International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Secured Electronic Clinical Unified Record Exchange using Blockchain Ledger and Optimized Cryptography

Mrs. Arti Krushnarao Walzade*[1], Dr. Pradnya A. Vikhar [2]

Submitted: 18/11/2023 **Revised**: 30/12/2023 **Accepted**: 10/01/2024

Abstract: Electronic Medical Records (EMRs) contain patients' critical personal and medical information, and safeguarding their privacy and confidentiality is paramount in healthcare. However, centralized EMR systems lack transparency, trust, and security, exposing them to data breaches and jeopardizing patient data privacy. This paper proposes a novel and robust EMR sharing system, Secured Electronic Clinical Unified Record Exchange, using Blockchain Ledger and Optimized Cryptography (SECURE-BLOCK), utilizing cutting-edge blockchain technology, cryptographic techniques, and access controls to address these challenges. The proposed system leverages blockchain technology to create a decentralized, transparent, and secure environment for EMR sharing. Smart contracts enforce access controls and authorization rules, permitting only authorized entities to access and modify EMRs. Digital certificates issued by a Certification Authority (CA) verify the identity of entities making transactions, enhancing trust and security. In addition to blockchain, an improved hybrid cryptographic technique using the Improved RSA algorithm (RSA+) and the Improved Blowfish Cryptography (BC+) algorithm is employed to protect the confidentiality and integrity of EMRs. Encryption algorithms encrypt the EMR data before storing it on the blockchain, ensuring only authorized users with decryption keys can access it. Hash functions generate unique digital fingerprints of EMR data, stored as hash values on the blockchain, ensuring data integrity and tamper-proofing. The proposed system also incorporates an access control and authorization mechanism, allowing only authenticated and authorized users to access and modify EMRs. An authentication server authenticates and authorizes user access, adding a layer of security. Network administrators maintain blockchain nodes, ensuring system integrity and availability. Test outcomes indicate that SECURE-BLOCK rapidly produces hashes, requiring merely 46 milliseconds to generate 100 blocks. Additionally, the system's utilization of RSA+ coupled with BC+ cryptography surpasses alternative cryptographic pairings, achieving encryption and decryption in just 3.106 seconds for 96-bit data. Such results emphasize SECURE-BLOCK's proficiency and reliability in ensuring secure Electronic Medical Records sharing.

Keywords: *EMRs, SECURE-BLOCK, CA, RSA+, BC+.*

1. Introduction

Electronic Medical Records (EMRs) are digital versions of patient health information containing critical personal and medical data [1]. They improve patient care, communication among healthcare providers, and clinical decision-making. However, privacy and confidentiality concerns arise due to the sensitive nature of EMRs.

Safeguarding the privacy and confidentiality of EMRs is paramount in healthcare [2]. Patients can choose who can see their personal and medical information, and healthcare providers must legally and ethically protect patient privacy. If there are breaches in keeping electronic medical records (EMRs) private, it can lead to serious problems like identity theft, medical fraud, and harm to the reputation of healthcare organizations. That's why it's extremely important to have strong security measures to keep EMRs confidential and protect patient privacy.

Existing centralized EMR systems that store records in a single database managed by one entity have some issues with transparency, trust, and security [3]. These systems can be vulnerable to data breaches, insider threats, and unauthorized access because everything is at risk if the central database is compromised [4]. Additionally, these systems lack transparency and accountability, which can result in a lack of trust among patients, healthcare providers, and payers. This lack of trust can make it difficult for EMRs to be widely adopted and hinder sharing of health information among different systems [5].

As a result, there is a need for a novel and robust EMR sharing system that addresses the limitations of existing systems and provides a secure and transparent environment for EMR sharing. This paper proposes a cutting-edge EMR sharing system, Secured Electronic Clinical Unified Record Exchange, using Blockchain Ledger and Optimized Cryptography (SECURE-BLOCK) that utilizes blockchain technology, cryptographic techniques, and access controls to overcome the limitations of existing systems and ensures the privacy and confidentiality of EMRs.

[1,2] *Department of Computer Science and Engineering*
[1] *Research Scholar, Dr. A. P. J. Abdul Kalam University, Indore*
[2] *Research Supervisor, Dr. A. P. J. Abdul Kalam University, Indore, (M.P.), India.*
E-mail Id: [1] artips15@gmail.com, [2]pradnyav123@gmail.com
** Corresponding Author: Mrs. Arti Krushnarao Walzade*
Email: artips15@gmail.com

The Secure-Block system uses a type of technology called blockchain, like a digital ledger that is spread out and doesn't rely on one central authority. It keeps information safe by using codes and ensuring data can't be changed. Blockchain also has built-in security features, like being transparent, being unable to be changed, and needing agreement from a group of people to verify the information. It makes it good for keeping private information, like EMRs, safe. The system doesn't need a central authority and ensures that sharing EMRs is transparent and trustworthy using blockchain.

Smart contracts on the blockchain act as self-executing agreements, ensuring that EMR access and modifications are restricted to authorized personnel. These contracts are encoded with specific conditions and rules that must be fulfilled before any interaction with EMRs is permitted, thereby guaranteeing secure and authorized access.

The Secure-Block system integrates digital certificates from a Certification Authority (CA) to authenticate the identities of parties transacting on the blockchain, bolstering trust and security. These certificates act as a digital form of identification, allowing only verified entities to engage with the system and ensuring legitimate access to EMRs.

Beyond blockchain, Secure-Block applies cryptographic methods to safeguard EMR confidentiality and integrity. It encrypts the EMR data prior to blockchain storage, allowing only those with corresponding decryption keys to access it. Moreover, it utilizes hash functions to create a unique digital fingerprint for the EMR data, which is then stored as a hash value on the blockchain, providing data integrity and preventing tampering.

Additionally, Secure-Block includes an access control and authorization system, permitting only authenticated entities to handle EMRs. Access is governed by smart contracts, which impose specific access rules and conditions. These criteria might encompass the user's role, access purpose, and patient consent. Authorization is granted based on the digital certificates from the CA, ensuring that only verified identities can access the EMRs.

The paper's structure is as follows: Section 2 delves into existing research on EMR sharing in the eHealth sector. Section 3 describes the Secure-Block system, its use of blockchain, cryptographic techniques, access controls, and authorization mechanisms, as well as detailing the Ethereum blockchain implementation, smart contracts, and digital certificates. Section 4 presents the experimental setup and findings discussion. Finally, Section 5 concludes the paper and proposes avenues for future research.

2. Related Work

Ensuring secure EMR sharing is crucial in healthcare, prompting the development and implementation of various innovative solutions. This literature review aims to summarize and evaluate these contemporary approaches focused on enhancing the security of electronic medical record sharing.

Zou et al. proposed SPChain, designed to address the challenges of EMR sharing within both permissioned and public blockchain-based eHealth systems. SPChain optimizes storage and retrieval through unique key blocks and micro blocks and incentivizes participation via a reputation system for medical institutions. It enhances privacy through proxy re-encryption schemes, demonstrating high throughput and minimal storage overhead in real-world testing.

Boumezbeur and Zarour introduced an architecture combining encryption, access control, and storage mechanisms utilizing cloud and blockchain technologies for sharing health records. This system ensures the integrity, privacy, and confidentiality of shared electronic health records by encrypting them and maintaining logs and keys on the blockchain.

Shen et al. developed MedChain, a solution aiming to improve the efficiency of blockchain-based healthcare data sharing. MedChain employs a mix of blockchain, digest chain, and structured P2P network techniques to effectively manage data from sensors and monitoring devices, showing enhanced efficiency and meeting security standards.

Niu et al. offered a permissioned blockchain scheme for medical data sharing, incorporating ciphertext-based attribute encryption for robust data confidentiality and access control. This scheme supports complex keyword connections while maintaining patient privacy and demonstrates resistance against adaptive chosen keyword attacks, along with high retrieval efficiency.

Lastly, Liu et al. presented the BPDS system, leveraging blockchain for privacy-preserving data sharing of EMRs. Addressing the sensitive nature of health data, BPDS focuses on protecting patient identity and ensuring secure data sharing, aiming to improve healthcare quality and reduce costs.

Nguyen et al. introduced an innovative approach for securely sharing Electronic Health Records (EHRs) by integrating blockchain with the decentralized interplanetary file system (IPFS) on a mobile cloud platform. The framework leverages smart contracts for stringent access control, offering reliable and secure data exchanges while safeguarding sensitive health data against potential threats. Key benefits of this system include reduced operational costs, enhanced flexibility, and improved EHR availability. A prototype was developed on the Ethereum blockchain and tested in a real data-sharing scenario via a mobile application and Amazon cloud computing, demonstrating effective results. The system's evaluation indicated notable performance enhancements in lightweight access control, reduced network latency, and elevated security and data privacy, surpassing traditional data-sharing models.

Sun et al. presented a healthcare data security system employing Hyperledger Fabric and the Attribute-Based Access Control (ABAC) framework. This system adopts ABAC for dynamic and granular access to medical data and utilizes smart contracts for secure, immutable storage on the blockchain. Additionally, it integrates IPFS technology to mitigate blockchain storage constraints. The proposed system's advantages include secure storage and integrity of medical information, high throughput for accessing medical data, and the capacity to manage the growing digitalization of healthcare data. Experiments conducted to assess the system's effectiveness validated its proposed benefits.

Mhamdi et al. introduced SEMRAchain, a system that merges RBAC and ABAC with smart contracts to facilitate decentralized, precise, and dynamic access control management for EMR. Leveraging blockchain as a secure, distributed ledger, SEMRAchain enhances the security, reliability, and trustworthiness of EMR sharing, addressing the limitations of centralized access control in traditional medical systems.

Ali et al. developed a blockchain-based framework for securely searching and accessing Personal Health Records (PHR) using homomorphic encryption. This system combines blockchain's distributed database capabilities with a trust chain to offer secure key revocation and policy updates, enhancing the security, efficiency, and effectiveness of digital healthcare data sharing within medical IoT environments.

Hashim et al. explored the application of blockchain for EHR sharing, focusing on maintaining privacy and security. They proposed a transaction-based sharding technique that alleviates scalability issues, resulting in improved throughput, faster consensus times, and handling more appointments compared to conventional healthcare blockchain methods. This technique also reduces cross-shard communication overhead, offering a more streamlined EHR sharing process in healthcare systems.

This section provides a thorough review of various systems and methodologies for sharing electronic medical records (EMRs) in the eHealth sector. Notably, the SECURE-BLOCK system, with its advanced hybrid cryptography utilizing both RSA+ and BC+ algorithms, surpasses previous models in terms of efficiency and other key metrics.

- **Faster Execution Time:** The system takes less time to execute than previous systems, making it a more efficient solution for secure EMR sharing.
- **Enhanced Privacy and Security:** The algorithm uses an improved hybrid cryptography technique to encrypt personal and medical information, ensuring the privacy and security of patient data. It mitigates the risk of unauthorized access and compromises to patient information, protecting patient privacy and confidentiality.
- **Higher Throughput:** The improved hybrid cryptography system provides the highest throughput compared to previous AES-RSA, AES-ECC, and RSA-ECC combinations with various data sizes. This higher throughput allows for faster and more efficient processing of information.
- **Decentralized Database:** The system utilizes a decentralized database less prone to errors and security breaches than centralized databases used in previous systems.
- **Increased Transparency:** The system utilizes a decentralized database, providing real-time updates on the EMR. This increased transparency helps improve the efficiency of the process and reduces the likelihood of errors.
- **Immutable Record Keeping:** Blockchain technology ensures that once data is recorded on the blockchain, it cannot be altered or deleted, providing an immutable record of all transactions.
- **Traceability:** The use of blockchain technology in the system provides a secure and tamper-proof record of all transactions and interactions, making it easier to track and trace any information related to EMR.
- **Access Control and Authorization:** The system utilizes smart contracts to enforce access controls and authorization rules, ensuring that only authorized parties can access patient EMRs. It helps prevent unauthorized access and enhances data security.
- **Secure Sharing of Patient Data:** Authorized medical professionals can securely share patient EMRs with other authorized parties, facilitating a seamless exchange of relevant patient information among healthcare professionals. It can improve patient care coordination and enhance overall healthcare outcomes.
- **Digital Certificates for Identity Verification:** The system uses digital certificates issued by the Certification Authority (CA) to verify the identity of entities such as patients, doctors, and other medical

professionals. It helps prevent identity fraud and enhances the security of the system.

- **Secure Authentication:** The system uses an authentication server to authenticate and authorize user access, storing user credentials securely. It helps prevent unauthorized access to the system, enhancing data security.
- **Role-Based Access:** The system grants access to EMRs based on the roles and responsibilities of healthcare professionals, ensuring that they can only view and update patient data relevant to their scope of practice. It enhances data privacy and security by limiting access to authorized personnel only.

Overall, the proposed SECURE-BLOCK system with improved hybrid cryptography provides a more secure, efficient, transparent, high-throughput, reliable record-keeping and traceable solution for secure EMR sharing compared to previous works.

3. Secured Electronic Clinical Unified Record Exchange using Blockchain Ledger and Optimized Cryptography (Secure-Block)

The safeguarding of EMRs is of paramount importance in healthcare to ensure patient data privacy and confidentiality. However, existing centralized EMR systems often lack transparency, trust, and security, making them vulnerable to data breaches and posing risks to patient information. To address these challenges, a novel and robust EMR sharing system called SECURE-BLOCK (Secured Electronic Clinical Unified Record Exchange using Blockchain Ledger and Optimized Cryptography) has been proposed. SECURE-BLOCK leverages cutting-edge blockchain technology, cryptographic techniques, smart contracts, digital certificates, and access control mechanisms to create a decentralized, transparent, and secure environment for EMR sharing. By utilizing improved hybrid cryptographic techniques, SECURE-BLOCK aims to enhance the security, privacy, and trustworthiness of EMR sharing. This section presents a detailed description of the SECURE-BLOCK system, highlighting its key features and benefits for patients, healthcare professionals, and other stakeholders in ensuring the confidentiality and integrity of EMR data, mitigating risks associated with data breaches, and providing a secure and transparent environment for EMR sharing.

3.1 System architecture with the implementation of SECURE-BLOCK system:

The SECURE-BLOCK system is specifically developed to overcome the constraints and difficulties encountered in conventional EMR sharing systems. The next part will provide a concise overview of the system architecture used by the SECURE-BLOCK system. The system architecture has eight primary components: 1) Administrator 2) Distributed ledger technology 3) Server responsible for verifying the identity of users. 4) Organisation that issues digital certificates to verify the authenticity of entities. 5) Physician 6) Registered Nurse 7) Pharmacist 8) Individual seeking medical treatment. Figure 1 depicts the graphical depiction of the system architecture of the SECURE-BLOCK system.

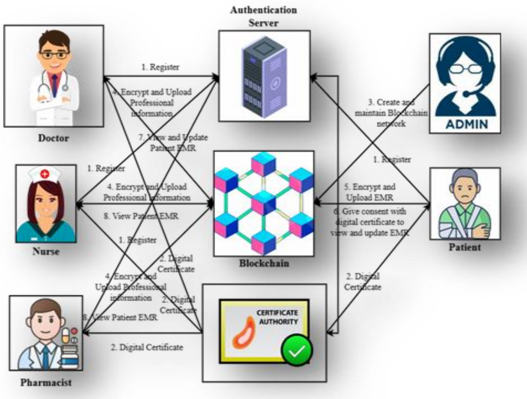
Authentication Server
Doctor
1. Register
4. Encrypt and Upload Professional information
7. View and Update Patient EMR
3. Create and maintain Blockchain network
ADMIN
Nurse
1. Register
4. Encrypt and Upload Professional information
8. View Patient EMR
Blockchain
1. Register
5. Encrypt and Upload EMR
6. Give consent with digital certificate to view and update EMR
Patient
1. Register
2. Digital Certificate
4. Encrypt and Upload Professional information
2. Digital Certificate
8. View Patient EMR
2. Digital Certificate
2. Digital Certificate
CERTIFICATE AUTHORITY
Pharmacist

Fig 1: Architecture of the SECURE-BLOCK system

Figure 1 depicts a system specifically created to facilitate the registration and verification process of medical personnel, such as physicians, nurses, chemists, and patients, in a healthcare setting. This system primarily focuses on the provision of digital certificates for authentication, the development and maintenance of a blockchain network, and the safe encryption and uploading of professional credentials and electronic medical records (EMR).

1. Registration and Authentication: The procedure starts by enrolling physicians, nurses, chemists, and patients on an authentication server. During this phase, users are required to provide crucial personal and professional information for the purpose of verification, in order to establish the authenticity of their identity.
2. Digital certificates are issued by a certifying authority to persons who have successfully registered and been validated. These certificates serve as evidence of identification and are essential for the encryption and decryption operations inside the system.
3. Blockchain Network Establishment: The task of establishing and maintaining a blockchain network falls under the responsibility of a network administrator. This network functions as a safe and decentralised system for documenting transactions and information. It serves as a transparent and secure platform for

keeping encrypted professional data and electronic medical records (EMR).

4. Medical practitioners have the ability to encrypt and transfer their professional information, including qualifications and work experience, onto the blockchain. The data, which is securely kept on the network, may only be accessed by authorised users.
5. EMR Encryption and Upload: Patients have the option to encrypt and upload their EMRs, encompassing their medical history, diagnoses, and treatments. Encryption ensures the confidentiality and security of this sensitive information.
6. Patient Consent for EMR Access: Patients control who can access their EMR by providing consent through their digital certificate. They can specify which healthcare providers are permitted to view and modify their records.
7. Accessing Patient EMR: Authorized doctors have the privilege to both view and update a patient's EMR, facilitating accurate and timely medical care. Nurses and pharmacists, while able to view EMRs, do not possess the authority to alter them, ensuring they have the necessary information for patient care without the ability to make changes.

Essentially, this framework provides a safe and transparent system for registering and verifying healthcare professionals and patients. This process encompasses the issuance of digital certificates to verify identities, the establishment and maintenance of a blockchain network to securely store professional credentials and EMRs, and the facilitation of authorised access to patient EMRs based on designated roles and permissions. This system ensures the privacy, protection, and precision of healthcare data, while also allowing authorised persons to access and edit pertinent information for the purpose of providing exceptional healthcare services.

3.1.1 Doctor, Nurse, Pharmacist, and Patient Registration with Authentication Server:

The next section delineates the enrollment process of healthcare professionals and patients in the Secure-Block system, which involves an authentication server that verifies their identities. During the registration process, participants are required to enter vital information such as their name, email address, and their designated function, which might be doctor, nurse, chemist, or patient. The authentication server then validates these particulars to validate the validity of the user. Upon registration, a HmacSHA256 hash, which is obtained from the user's email, is created and sent to the user. The distinctive hash is then used for system authentication and authorization.

1. The registration procedure generally comprises the following steps:
2. Personal Information Collection: Individuals provide personal data, such as their name and email address, upon registering. Furthermore, they express their position within the healthcare environment.
3. Verification using Authentication Server: The server verifies the accuracy and genuineness of the submitted information. This may include verifying the email against existing data and validating the job inside the system.
4. After verifying, the server creates a HmacSHA256 hash using the given email. This cryptographic hash function generates a safe and immutable value of a predetermined size for the purpose of authentication.
5. The hash value is sent to users, usually via a secure means such as email, to ensure uniqueness. This value is crucial for future access to the system.
6. System Login and Access: Users are required to provide their email and the hash value they previously gave in order to log in. The system validates the hash by comparing it to its stored data, and if they match, it allows access according to the user's assigned role.

Algorithm 1 discusses the details of Doctor, Nurse, Pharmacist, and Patient Registration with the Authentication Server.

Algorithm 1: Doctor, Nurse, Pharmacist, and Patient Registration with Authentication Server

```
Input  : User information (name, email ID, role)
Output : A hash value (HmacSHA1) for authentication
                /* Registration Process */
Step 1 : Begin the registration process for doctors, nurses, pharmacists, and patients.
Step 2 : Input user information, including name, email ID, and role.
Step 3 : Verify user information with the authentication server.
Step 4 : If user information is valid:
Step 5 : Generate HmacSHA256 hash value using an email ID.   // Algorithm 2
Step 6 : Store the hash value in the authentication server’s database linked to the user’s account.
Step 7 : Notify user of successful registration and provide them with the generated hash value.
Step 8 : End registration process.
                /* Login Process */
```

Step 9 : Begin the login process for registered users.

Step 10 : Input email ID and a hash value for authentication.

Step 11 : Retrieve stored hash value from the authentication server’s database based on the provided email ID.

Step 12 : If the retrieved hash value matches the provided hash value:

Step 13 : Grant user access to the system.

Step 14 : User can now view and update their profile or perform other authorized actions based on their role.

Step 15 : End login process.

Algorithm 1 is a registration and authentication process for users with different system roles (doctor, nurse, pharmacist, and patient). The algorithm uses the HmacSHA256 hash algorithm for authentication.

The registration process begins by taking user information as input, including name, email ID, and role (Step 2). This information is then verified with the authentication server (Step 3). If the user information is valid, the algorithm generates a hash value using the HmacSHA256 algorithm, using the email ID explained in Algorithm 2 (Step 5). The hash value is then stored in the authentication server’s database, linked to the user’s account (Step 6). Finally, the user is notified of successful registration and provided with the generated hash value (Step 7). The registration process is then completed (Step 8).

The login process begins by taking the email ID and hash value as input for authentication (Step 10). Next, the algorithm retrieves the stored hash value from the authentication server’s database based on the provided email ID (Step 11). If the retrieved hash value matches the provided hash value, the user is granted access to the system (Step 13). The user can then view and update their profile or perform other authorized actions based on their role (Step 14). The login process is then completed (Step 15).

Algorithm 2 depicts generating a hash for Plaintext using the HMACSHA256 algorithm.

Algorithm 2: HMACSHA256-based Hash generation algorithm

Input: Plaintext (the text to be hashed), secretKey (the key for hashing)

Output: OuterHash (the final hash result)

// Step 1: Pad the secret key if necessary

1. **Check Secret Key Length**:

    - If the length of secretKey is greater than 64 bytes: 2. Hash the secretKey using SHA256: secretKey = SHA256(secretKey)
    - If the length of secretKey is less than 64 bytes: 4. Pad the secretKey with zeros or other specified padding values until its length reaches 64 bytes: secretKey = padKey(secretKey)

// Step 2: Generate the inner and outer padding keys 5. **Generate Inner Padding Key**:

- InnerPaddingKey = xor(secretKey, 0x36)

6. **Generate Outer Padding Key**:

    - OuterPaddingKey = xor(secretKey, 0x5C)

// Step 3: Calculate the inner hash 7. **Calculate Inner Hash**:

- InnerHash = SHA256(concatenate(InnerPaddingKey, Plaintext))

// Step 4: Calculate the outer hash 8. **Calculate Outer Hash**:

- OuterHash = SHA256(concatenate(OuterPaddingKey, InnerHash))

// Step 5: Return the final hash 9. **Return OuterHash**: Return the calculated OuterHash as the final hash result.

The word 'secretKey' in technique 2 denotes a vital element used in the HMACSHA256 technique for generating a secure hash from a provided plaintext. This key is crucial for establishing the connection between the plaintext and its hash, verifying its validity and integrity. It serves as a secure shared component between the sender and recipient of the communication and is combined with the plaintext to produce the hash. Therefore, ensuring the confidentiality and safe transmission of the 'secretKey' among the people involved is crucial for maintaining the security of the hash.

The 'secretKey' plays a role in generating two padded keys inside the framework of the HMACSHA256 method. These keys are then used to compute the inner and outer hashes, which together constitute the final HMACSHA256 hash. Algorithm 2 employs several functions: 'SHA256' to calculate the SHA-256 hash of the input, 'concatenate' to combine two strings, 'xor' to execute a bitwise exclusive-or operation, and 'padKey' to expand the secret key to a length of 64 bytes by adding zeros or other specified padding if necessary. The 'HMACSHA256' function requires 'plaintext' and 'secret_Key' as inputs and produces the resulting HMACSHA256 hash.

The method assigns the hexadecimal values '0x36' and '0x5C' to the binary sequences '0011 0110' and '0101 1100', respectively. These are used as both internal and external padding in the HMACSHA256 method. The inner padding is obtained by performing a bitwise exclusive-or ('xor') operation between the 'secretKey' and '0x36'. Similarly, the outer padding is generated by 'xor'ing the 'secretKey' with '0x5C'. The padding keys play a significant role in calculating both the inner and outer hashes, which together provide the entire HMACSHA256 hash. The padding step enhances the security of the HMACSHA256 technique by including both the plaintext and the 'secretKey' into the hash calculation.

The registration procedure and subsequent hash value creation, which is dependent on the email id, give an extra degree of protection, guaranteeing that only authorised users may register and get access to the system.

3.1.2 Issuance of Digital Certificates:

After a successful registration, a Certification Authority (CA) grants digital certificates to the newly enrolled persons. These certificates serve as authenticated identification for users in the system and include the user's email ID as well as cryptographic keys, including an RSA+ public key, an RSA+ private key, and a BC+ secret key, which are necessary for encryption and decryption operations. The RSA+ public key and BC+ secret key are used for encrypting communications sent to the user, while the RSA+ private key and BC+ secret key are used for decrypting received messages and performing other authorised actions inside the system.

The system uses a hybrid encryption methodology, capitalising on the benefits of both symmetric and asymmetric encryption to attain an ideal balance of velocity and security. This technique guarantees strong security while still ensuring efficient operations. Figure 2 illustrates the installation of an upgraded hybrid cryptography architecture, showcasing the use of this sophisticated and safe technology.

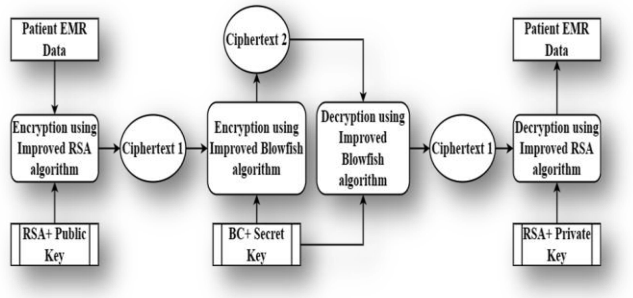
Patient EMR Data
Encryption using Improved RSA algorithm
RSA+ Public Key
Ciphertext 1
Encryption using Improved Blowfish algorithm
BC+ Secret Key
Ciphertext 2
Decryption using Improved Blowfish algorithm
Ciphertext 1
Decryption using Improved RSA algorithm
RSA+ Private Key
Patient EMR Data

Fig 2: Improved hybrid cryptography

Complex cryptographic processes are required in order to keep the system's security and privacy intact. These operations include the generation of the RSA+ public key, the encryption of the RSA+ private key, and the decryption of the BC+ secret key. In order to maintain the integrity and confidentiality of user data and communications inside the system, it is vital to ensure that these keys are managed and protected in a secure manner. In this part, the intricacies of producing, encrypting, and decrypting the RSA+ public key, RSA+ private key, and BC+ secret key are discussed in depth.

Improved RSA algorithm (RSA+) based asymmetric key cryptography*:*

The RSA encryption algorithm, a standard tool for secure data transmission, can be bolstered by increasing the key size, although this often leads to longer encryption and decryption durations. The RSA+ algorithm mitigates this issue by segmenting data into smaller parts and integrating two extra prime numbers beyond the standard RSA framework. This enhancement not only strengthens the encryption process but also expedites encryption and decryption times, making it highly suitable for blockchain systems.

Despite its widespread use, the traditional RSA algorithm has drawbacks, including slow operational speeds and constrained key sizes, making it potentially vulnerable to security breaches. The RSA+ algorithm, as implemented in the proposed EMR sharing system, intensifies the encryption complexity and quickens the cryptographic processes, addressing these vulnerabilities.

Crucial for bolstering data security and efficiency in systems like blockchain, which necessitate the secure handling of extensive data, the RSA+ algorithm offers significant advantages. It ensures enhanced security through larger key sizes and quicker encryption and decryption, making it an ideal choice for large-scale systems. The process and merits of this asymmetric key cryptography are further elucidated in Algorithm 3, detailing the RSA+ algorithm's application.

Algorithm 3: Improved RSA algorithm (RSA+) based asymmetric key cryptography

Input: Plaintext (the text to be encrypted)

Output: EncryptedPlaintext_1 (the encrypted text after encryption) and Plaintext (the text after decryption)

```
/* Improved RSA Public Key and Private Key generation */
    1. Choose Large Prime Numbers: Select four large prime numbers X, Y, Z, and W.
    2. Calculate M: Compute M as the product of X, Y, Z, and W.
    3. Calculate N: Determine N as the product of X and Y.
    4. Calculate S: Calculate S as (X-1) * (Y-1) * (Z-1) * (W-1).
    5. Generate Public Key: Choose a Public Key (PK) such that GCD(PK, S) = 1.
    6. Generate Private Key: Choose a Private Key (SK) such that SK = (1 / PK) mod(S).
    7. Create Public and Private Key Pairs: Form PublicKey as a pair of PK and M, and PrivateKey as a pair of SK and N.
/* Encryption */ 8. Extract PK and M: Retrieve PK and M from PublicKey.
    9. Split Plaintext: Divide the Plaintext into n chunks.
    10. Initialize EncryptedPlaintext_1: Set EncryptedPlaintext_1 as an empty string.
    11. Encrypt Each Chunk:
        • For i = 0 to n:
            • Convert T[i] to a byte array BA[].
            • Calculate T2 as (T[i] ^ PK) mod (M).
            • Append T2 to EncryptedPlaintext_1.
        • End for
/* Decryption */ 12. Extract SK and N: Extract SK and N from PrivateKey.
    13. Split EncryptedPlaintext_1: Split EncryptedPlaintext_1 into n chunks.
    14. Initialize Plaintext: Set Plaintext as an empty string.
    15. Decrypt Each Chunk:
        • For i = 0 to n:
            • Retrieve ET as T[i].
            • Calculate BA[] as (ET ^ SK) mod (N).
            • Convert BA to a string T1.
            • Append T1 to Plaintext.
        • End for
```

This process outlines the steps for both encryption and decryption using the Improved RSA Public Key and Private Key generation method.

lgorithm 3 described is an improved version of the RSA (Rivest-Shamir-Adleman) asymmetric key cryptography algorithm, denoted as RSA+. It takes a plaintext message as input and generates an encrypted version of the Plaintext using a public key. The encryption process involves splitting the Plaintext into chunks, converting each chunk into a byte array, raising it to the power of a chosen public key (PK), and taking the modulus with a large number (M). Finally, the resulting values are concatenated to form the encrypted Plaintext (EncryptedPlaintext_1).

The algorithm uses a private key, which is generated along with the public key, to decrypt the encrypted Plaintext. The private key comprises a secret key (SK) and a modulus (N). The decryption process involves splitting the encrypted Plaintext into chunks, raising each chunk to the secret key's (SK) power and taking the modulus with N. The resulting values are converted from byte arrays to strings and concatenated to obtain the original Plaintext.

The improved RSA+ algorithm also incorporates additional steps for key generation. It involves choosing four large prime numbers (X, Y, Z, and W) instead of two, increasing the algorithm's security. It also includes a step to ensure that the chosen public key (PK) is relatively prime to the product of (X-1), (Y-1), (Z-1), and (W-1), denoted as S, to enhance the security of the encryption process. Overall, RSA+ aims to provide a more secure and efficient method for asymmetric key cryptography than the original RSA algorithm.

Improved Blowfish Cryptography (BC+):

Blowfish, a symmetric block cipher algorithm, operates on 64-bit data blocks and supports key lengths up to 448 bits. However, the ever-evolving nature of cryptographic threats necessitates continual enhancements in security measures. To address this, an enhanced version termed Improved Blowfish Cryptography (BC+) has been introduced. BC+ modifies the original Blowfish algorithm, possibly through additional encryption rounds or alterations in block or key sizes, aiming to fortify its defense mechanisms. The primary benefit of BC+ lies in its heightened security capabilities, making it more resilient against emerging and sophisticated cyber threats. Its robustness renders BC+ applicable across various security-reliant domains. The operational mechanics and specifics of BC+ are further expounded in Algorithm 4, providing insight into its enhanced cryptographic process.

Algorithm 4: Improved Blowfish Cryptography (BC+)

Input: Input_Plain_text (the text to be encrypted)

Output: Output_Cipher_text (the encrypted text) and Output_Decrypted_Plain_text (the decrypted text)

/* Encryption */

1. **Initialize Key Schedule**: Create the key_schedule using the provided key, which can be either a string or a binary key.
2. **Divide into Blocks**: Split the Input_Plain_text into blocks of N bytes (N * 8 bits), with N as the new block_size.
3. **Process Each Block**:
    - For each block, perform an XOR operation with the P_box, and then split it into two halves (left and right).
4. **Substitution and Permutation Rounds**: Execute M rounds of substitution and permutation on the two halves using the key_schedule, where M represents the number of additional rounds.
5. **Swap and XOR**: Swap the left and right halves, and XOR the result with the P_box.
6. **Repeat for All Blocks**: Repeat steps 4 and 5 for every block of Input_Plain_text.
7. **Assemble Cipher Text**: Compile the Cipher_text blocks and provide the final result as Output_Cipher_text.

/* Decryption */ 8. **Initialize Key Schedule**: Establish the key_schedule using the provided key, which can be either a string or a binary key.

9. **Divide into Blocks**: Segment the Output_Cipher_text into blocks of N bytes (N * 8 bits), with N as the block_size.
10. **Process Each Block**:
    - For each block, perform an XOR operation with the P_box, and then split it into two halves (left and right).
11. **Reverse Substitution and Permutation Rounds**: Execute M rounds of substitution and permutation on the two halves using the key_schedule in reverse order, where M represents the number of additional rounds.
12. **Swap and XOR**: Swap the left and right halves, and XOR the result with the P_box.
13. **Repeat for All Blocks**: Repeat steps 4 and 5 for every block of Output_Cipher_text.
14. **Assemble Decrypted Plain Text**: Compile the Decrypted_Plain_text blocks and provide the final result as Output_Decrypted_Plain_text.

The Enhanced Blowfish Cryptography (BC+) algorithm, detailed in Algorithm 4, offers a significant security advantage. By incorporating additional rounds, it increases the complexity of the data encryption process. This escalation in the number of transformations that the plaintext undergoes substantially complicates the task for potential attackers trying to discern patterns or decrypt the key. The enhanced security of the BC+ algorithm is further bolstered by the use of longer keys. With an increase in key length, the number of potential keys an attacker must attempt grows exponentially, making unauthorized decryption considerably more difficult. Additionally, modifying the block size in the BC+ algorithm enables it to handle larger data units more efficiently, which can lead to both improved processing speed and potentially greater security. Overall, the combination of more encryption rounds and the use of a larger block size significantly elevates the overall security efficacy of the BC+ algorithm.

3.1.3 The Role of Blockchain in SECURE-BLOCK system:

When it comes to the SECURE-BLOCK system, the blockchain is an essential component in maintaining the safety and reliability of the electronic medical record (EMR). In this system, the blockchain is primarily responsible for the following responsibilities:

1. Decentralisation: Blockchain technology makes it possible to create a decentralised and distributed network. This eliminates the need for a central authority to oversee and administer electronic medical record (EMR) data, which in turn ensures that the system is transparent and trustworthy.
2. Security: Blockchain uses an upgraded hybrid cryptographic technology to safeguard electronic medical record (EMR) data. This protects the data from unauthorised access, alteration, and data breaches, therefore maintaining the confidentiality and integrity of patient information.
3. Transparency: Blockchain technology offers a record that is both visible and auditable of all transactions and modifications made to electronic medical record

(EMR) data. This transparency enables traceability and accountability.

4. Consensus: Blockchain technology employs a consensus method, such as Proof-of-Work (PoW), to guarantee that all transactions are validated by several nodes in the network. This helps to maintain the integrity of transactions and prevents fraudulent activity.
5. Smart Contracts: Blockchain technology makes use of smart contracts, which are agreements that can carry out their own execution, to enforce access control measures, permissions, and consent preferences of patients. This ensures that only authorised parties are able to view and alter electronic medical record documents (EMR).
6. Digital Certificates Blockchain technology leverages certifying authorities to issue digital certificates to healthcare professionals and patients. These certificates may be used as a form of identification verification as well as for the purposes of encryption and decryption.
7. An Immutable Ledger Blockchain technology maintains an immutable ledger of all transactions, modifications, and updates to electronic medical record (EMR) data. This provides a chronicle of events that cannot be altered.
8. By decreasing the need for duplicate data input and guaranteeing that data is accurate and consistent, blockchain technology makes it possible for authorised parties to share electronic medical record (EMR) data in a way that is both efficient and safe.
9. Patients are able to exercise control over their electronic medical record (EMR) data via the use of digital certificates, which is made possible by blockchain technology. It gives patients the ability to decide who may access and change their information, which protects the patient's autonomy and maintaining their privacy.
10. Reliability: Blockchain technology improves the level of confidence among many stakeholders, such as patients, healthcare professionals, and others, by offering a system that is safe, transparent, and auditable for electronic medical record (EMR) sharing. This system guarantees the confidentiality, security, and integrity of data.
11. Disaster Recovery: Blockchain technology maintains electronic medical record (EMR) data over a distributed network of nodes. This ensures that the data is redundant and resilient to single-point failures, which provides better capabilities for disaster recovery.
12. Trust and Reputation: Blockchain technology makes it possible for healthcare organisations to build trust and reputation by including their verified digital certificates and transaction history into their records. In addition to lowering the likelihood of fraudulent acts, it fosters trust among the many parties.

Algorithm 5 discusses the use of blockchain in the SECURE-BLOCK system.

Algorithm 5: Use of blockchain in SECURE-BLOCK system

Input: Encrypted User Details (EUD), which include the professional profiles of Doctors, Nurses, and Pharmacists, as well as the EMR details of patients, and the blockchain itself.

Output: A notification event signaling the successful addition of new user details to the blockchain.

1. **Initialize Previous Block Hash**: Set Previous_Block_Hash to 0.
2. **Initialize Blockchain Size**: Set Size_of_the_Blockchain to 0.
3. **Iterate Through Blocks**: Begin looping through each block B in the blockchain.
4. **Increment Blockchain Size**: Increase Size_of_the_Blockchain by 1 for each block.
5. **Update Previous Block Hash**: Set Previous_Block_Hash to the hash of the current block B.
6. **End Loop**: Conclude the iteration through the blocks.
7. **Determine Block Number**: Assign Block_number as Size_of_the_Blockchain + 1.
8. **Record Time**: Assign Time-stamp with the current time.
9. **Generate Nonce**: Assign Nonce a randomly generated number.
10. **Generate Hash**: Create a hash for Block_number, Time-stamp, Nonce, and EUD using the specified algorithm (Algorithm 2).
11. **Check Blockchain Size**: If Size_of_the_Blockchain is 0,
12. **Create Genesis Block**: Formulate the Genesis_Block with Block_number, Time-stamp, Nonce, Hash, and Previous_Block_Hash.
13. **Upload Genesis Block**: Add the Genesis_Block to the blockchain.
14. **Otherwise, Create Succeeding Block**: If not, formulate the Succeeding_Block with the same details.
15. **Upload Succeeding Block**: Add the Succeeding_Block to the blockchain.

16. **Conclude the Process**: Finish the conditional structure.
17. This sequence ensures that either a Genesis Block or a Succeeding Block, containing the new user details, is securely added to the blockchain, and an event is announced to confirm the addition.

Algorithm 5 delineates the procedure for incorporating encrypted user details (EUD) such as the professional credentials of doctors, nurses, and pharmacists, along with the EMR details of patients, into a blockchain. The algorithm commences with the initialization of variables like Previous_Block_Hash and Size_of_the_Blockchain to monitor the current state of the blockchain. As it progresses through each block, it increments the Size_of_the_Blockchain and refreshes the Previous_Block_Hash with the hash of the latest block. Subsequently, the algorithm concocts a hash for the impending block by amalgamating elements such as Block_number, Time-stamp, Nonce, and the EUD. In scenarios where the blockchain is initially empty (Size_of_the_Blockchain equals 0), it fabricates a Genesis_Block containing the pertinent details and integrates it into the blockchain. Conversely, if the blockchain already contains data, a Succeeding_Block is formulated and appended. This methodology ensures the secure and verifiable addition of user details to the blockchain, thereby broadcasting the event of new user additions. By harnessing the capabilities of blockchain technology, this algorithm significantly bolsters the security and transparency of the SECURE-BLOCK system in managing and recording user details.

3.1.4 Digital Certificate-Based Access Control for Patient EMR:

Digital Certificate-Based Access Control for Patient EMR is a method of controlling access to patient health information stored in the blockchain using digital certificates. Digital certificates are electronic credentials that can be used to securely verify users' identity and permissions.

In the context of patient EMR, digital certificates can be used to authenticate healthcare providers such as doctors, nurses, pharmacists, and patients. These digital certificates contain consent preferences and access permissions, which specify who can view and update the patient's EMR data. The digital certificates are issued and managed by a certificate authority. Algorithm 6 outlines the Digital Certificate-Based Access Control for Patient EMR.

Algorithm 6: Digital Certificate-Based Access Control for Patient EMR

Input: Patient's digital certificate with consent preferences and access permissions, healthcare professionals' (doctors, nurses, pharmacists) credentials for authentication, and patient's EMR data.

Output: Authorized viewing and/or modification of the patient's EMR based on established consent preferences and access permissions.

1. **Consent Preferences**: Patients explicitly outline who is permitted to view and modify their EMR through their digital certificate.
2. **Credential Authentication**: The system authenticates the credentials of doctors, nurses, and pharmacists to verify their registration and appropriate access rights.
3. **Certificate Verification**: The validity of the patient's digital certificate is confirmed to ensure it aligns with the patient's stated consent preferences.
4. **Doctor Access**: If a doctor is authenticated and permitted by the digital certificate, they gain access to view and modify the patient's EMR.
5. **Nurse Access**: Authenticated nurses with appropriate permissions via the digital certificate can view but not modify the patient's EMR.
6. **Pharmacist Access**: Similarly, authenticated pharmacists with access permissions can view the patient's EMR but are restricted from making changes.
7. **Access Denial**: Access to the patient's EMR is denied if any user's credentials are unauthenticated or if the digital certificate doesn't grant permission.
8. **Audit Logging**: All attempts to alter the patient's EMR are recorded for future audits.
9. **Monitoring Compliance**: The algorithm continuously oversees access and modifications to maintain adherence to consent preferences and access permissions.
10. **Termination**: The algorithm concludes its operation once the authorized tasks are fulfilled or when users log out of the system.

Algorithm 6 provides digital certificate-based access control for patient EMR management. The patient's digital certificate, which contains consent choices and access rights, healthcare providers' authentication credentials, and EMR data are needed. The algorithm allows authorised reading and/or alteration of the patient's EMR based on consent and permissions.

Patients provide express EMR access consent choices using their digital certificates. The algorithm then verifies physicians, nurses, and chemists' system registration and access rights. It also verifies the patient's digital certificate and consent choices.

Physicians may see and alter patient EMRs if their credentials are validated and their digital certificate validates access rights. After credential authentication and digital certificate verification, nurses and chemists may see and modify the patient's EMR on permission.

Users cannot access the patient's EMR if their credentials fail authentication or the digital certificate does not authorise access. For auditing, each unauthorised EMR change is documented. The algorithm monitors access and changes to ensure consent choices and permissions are followed until authorised tasks are performed or users leave the system.

This framework securely and transparently registers and authenticates healthcare professionals and patients. Digital certificates are used for authentication, a blockchain network secures professional information and EMR, and roles and permissions allow EMR access. This system protects healthcare data while enabling authorised parties to access and alter it for better treatment. Digital certificates and blockchain technology provide this strong, efficient solution to patient EMR management and healthcare data privacy, security, and integrity.

4. Experimental Results and Discussions:

This section evaluates the SECURE-BLOCK system via a practical experiment. This test used a Java-developed blockchain. In this experiment, the data string may include smart contracts like Ethereum. The SECURE-BLOCK system was evaluated based on hash generation and encryption/decryption times.

Hash Generation Time (HGT) measures system efficiency. HGT is the millisecond time between blockchain network hash generation start and finish. Equation (1) represents this measure.

$$HGT = AH - BH \qquad (1)$$

The term "BH" refers to the time in milliseconds that was recorded immediately before to the beginning of the hash creation, while the term "AH" refers to the time in milliseconds that was recorded immediately after the hash generation finishes. An examination of the differences and similarities between the timeframes required to generate hashes using the various algorithms used by the blockchain network is shown in Table 1. The method developed by Shynu et al. [16], the BSF-EHR algorithm developed by Abunadi et al. [17], the BBPM algorithm developed by Abunadi et al. [18], and the hash generation performance of the SECURE-BLOCK system are all included in this comparison.

Table 2: Hash generation time comparison

Number of Blocks	Shynu et al. [16]	Abu nadi et al. [17]	BBP M [18]	SECUR E-BLOCK
10	22	20	16	8
25	38	36	32	15
50	60	42	37	17
75	90	78	71	20
100	130	118	110	46

In contrast to alternative methodologies, the SECURE-BLOCK system hastens hash creation, as illustrated in Figure 3. This enhanced speed is attributed to the adoption of the lightweight HmacSHA256 hash generation technique within SECURE-BLOCK, which surpasses other algorithms in terms of velocity. It's observed that as the quantity of blocks in the system grows, so does the HGT (hash generation time).

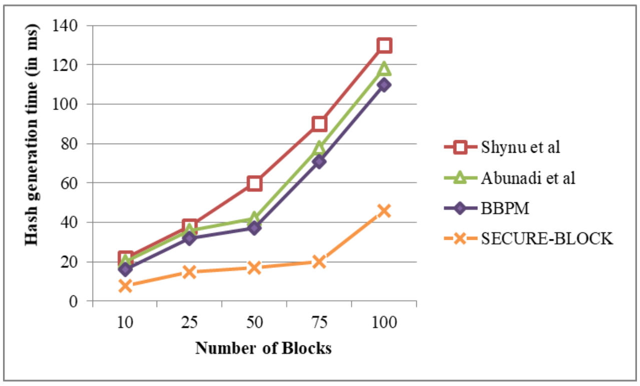

Fig 4: Hash generation time comparison

In addition, Table 2 provides a comparative comparison of the lengths of time required for the procedures of encryption and decryption. It compares and contrasts the cryptography approach suggested in this work (RSA+ - BC+) with two hybrid cryptographic pairings, namely AES - RSA and RSA - ECC, as mentioned in reference [19].

Table 2: Comparative assessment of the time durations (measured in seconds) required for encryption and decryption using the RSA+ - BC+ cryptographic approach against two prevalent hybrid cryptographic pairings

Data size (bits)	AES - RSA	RSA - ECC	RSA+ - BC+
48	10.2493	8.3266	3.166
64	14.7360	14.8445	3.084
80	21.8297	17.1274	3.141
96	34.0297	21.6842	3.106

According to the information in Table 2, the RSA+ - BC+ cryptographic approach allows users to securely encrypt data using potent hybrid algorithms while ensuring minimal time is needed for both encryption and decryption, regardless of the data size. Figure 5 graphically illustrates the comparative analysis of the encryption and decryption durations.

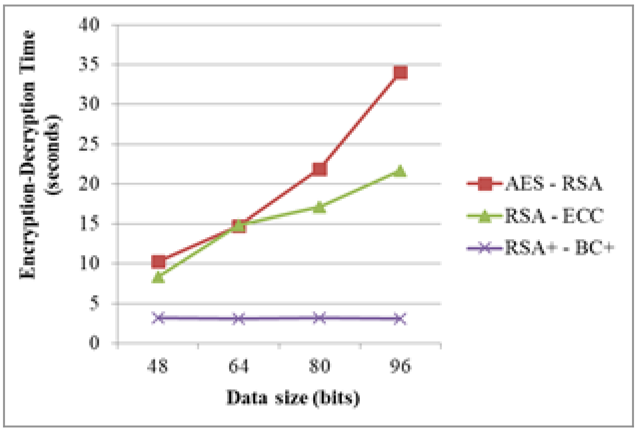

Fig 5:
Assessment of encryption and decryption durations against data size for the RSA+ - BC+ cryptographic approach compared with two other established hybrid cryptographic pairings

the SECURE-BLOCK system exhibits quicker hash generation compared to other algorithms, thanks to its implementation of the lightweight HmacSHA256 technique for hash creation. Additionally, the RSA+ - BC+ cryptographic method proposed in this system provides time-efficient encryption and decryption across various data sizes.

5. Conclusion

SECURE-BLOCK system introduces an innovative and effective method for safeguarding Electronic Medical Records (EMRs) through the integration of blockchain technology, cryptographic strategies, and access control mechanisms. This system significantly improves the security, confidentiality, and trust in EMR exchanges, providing a solution to the limitations of traditional centralized systems. Experimental data confirms that SECURE-BLOCK not only generates hashes more rapidly but also employs the RSA+ and BC+ cryptographic combination, which demonstrates quicker encryption and decryption times compared to other methods. These findings underscore the system's competence and potential in facilitating secure EMR sharing. Nevertheless, there's a need for additional exploration into its scalability, compatibility, and practical deployment. Future initiatives should aim to enhance the system’s performance, rigorously test it in actual healthcare scenarios, and navigate the regulatory and legal intricacies linked to employing blockchain in healthcare. Moreover, as blockchain and cryptographic technologies continue to evolve, they should be utilized to further reinforce the security and privacy of EMR systems, ultimately serving the interests of patients, healthcare providers, and the broader healthcare sector.

Author contributions

Mrs. Arti Krushnarao Walzade: Conceptualization, Methodology, Software, Field study, Data curation, Writing-Original draft preparation, Software, Validation., Field study. **Dr. Pradnya A Vikhar:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Pai, M. M., Ganiga, R., Pai, R. M., & Sinha, R. K. (2021). Standard electronic health record (EHR) framework for the Indian healthcare system. Health Services and Outcomes Research Methodology, 21(3), 339-362.

[2] Almaghrabi, N. S., & Bugis, B. A. (2022). Patient Confidentiality of Electronic Health Records: A Recent Review of the Saudi Literature. Dr. Sulaiman Al Habib Medical Journal, 4(3), 126-135.

[3] Ahmad, R. W., Salah, K., Jayaraman, R., Yaqoob, I., Ellahham, S., & Omar, M. (2021). The role of blockchain technology in telehealth and telemedicine. International journal of medical informatics, 148, 104399.

[4] Yaqoob, I., Salah, K., Jayaraman, R., & Al-Hammadi, Y. (2021). Blockchain for healthcare data management: opportunities, challenges, and future recommendations. Neural Computing and Applications, 1-16.

[5] Villarreal, E. R. D., García-Alonso, J., Moguel, E., & Alegría, J. A. H. (2023). Blockchain for Healthcare Management Systems: A Survey on Interoperability and Security. IEEE Access, 11, 5629-5652.

[6] Zou, R., Lv, X., & Zhao, J. (2021). SPChain: Blockchain-based medical data sharing and privacy-preserving eHealth system. Information Processing & Management, 58(4), 102604.

[7] Boumezbeur, I., & Zarour, K. (2021). Blockchain-Based Electronic Health Records Sharing Scheme with Data Privacy Verifiable. Applied Medical Informatics, 43(4), 124-135.

[8] Shen, B., Guo, J., & Yang, Y. (2019). MedChain: Efficient healthcare data sharing via blockchain. Applied sciences, 9(6), 1207.

[9] Niu, S., Chen, L., Wang, J., & Yu, F. (2019). Electronic health record sharing scheme with searchable attribute-based encryption on the blockchain. IEEE Access, 8, 7195-7204.

[10] Liu, J., Li, X., Ye, L., Zhang, H., Du, X., & Guizani, M. (2018, December). BPDS: A blockchain-based privacy-preserving data sharing for electronic medical records. In 2018 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

[11] Nguyen, D. C., Pathirana, P. N., Ding, M., & Seneviratne, A. (2019). Blockchain for secure ehrs sharing of mobile cloud-based e-health systems. IEEE Access, 7, 66792-66806.

[12] Sun, Z., Han, D., Li, D., Wang, X., Chang, C. C., & Wu, Z. (2022). A blockchain-based secure storage scheme for medical information. EURASIP Journal on Wireless Communications and Networking, 2022(1), 40.

[13] Mhamdi, H., Ayadi, M., Ksibi, A., Al-Rasheed, A., Soufiene, B. O., & Hedi, S. (2022). SEMRAchain: A Secure Electronic Medical Record Based on Blockchain Technology. Electronics, 11(21), 3617.

[14] Ali, A., Masud, M., Chen, C., AlZain, M. A., & Ali, J. (2022). An Effective Blockchain-Based Secure Searchable Encryption System. Intelligent Automation & Soft Computing, 33(2).

[15] Hashim, F., Shuaib, K., & Sallabi, F. (2021). Medshard: Electronic health record sharing using blockchain sharding. Sustainability, 13(11), 5889.

[16] P. G. Shynu, V. G. Menon, R. L. Kumar, S. Kadry, and Y. Nam, “Blockchain-based secure healthcare application for diabetic-cardio disease prediction in fog computing,” IEEE Access, vol. 9, pp. 45706–45720, 2021.

[17] Abunadi and R. L. Kumar, “BSF-EHR: blockchain security framework for electronic health records of patients,” Sensors, vol. 21, no. 8, Article ID 2865, 2021.

[18] Abunadi, I., & Kumar, R. L. (2021). Blockchain and business process management in health care, especially for covid-19 cases. Security and Communication Networks, 2021.

[19] Subedar, Z., & Araballi, A. (2020). Hybrid cryptography: Performance analysis of various cryptographic combinations for secure communication. International Journal of Mathematical Sciences and Computing (IJMSC), 6(4), 35-41.