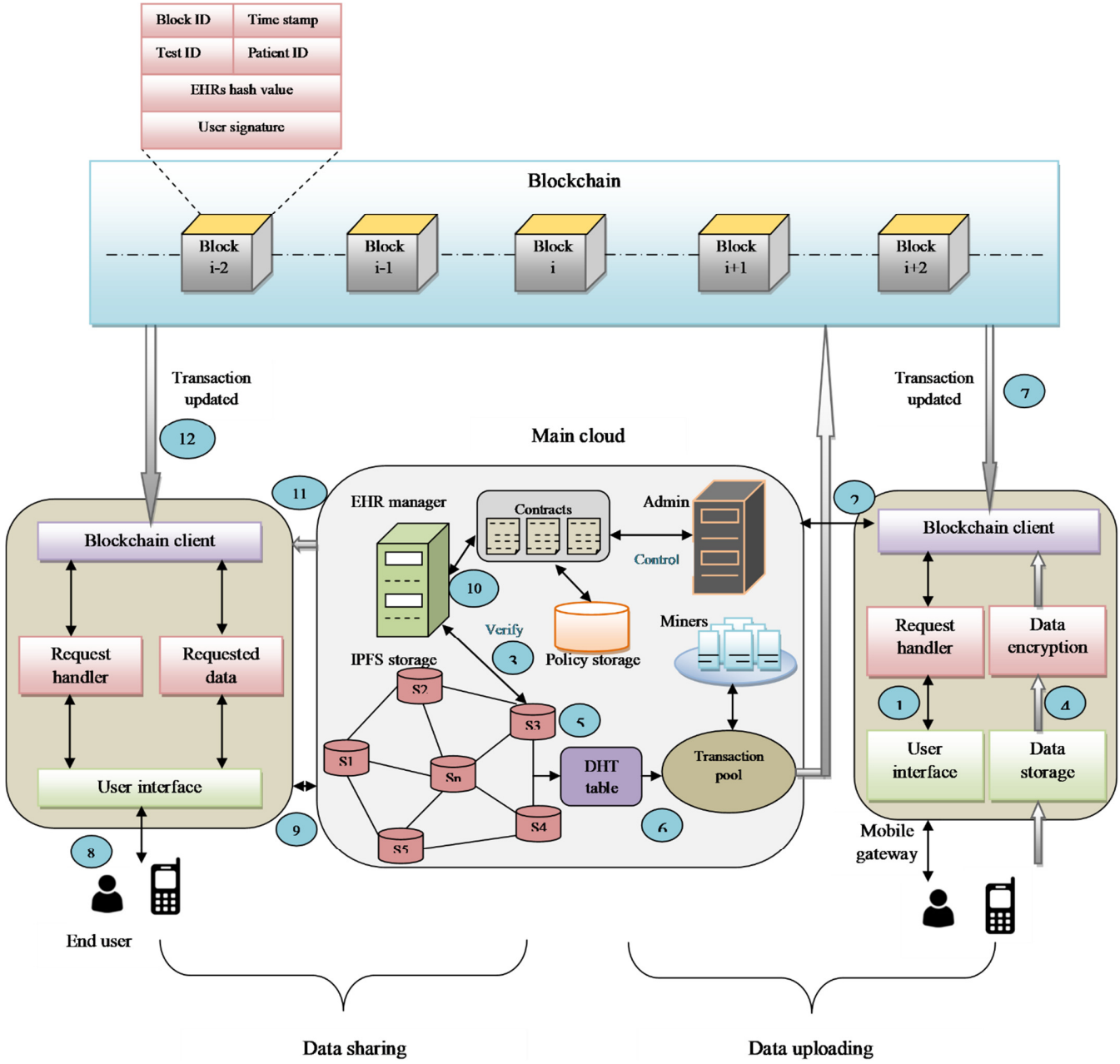


Fig. 4. Data uploading and sharing procedure using IECC.

Using a pseudorandom function, a large pool of keys is generated. Assume that a subset of these random pool keys, denoted as Ky^{pool}, is assigned to each sensor device, since storing and transmitting keys in wireless devices without protection is insecure. The symmetric encryption key is generated by selecting each key Ky from Ky^{pool} and XORing it with a random value to produce the encryption key Ky_{Enc}. In the proposed work, the data are encrypted using the modified ECC-based method.

The session key is generated after data encryption. Each session key is derived using the modified cubic map and can be expressed as in (2). Here, g is the control parameter taking values between 0 and 4, and Sn_n^{Ky} is the session key value between 0 and 1:

$$Sn_{n+1}^{Ky} = \left[g.e^{Sn_n^{Ky}} \left(1 - e^{Sn_n^{Ky2}}\right) \right] \bmod 1 \quad (2)$$

The generated session key is then distributed to all sensor devices to secure their encrypted symmetric key. Also, the session key Sn_{n+1}^{Ky} is used to generate the master key $Mstr^{Ky}$, which encrypts the encryption parameters. Moreover, this master key is used to decrypt the encrypted data.

F. Data Uploading and Encryption Phase

Data access transactions are recorded in the blockchain ledger, whereas the encrypted EHR data are stored off-chain in decentralized storage. Before being appended to the blockchain, transactions are grouped into data blocks and placed in the transaction pool for approval by miners. The uploading transaction is processed through the mobile gateway to track relevant factors. The uploaded data are encrypted using the proposed IECC approach [23].

1) Encryption Phase: IECC Approach

To encrypt and decrypt the data, the ECC technique is used to generate both a private key and a public key. ECC provides high security with low processing power, making it suitable for mobile platforms. However, ECC implementation is prone to larger error rates, which can complicate system construction and lower security. The proposed IECC mechanism does not alter the underlying elliptic curve structure or hardness assumptions of ECC. Instead, it introduces an additional secret key component into the encryption and decryption workflow, combined with standard ECC operations, to increase resistance against key compromise and brute-force attacks while maintaining computational efficiency.

The identification Hp^{id} is authorized whenever the data need to be accessed by medical practitioners. The proposed approach is mapped onto the curve using specified fundamental points through a prime number function. The mathematical formulation of ECC is defined as in (3), where p and q are integers.

$$s^2 = t^3 + pt + q \tag{3}$$

Assume that the three types of keys, public key α^{Ky}, private key β^{Ky}, and secret key γ^{Ky} are used. Initially, the public key α^{Ky} is created on the server and used to encrypt the data. Then, the private key β^{Ky} is generated at the server, and the secret key γ^{Ky} is derived when α^{Ky} and β^{Ky} are initialized; the point on the curve can be represented as Poc. Traditionally, the secret key γ^{Ky} is added during encryption and subtracted during decryption. The public key is generated using the private key and point on the curve as defined in (4):

$$\alpha^{Ky} = \beta^{Ky} \times Poc \tag{4}$$

Similarly, the secret key is computed as the sum of the public key, private key, and point on the curve, as defined in (5):

$$\gamma^{Ky} = \alpha^{Ky} + \beta^{Ky} + Poc \tag{5}$$

It is important to note that the additional secret key, γ^{Ky}, is combined with the standard ECC public and private keys only at the encryption layer. The elliptic curve parameters and point multiplication operations remain unchanged from conventional ECC. The mathematical notations used in the IECC process are summarized in Table III.

TABLE III. MATHEMATICAL NOTATIONS OF IECC PROCESS

Notation	Description
α^{Ky}	Public key
β^{Ky}	Private key
γ^{Ky}	Secret key
Poc	Point on curve
CT_1	Traditional ciphertext at point 1
CT_2	Traditional ciphertext at point 2
P^{rec}	Patient records
r	Random number
ICT_1	Improved ciphertext at point 1
ICT_2	Improved ciphertext at point 2

a) Encryption

Let us assume that the patients' records (data) are P^{rec}, which are mapped to the affine Poc [23]. The data are then encrypted into two ciphertexts (CT_1 and CT_2) using traditional ECC calculations, as defined in (6) and (7), where r is a random number and P^{rec} specifies the original patients' records:

$$CT_1 = \left(r \times Poc\right) + \gamma^{Ky} \tag{6}$$

$$CT_2 = \left(P^{rec} + \left(r \times \alpha^{Ky}\right)\right) + \gamma^{Ky} \tag{7}$$

These equations are improved for higher security in the encrypted data, resulting in (8) and (9):

$$ICT_1 = r\left(Poc + \gamma^{Ky}\right) \tag{8}$$

$$ICT_2 = P^{rec} + \left[r \times \alpha^{Ky} + r\left(\beta^{Ky}\right) + \gamma^{Ky}\right] + ICT_1 \tag{9}$$

The encrypted data are then securely uploaded to the IPFS system.

b) Decryption

On the medical practitioner side, the encrypted ciphertext is downloaded securely using the same IECC approach. The traditional decryption function is defined in (10):

$$P^{rec} = \left[\left(CT_2 - \beta^{Ky}\right) * CT_1 - \gamma^{Ky}\right] \tag{10}$$

The improved decryption function is defined in (11), ensuring secure and efficient recovery of the data:

$$ICT_2 - \left[r\left(\alpha^{Ky} + \beta^{Ky}\right)\right] - ICT_1 = P^{rec} \tag{11}$$

Thus, the proposed IECC mechanism decrypts the data securely while maintaining good speed and low computational cost.

III. ANALYSIS OF IECC PERFORMANCE FOR DATA SECURITY IN EHR

A. Simulation Setup

The proposed framework is evaluated using simulation-based experiments implemented in Python 3.7. The blockchain and smart contract components were modeled at a functional level rather than deployed on a live blockchain network. The processor used was an 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40 Ghz (2.42 Ghz), and the installed RAM size was 16.0 GB (15.7 GB usable).

To evaluate the performance of IECC, it was compared with conventional encryption methods, including AES-RSA [7], IKGSR [4], RSA, Fernet, Elgammal and ECC. The evaluation considered metrics such as key sensitivity, throughput, and encryption time. Additionally, the system was analyzed under different attacks, including Chosen Ciphertext Attack (CCA), Chosen Plaintext Attack (CPA), Fault Injection Attack (FIA), Known Plaintext Attack (KPA), and Side-Channel Attack (SCA).

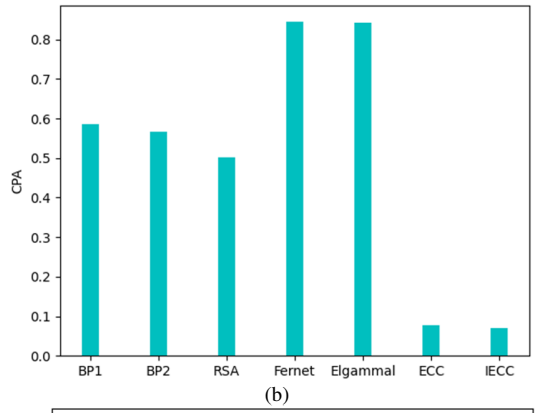
B. Cleveland Dataset (Heart Disease Dataset)

The experimental evaluation employs the UCI Heart Disease dataset [24], which contains 303 instances and 14 commonly used clinical attributes. It is important to note that this dataset does not represent complete clinical semantics or real-time EHR workflows. Instead, it is employed solely to generate representative numerical inputs for cryptographic operations, key management evaluation, and performance analysis. Consequently, the experimental results reflect security and computational behavior rather than full semantic-level healthcare processes. This approach also enables consistent comparison of computational overhead without relying on sensitive real-world EHR data**.**

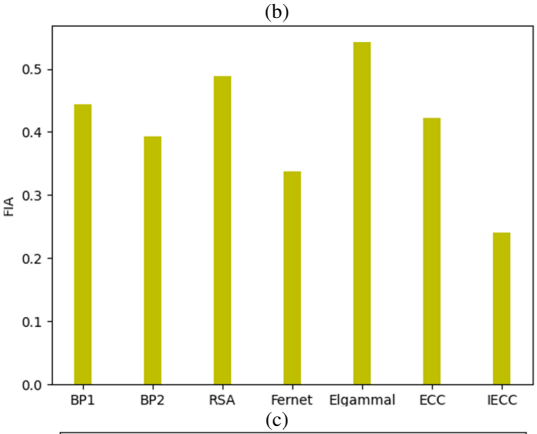
C. Attack Analysis on IECC and Conventional Strategies for Data Security in EHR

The attack analysis of IECC is compared with AES-RSA, IKGSR, RSA, Fernet, Elgammal, and ECC in terms of FIA, CCA, KPA, CPA, and SCA for data security in EHR, as shown in Figure 5. To provide high data security in the EHR system, the model should achieve minimal attack rates. In this regard, IECC consistently demonstrated superior outcomes with lower attack ratings.

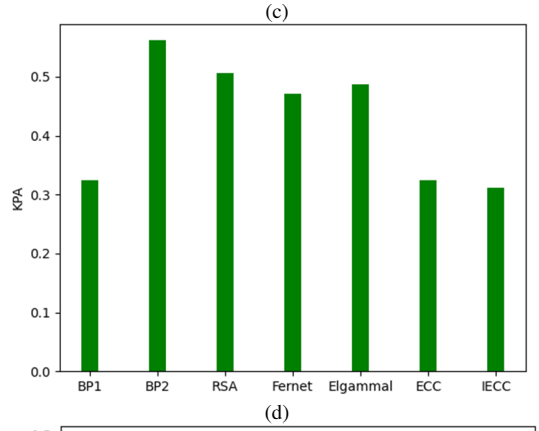
(a)



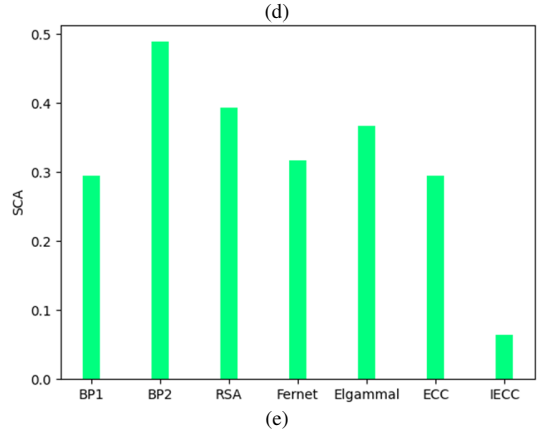
(b)



(c)



(d)



(e)

Fig. 5. Attack evaluation on IECC and conventional schemes for data security in EHR: (a) CCA, (b) CPA, (c) FIA, (d) KPA, and (e) SCA.

CCA is a cryptanalysis method in which the attacker can obtain information by collecting decryptions of selected ciphertexts. According to Figure 5(a), the CCA attack rate for IECC is 0.2315, whereas conventional methods yield higher rates: AES-RSA = 0.4872, IKGSR = 0.4791, RSA = 0.4563, Fernet = 0.3959, Elgammal = 0.3826, and ECC = 0.3713. CPA is a cryptanalysis attack in which the attacker gains access to ciphertexts corresponding to chosen plaintexts, aiming to compromise the encryption scheme. Lower CPA attack rates indicate stronger security. In this criterion, IECC achieved minimal CPA attack values compared with AES-RSA, IKGSR, RSA, Fernet, Elgammal, and ECC.

FIA is a hardware-based attack that exploits vulnerabilities in software to analyze system behavior under induced faults. Fault injection testing allows developers to understand how the system reacts to faults and make necessary design modifications before deployment. The FIA rate for IECC is 0.2336, whereas AES-RSA = 0.4327, IKGSR = 0.3986, RSA = 0.4835, Fernet = 0.3259, Elgammal = 0.5448, and ECC = 0.4217.

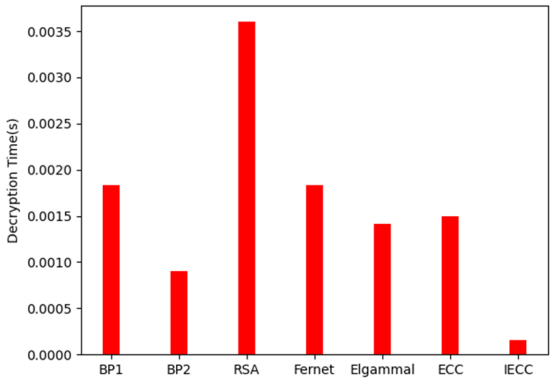
KPA occurs when the attacker has access to both plaintext (crib) and ciphertext, which can reveal secret keys or sensitive information. The KPA attack value for IECC is 0.3168, significantly lower than AES-RSA, IKGSR, RSA, Fernet, Elgammal, and ECC.

SCA is a side-channel attack that targets indirect effects of system execution or hardware behavior rather than the program itself. The lowest SCA attack rate of 0.0348 is obtained using IECC, whereas AES-RSA = 0.2985, IKGSR = 0.4831, RSA = 0.3773, Fernet = 0.3169, Elgammal = 0.3628, and ECC = 0.2871. These results clearly indicate that the IECC scheme provides enhanced security and is more effective against various cryptographic attacks than conventional encryption methods.

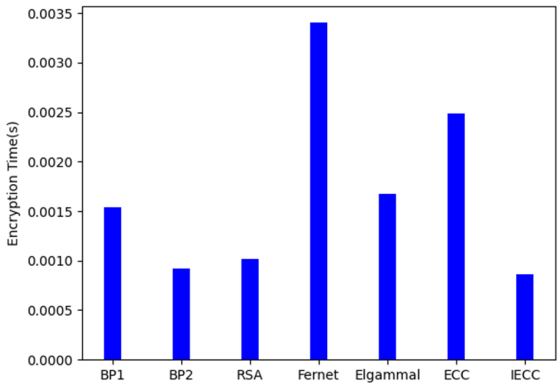
D. Performance Evaluation of IECC and Conventional Strategies for Data Security in EHR

The evaluation of IECC against AES-RSA, IKGSR, RSA, Fernet, Elgammal, and ECC in terms of decryption time, encryption time, key sensitivity, latency, and throughput for data security in EHR is presented in Figure 6. From Figure 6, it can be observed that the IECC approach demonstrated superior performance with enhanced data security. Specifically, the decryption time of the IECC scheme is 0.0013 s, whereas AES-RSA, IKGSR, RSA, Fernet, Elgammal, and ECC achieved 0.0019 s, 0.0083 s, 0.0035 s, 0.0017 s, 0.0014 s, and 0.0015 s, respectively. Likewise, lower encryption time is desirable for improved data security, and IECC achieved lower encryption times compared to the other schemes.

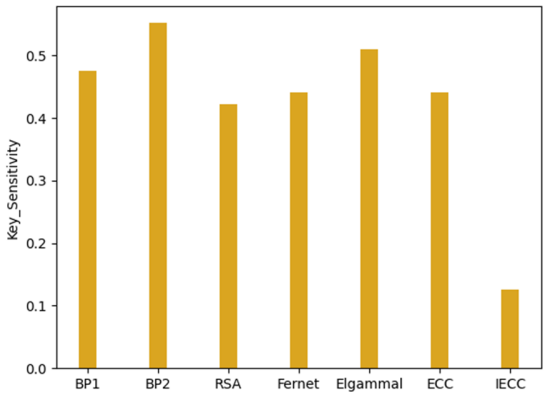
Considering Figure 6(c), the lowest key sensitivity is attained with IECC, outperforming AES-RSA, IKGSR, RSA, Fernet, Elgammal and ECC. Throughput must also be maximized for effective system performance. The IECC scheme achieved a throughput of 40,986, whereas the conventional methods scored lower throughput values. Overall, the ILWKM-based authentication mechanism enables IECC to provide improved data security in EHR systems.



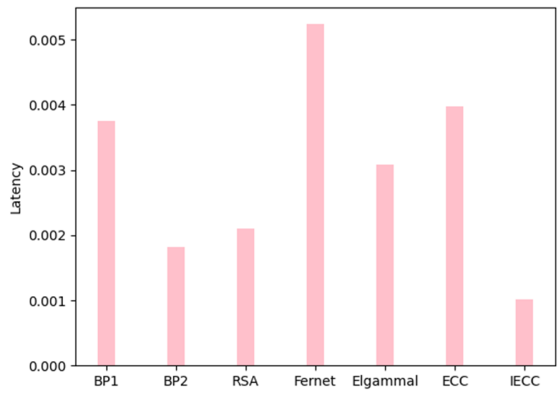
(a)



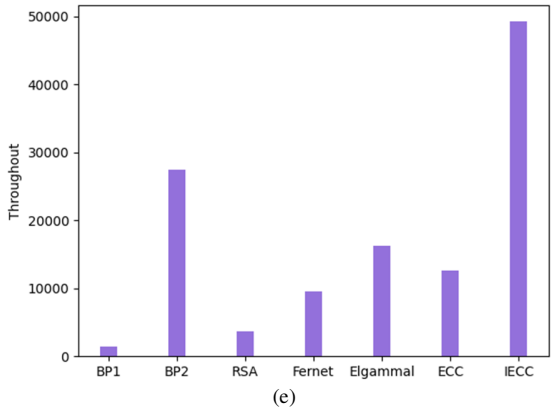
(b)



(c)



(d)



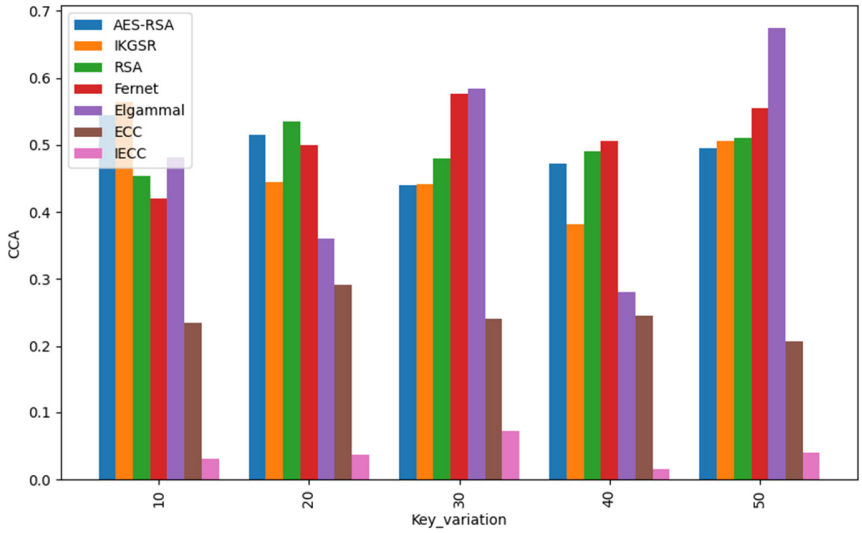
(e)

Fig. 6. Performance evaluation of IECC and conventional schemes for data security in EHR: (a) decryption time, (b) encryption time, (c) key sensitivity, (d) latency, and (e) throughput.

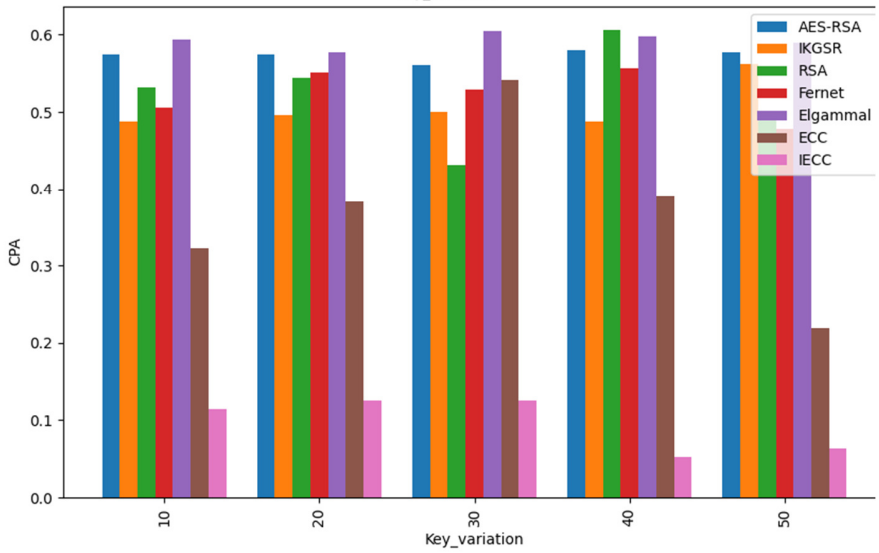
E. Attack Analysis on IECC and Conventional Strategies with Respect to Key Variation

The evaluation of IECC compared to conventional methodologies for different types of attacks (CCA, CPA, FIA, KPA, and SCA) while varying the number of key instances (10–50) is illustrated in Figure 7. Across all attacks, the IECC consistently achieved lower attack rates, providing higher data security in EHR. For the 40th key variation, the IECC exhibited a reduced CCA attack rate compared to AES-RSA, IKGSR, RSA, Elgammal, and ECC. The FIA rate of the IECC was 0.0002 at the 10th key variation, whereas the conventional methods yielded higher values: AES-RSA (0.4735), IKGSR (0.4821), RSA (0.4946), Fernet (0.4895), Elgammal (0.5138), and ECC (0.3487). Additionally, the SCA of the IECC methodology remained extremely low across nearly all key variations. In particular, the IECC achieved an SCA attack rate of 0.0001 at the 20th key variation, significantly lower than AES-RSA, IKGSR, RSA, Fernet, Elgammal, and ECC. Overall, IECC shows stronger resistance to all attack types compared to conventional methods, thanks to ILWKM-based authentication with improved ECC.

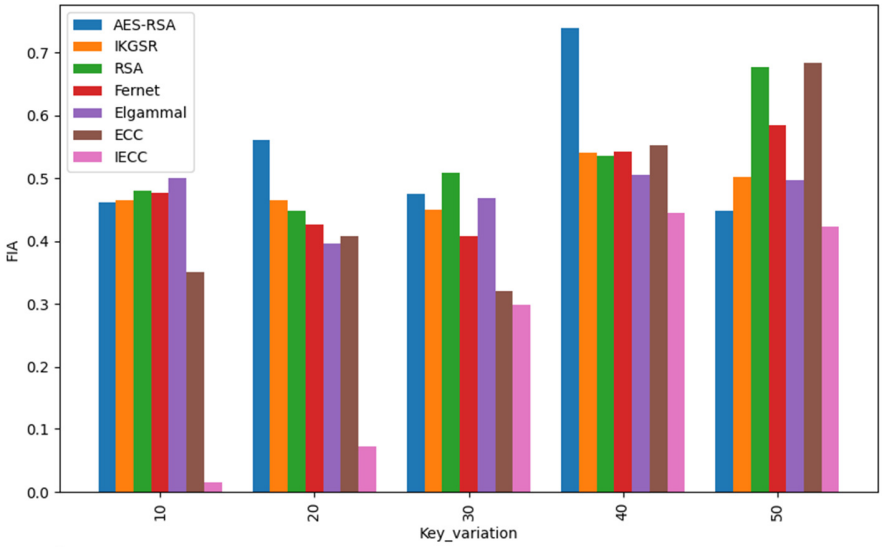
(a)



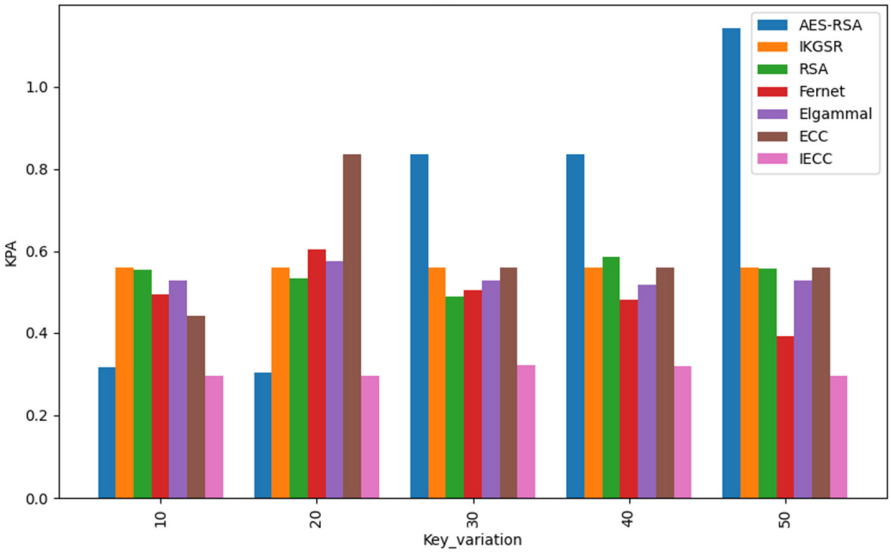
(b)



(c)



(d)



(e)

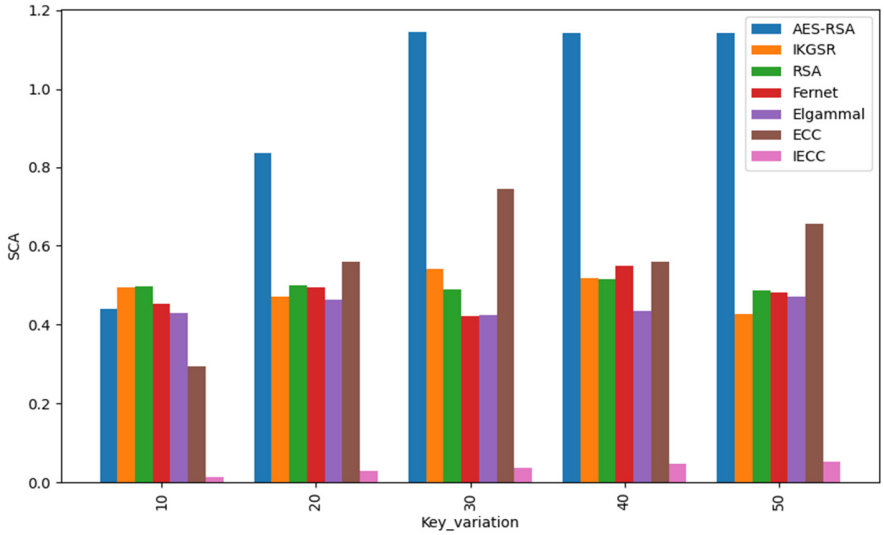


Fig. 7. Attack evaluation on IECC and conventional schemes under varying key variations: (a) CCA, (b) CPA, (c) FIA, (d) KPA, and (e) SCA.

F. Performance Evaluation of IECC and Conventional Strategies with Respect to Key Variation

The performance of IECC compared with conventional methods is evaluated in terms of decryption time, encryption time, key sensitivity, latency, and throughput under varying key variations, as shown in Figure 8. For the 10th key variation, the latency of IECC is 0.0137, whereas the conventional methods recorded higher latency values: AES-RSA = 0.01794, IKGSR = 0.02383, RSA = 0.01639, Fernet = 0.1917, Elgammal = 0.0458, and ECC = 0.01748. Additionally, the lowest encryption and decryption times are achieved by IECC at the 30th key variation, whereas AES-RSA, IKGSR, RSA, Fernet, Elgammal, and ECC show higher values.

G. Comparison on IECC and Levy Flight Adapted Butterfly Optimization Algorithm for Data Security in EHR

The performance of IECC was compared with the Levy Flight Adapted Butterfly Optimization Algorithm (LABOA) [25] in terms of various security and efficiency metrics, as summarized in Table IV.

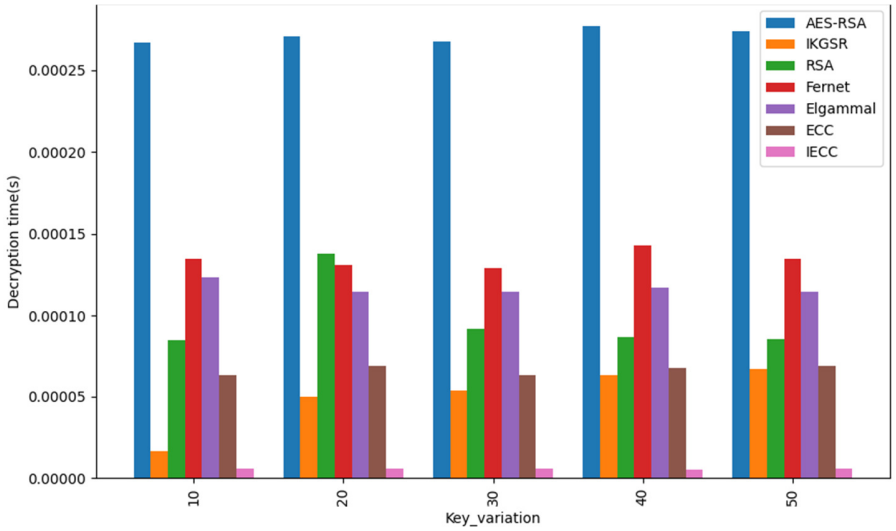
The IECC consistently outperformed LABOA, providing enhanced data security and computational efficiency. Specifically, IECC achieved an encryption time of 0.0009 s, decryption time of 0.0002 s, KPA = 0.3120, CPA = 0.0694, and CCA = 0.2370.

In contrast, LABOA recorded an encryption time of 0.0443 s, decryption time of 0.0438 s, KPA = 0.4564, CPA = 0.4456, and CCA = 0.4865. Thus, the improved ECC contributes to higher data security in EHR.

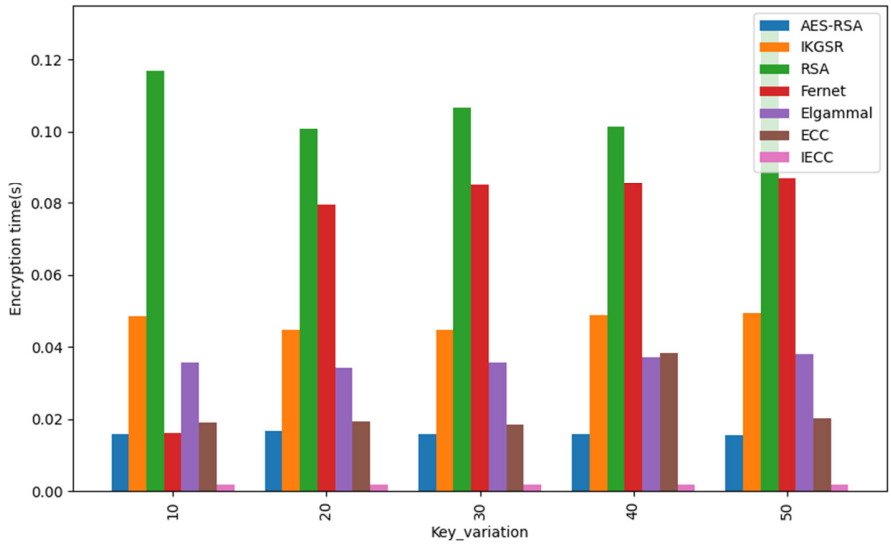
TABLE IV. PERFORMANCE COMPARISON OF IECC AND LABOA

Method	LABOA	IECC
CCA	0.4865	0.2370
Encryption time	0.0443	0.0009
CPA	0.4456	0.0694
Decryption time	0.0438	0.0002
KPA	0.4564	0.3120

(a)



(b)



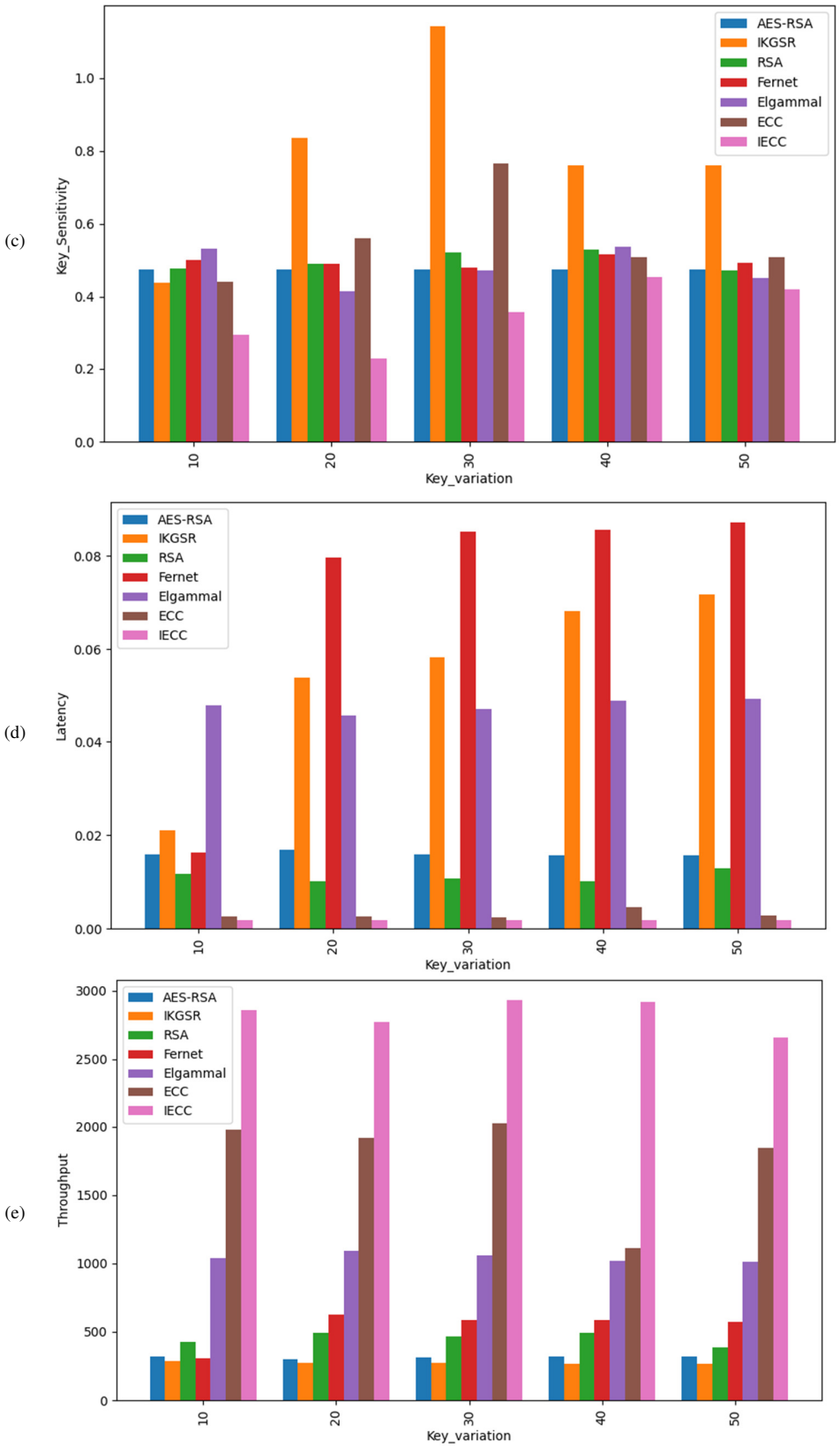


Fig. 8. Performance evaluation of IECC and conventional schemes under varying key variations: (a) decryption time, (b) encryption time, (c) key sensitivity, (d) latency, and (e) throughput.

H. Statistical Study on IECC and Conventional Strategies with Regard to Key Sensitivity for Data Security in EHR

Table V presents a statistical evaluation of the IECC scheme compared with AES-RSA, IKGSR, RSA, Fernet, Elgammal, and ECC for data security in EHR. It is assessed using various statistical metrics. An optimal encryption scheme should have a secret key that is highly sensitive, meaning a small change in the key should produce a completely different ciphertext. The IECC achieved the lowest median key sensitivity value of 0.2277, compared with AES-RSA (0.4751), IKGSR (0.4391), RSA (0.4725), Fernet (0.4804), Elgammal (0.4162), and ECC (0.4405). For the worst-case metric, IECC also produced the minimal value of 0.3583, whereas the other schemes produced higher key sensitivity values. These improvements in the authentication and encryption scheme result in lower key sensitivity, enhancing data security in EHR systems.

TABLE V. STATISTICAL EVALUATION ON IECC AND CONVENTIONAL SCHEMES WITH REGARD TO KEY SENSITIVITY FOR DATA SECURITY IN EHR

Method	Best	Median	Worst	Standard deviation	Mean
AES-RSA	0.4751	0.4751	0.4751	0.4751	0.0000
IKGSR	0.7873	0.4391	0.7602	1.1422	0.2240
RSA	0.4985	0.4725	0.4911	0.5303	0.0236
Fernet	0.4960	0.4804	0.4928	0.5156	0.0118
Elgammal	0.4822	0.4162	0.4731	0.5378	0.0470
ECC	0.5569	0.4405	0.5096	0.7656	0.1110
IECC	0.3513	0.2277	0.3583	0.4555	0.0825

Compared to existing blockchain-based EHR frameworks, such as MedRec and Hyperledger Fabric–based healthcare systems, the proposed approach places greater emphasis on lightweight authentication and encryption efficiency rather than blockchain infrastructure design. While prior works focus on permission management and architectural security, the proposed framework reduces cryptographic computation overhead, making it more suitable for MCC environments. Simulation results demonstrate improved encryption and authentication performance when compared with conventional ECC-based schemes, while maintaining comparable security properties.

IV. CONCLUSION

This work proposed an innovative blockchain-based data security system for Electronic Health Records (EHRs), structured into four phases: registration, contract agreement, authentication, and data uploading and encryption. Although blockchain-based EHR frameworks and Elliptic Curve Cryptography (ECC) techniques have been widely studied, this work distinguishes itself by integrating a lightweight key management and improved ECC encryption within a Mobile Cloud Computing (MCC) environment. Rather than proposing a new blockchain protocol, the primary contribution lies in optimizing authentication efficiency and encryption performance for resource-constrained environments while maintaining compatibility with existing blockchain-based healthcare systems. Experimental evaluation demonstrates that the proposed Integrated Lightweight Key Management Mechanism (ILWKM) and Improved Elliptic Curve Cryptography (IECC) improve cryptographic performance and security metrics in a simulated MCC environment. However, the current study focuses on security and access control behavior rather than full clinical workflow semantics. Incorporating real-world EHR semantics, deploying smart contracts on enterprise blockchains, and providing formal cryptographic proofs represent important directions for future research.

REFERENCES

[1] N. Domadiya and U. P. Rao, "Improving healthcare services using source anonymous scheme with privacy preserving distributed healthcare data collection and mining," *Computing*, vol. 103, no. 1, pp. 155–177, Jan. 2021, https://doi.org/10.1007/s00607-020-00847-0.

[2] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems," *IEEE Access*, vol. 7, pp. 66792–66806, 2019, https://doi.org/10.1109/ACCESS.2019.2917555.

[3] X. Yang, W. Li, and K. Fan, "A revocable attribute-based encryption EHR sharing scheme with multiple authorities in blockchain," *Peer-to-Peer Networking and Applications*, vol. 16, no. 1, pp. 107–125, Jan. 2023, https://doi.org/10.1007/s12083-022-01387-4.

[4] P. Chinnasamy and P. Deepalakshmi, "HCAC-EHR: hybrid cryptographic access control for secure EHR retrieval in healthcare cloud," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 2, pp. 1001–1019, Feb. 2022, https://doi.org/10.1007/s12652-021-02942-2.

[5] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," in *2016 2nd International Conference on Open and Big Data*, Vienna, Austria, 2016, pp. 25–30, https://doi.org/10.1109/OBD.2016.11.

[6] V. K. Yadav, R. K. Yadav, S. Verma, and S. Venkatesan, "CP2EH: a comprehensive privacy-preserving e-health scheme over cloud," *The Journal of Supercomputing*, vol. 78, no. 2, pp. 2386–2416, Feb. 2022, https://doi.org/10.1007/s11227-021-03967-2.

[7] W.-X. Yuan, B. Yan, W. Li, L.-Y. Hao, and H.-M. Yang, "Blockchain-based medical health record access control scheme with efficient protection mechanism and patient control," *Multimedia Tools and Applications*, vol. 82, no. 11, pp. 16279–16300, May 2023, https://doi.org/10.1007/s11042-022-14023-3.

[8] G. Ali *et al.*, "xDBAuth: Blockchain Based Cross Domain Authentication and Authorization Framework for Internet of Things," *IEEE Access*, vol. 8, pp. 58800–58816, 2020, https://doi.org/10.1109/ACCESS.2020.2982542.

[9] M. Shen *et al.*, "Blockchain-Assisted Secure Device Authentication for Cross-Domain Industrial IoT," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 942–954, May 2020, https://doi.org/10.1109/JSAC.2020.2980916.

[10] R. Goyat *et al.*, "Blockchain-Based Data Storage With Privacy and Authentication in Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14203–14215, Aug. 2022, https://doi.org/10.1109/JIOT.2020.3019074.

[11] P. Huang, L. Guo, M. Li, and Y. Fang, "Practical Privacy-Preserving ECG-Based Authentication for IoT-Based Healthcare," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9200–9210, Oct. 2019, https://doi.org/10.1109/JIOT.2019.2929087.

[12] H. Guo, W. Li, M. Nejad, and C.-C. Shen, "Access Control for Electronic Health Records with Hybrid Blockchain-Edge Architecture," in *2019 IEEE International Conference on Blockchain*, Atlanta, GA, USA, 2019, pp. 44–51, https://doi.org/10.1109/Blockchain.2019.00015.

[13] M. Ma, G. Shi, and F. Li, "Privacy-Oriented Blockchain-Based Distributed Key Management Architecture for Hierarchical Access Control in the IoT Scenario," *IEEE Access*, vol. 7, pp. 34045–34059, 2019, https://doi.org/10.1109/ACCESS.2019.2904042.

[14] U. Khalid, M. Asim, T. Baker, P. C. K. Hung, M. A. Tariq, and L. Rafferty, "A decentralized lightweight blockchain-based authentication

mechanism for IoT systems," *Cluster Computing*, vol. 23, no. 3, pp. 2067–2087, Sept. 2020, https://doi.org/10.1007/s10586-020-03058-6.

[15] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K.-K. R. Choo, "An Energy-Efficient SDN Controller Architecture for IoT Networks With Blockchain-Based Security," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 625–638, July 2020, https://doi.org/10.1109/TSC.2020.2966970.

[16] Z. Tian, B. Yan, Q. Guo, J. Huang, and Q. Du, "Feasibility of Identity Authentication for IoT Based on Blockchain," *Procedia Computer Science*, vol. 174, pp. 328–332, Jan. 2020, https://doi.org/10.1016/j.procs.2020.06.094.

[17] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, and X. Lin, "PTAS: Privacy-preserving Thin-client Authentication Scheme in blockchain-based PKI," *Future Generation Computer Systems*, vol. 96, pp. 185–195, July 2019, https://doi.org/10.1016/j.future.2019.01.026.

[18] W. J. Gordon and C. Catalini, "Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 224–230, Jan. 2018, https://doi.org/10.1016/j.csbj.2018.06.003.

[19] M. Min *et al.*, "Learning-Based Privacy-Aware Offloading for Healthcare IoT With Energy Harvesting," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4307–4316, June 2019, https://doi.org/10.1109/JIOT.2018.2875926.

[20] S. Wang, Y. Zhang, and Y. Zhang, "A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018, https://doi.org/10.1109/ACCESS.2018.2851611.

[21] V. Komuravelly and M. Ramchander, "Security and Privacy of Electronic Health Records Sharing using Hyperledger Fabric," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 8, pp. 2410–2413, Aug. 2022, https://doi.org/10.56726/IRJMETS29499.

[22] O. A. Khashan, R. Ahmad, and N. M. Khafajah, "An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks," *Ad Hoc Networks*, vol. 115, Apr. 2021, Art. no. 102448, https://doi.org/10.1016/j.adhoc.2021.102448.

[23] D. V. K. Vengala, D. Kavitha, and A. P. S. Kumar, "Three factor authentication system with modified ECC based secured data transfer: untrusted cloud environment," *Complex & Intelligent Systems*, vol. 9, no. 3, pp. 2915–2928, June 2023, https://doi.org/10.1007/s40747-021-00305-0.

[24] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Heart Disease." UCI Machine Learning Repository, 1989, https://doi.org/10.24432/C52P4X.

[25] B. P. Sindhuri and M. K. Rao, "Blockchain model for authentication and access control-based data privacy in EHR system under mobile cloud platform," *International Journal of Wireless and Mobile Computing*, vol. 29, no. 1, pp. 56–67, Jan. 2025, https://doi.org/10.1504/IJWMC.2025.147647.