Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7923075

SECURED ELECTRONIC CLINICAL UNIFIED RECORD EXCHANGE USING BLOCKCHAIN LEDGER AND OPTIMIZED CRYPTOGRAPHY

Walzade Arti Krushnarao and Dr. Sharanbasappa Gandage
Department of Computer Science and Engineering
Dr. A. P. J. Abdul Kalam University, Indore (M. P.) – 452010
Corresponding Author Email : artips15@gmail.com

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+) algorithmis 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. Experimental results show that SECURE-BLOCK generates hashes quickly, taking only 46 milliseconds for 100 blocks. RSA+ with BC+ cryptography used in SECURE-BLOCK also outperforms other cryptography combinations, with encryption and decryption times of 3.106 seconds for 96-bit data. These findings highlight the efficiency and effectiveness of SECURE-BLOCK for secure EMR 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 sharingof 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.

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 trustworthyusing blockchain.

Smart contracts, self-executing contracts that run on the blockchain, ensuring that only the right people can access and change EMRs. Smart contracts can be programmed to have specific rules and conditions that must be met before any transaction can be done with the EMRs. It makes sure that only authorized people can access and change the information.

TheSECURE-BLOCK system incorporates digital certificates issued by a Certification Authority (CA) to verify the identity of entities making transactions on the blockchainto enhance trust and security. These digital certificates serve as digital proof of identity, allowing entities to participate in the system security and ensuring that only legitimate entities can access and modify EMRs.

In addition to blockchain technology, theSECURE-BLOCK system employs cryptographic techniques to protect the confidentiality and integrity of EMRs. Encryption algorithms encrypt the EMR data before storing it on the blockchain, ensuring that only authorized users with the appropriate decryption keys can access it. In addition, hash functions generate a unique digital fingerprint of the EMR data, which is stored on the blockchain as a hash value, ensuring data integrity and tamper-proofing.

Furthermore, theSECURE-BLOCK system incorporates an access control and authorization mechanism that allows only authorized entities to access and modifies EMRs. Access controls are implemented through smart contracts, which enforce predefined rules and conditions for accessing EMRs. These rules can include criteria such as the user's role, the purpose of access, and the patient's consent. Authorization is granted based on the digital certificates issued by the Certification Authority, ensuring that only entities with valid and verified identities can access EMRs.

The following is how the paper is set up: Section 2 reviews existing literature and research on EMR sharing systems in the eHealth domain. Section 3 explains the SECURE-BLOCK system, including using blockchain technology, cryptographic techniques, access controls, and authorization mechanisms. Furthermore, it describes the implementation details of the SECURE-BLOCK system, including the use of the Ethereum blockchain, smart contracts, and digital certificates. Experiments are presented in Section 4, and a discussion of the findings. Finally, section 5 provides the conclusions and suggestions for future research.

2. Related work

Protecting the sharing of EMRs is essential for healthcare. Various recent solutions have been suggested and implemented to tackle this problem. This literature review aims to provide an overview and assessment of the existing solutions concerning the security of electronic medical record sharing.

Zou et al. [6] proposed a system called SPChain that aims to overcome the challenges of sharing electronic medical records (EMRs) in permissioned and public blockchain-based eHealth systems. SPChain uses special keyblocks and microblocks to help patients store their EMRs in a way that allows for efficient retrieval. It also includes a reputation system encouraging medical institutions to participate in the SPChain system. Furthermore, to protect patient privacy, SPChain uses proxy re-encryption schemes for sharing medical data. Finally, the performance and security of the system are evaluated using the real-world distribution of miners, showing that it has high throughput (220 TPS) and low storage overhead.

Boumezbeur and Zarour [7] presented an architecture for sharing electronic health records that integrates encryption, access control, and a storage mechanism using cloud and blockchain technologies. The health records are encrypted and stored in the cloud server, while the blockchain stores traceable log information and encryption keys. This approach guarantees integrity, privacy, and confidentiality, effectively protecting shared electronic health records.

Shen et al. [8] presented a solution called MedChain that addresses the efficiency challenges of current blockchain-based healthcare data-sharing methods. MedChain combines blockchain, digest chain, and structured peer-to-peer (P2P) network techniques to efficiently share continuously generated data streams from sensors and other monitoring devices while supporting flexible metadata changes. Evaluation results show that MedChain achieves higher efficiency and meets the security requirements for sharing healthcare data.

Niu et al. [9] presented a permissioned blockchain-based scheme for medical data sharing that uses ciphertext-based attribute encryption to protect data confidentiality and access control. The scheme utilizes a polynomial equation to enable arbitrary connection of keywords while preserving patient identity privacy. Additionally, the proposed scheme provides

keyword-indistinguishability against adaptive chosen keyword attacks and demonstrates high retrieval efficiency.

Liu et al. [10] introduced the BPDS system, which utilizes blockchain to enable privacy-preserving data sharing of electronic medical records (EMRs). The system addresses the challenges of sharing sensitive health data by safeguarding patient identity privacy and ensuring secure data sharing. The proposed system is evaluated in the context of improving the quality of healthcare service and reducing medical costs.

Nguyen et al. [11] proposed a novel framework for the secure sharing of Electronic Health Records (EHRs) using blockchain and the decentralized interplanetary file system (IPFS) on a mobile cloud platform. The system utilizes smart contracts for access control, ensuring reliable and secure data exchanges on mobile clouds while protecting sensitive health information from potential threats. The advantages of the proposed system include low operational cost, high flexibility, and availability of EHRs. The authors implemented a prototype using the Ethereum blockchain in a real data-sharing scenario on a mobile app with Amazon cloud computing. They demonstrated empirical results that showed the effectiveness of their proposal. The system evaluation and security analysis also showed performance improvements in lightweight access control design, minimum network latency with high security, and data privacy levels compared to existing data-sharing models.

Sun et al. [12] proposed a healthcare data security system that utilizes Hyperledger Fabric and the Attribute-Based Access Control (ABAC) framework. The proposed scheme employs attribute-based access control for dynamic and fine-grained access to medical information. It uses smart contracts to store the information in the blockchain for secure and tamper-proof storage. The system also incorporates IPFS technology to alleviate the storage pressure of the blockchain. The advantages of the proposed system include secure storage and integrity of medical information, high throughput when accessing medical information, and the ability to handle the increasing digitalization of healthcare data. Experiments were conducted to validate the effectiveness of the proposed scheme.

Mhamdi et al. [13] proposed SEMRAchain, a system based on RBAC and ABAC combined with smart contracts for decentralized, fine-grained, and dynamic access control management for EMR. The system utilizes blockchain technology as a secure distributed ledger to improve the security, reliability, and dependability of the EMR sharing process, overcoming challenges associated with centralized access control methods in existing medical systems.

Ali et al. [14] proposed a blockchain-based system for secure search and keywords-based access to PHR using homomorphic encryption. The system integrates blockchain as a distributed database with a trust chain, providing a secure key revocation mechanism and policy updates. It improves security, efficiency, and effectiveness in sharing digital healthcare data in the medical IoT.

Hashim et al. [15] investigate the use of blockchain for EHR sharing while maintaining privacy and security. The proposed transaction-based sharding technique addresses scalability challenges, achieving higher throughput, reduced consensus latency, and increased number of appointments processed compared to standard-based healthcare blockchain techniques. In addition, the technique eliminates cross-shared communication overhead, providing a more efficient approach to EHR sharing in the healthcare system.

Overall, this section comprehensively reviews various existing systems and approaches for sharing electronic medical records (EMRs) in the eHealth domain. The SECURE-BLOCK system with improved hybrid cryptography using both the RSA+ algorithm and the BC+ algorithm is more efficient than previous works in several ways:

- **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 designed to address the limitations and challenges of traditional EMR sharing systems. This section will outline the system architecture of the SECURE-BLOCK system. The system architecture consists of eight main components: 1) Admin 2) Blockchain 3) Authentication server 4) Certification authority 5) Doctor 6) Nurse 7) Pharmacist 8) Patient. Figure 1 shows the visual representation of the system architecture of the SECURE-BLOCK system.

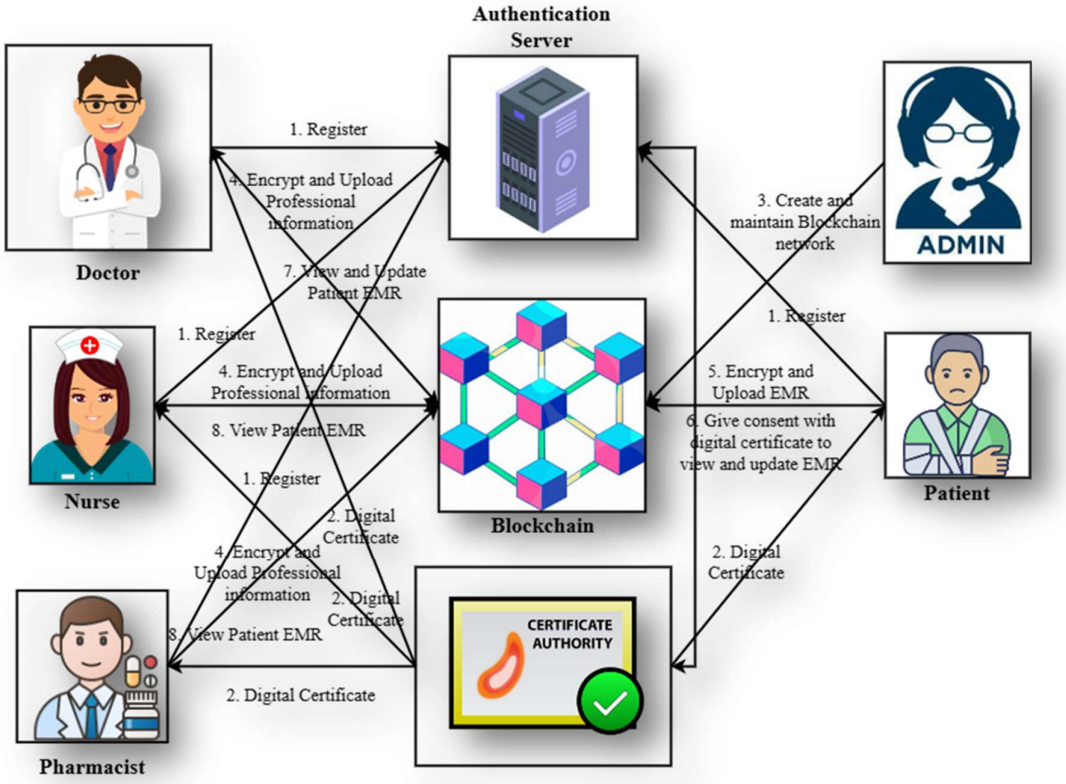

Figure 1: Architecture of the SECURE-BLOCK system

The concept described in Figure 1 involves a system for registering and authenticating doctors, nurses, pharmacists, and patients in a healthcare setting, issuing digital certificates for authentication, creating and maintaining a blockchain network, and enabling secure encryption and uploading of professional information and electronic medical records (EMR).

1. **Doctor, Nurse, Pharmacist, and Patient Registration with Authentication Server:** The system begins with registering doctors, nurses, pharmacists, and patients with an authentication server. This registration process involves providing necessary personal and professional information, such as name, contact details, and credentials, which are verified by the authentication server to ensure the authenticity of the users.
2. **Issuance of Digital Certificates:** Upon successful registration, a certification authority issues digital certificates to the registered doctors, nurses, pharmacists, and patients. These digital certificates serve as proof of identity and are used for encryption and decryption purposes in the system.
3. **Blockchain Network Creation and Maintenance:** A network administrator creates and maintains a blockchain network, which is a decentralized and distributed ledger that securely records transactions and information across multiple nodes. The blockchain network is a secure and transparent platform for storing and managing encrypted professional information and EMR.
4. **Encryption and Uploading of Professional Information:** Doctors, nurses, and pharmacists can encrypt and upload their professional information, such as

qualifications, certifications, and work history, to the blockchain network. This information is securely stored in the blockchain and can be accessed only by authorized parties.

5. **Encryption and Uploading of EMR:** Patients can encrypt and upload their electronic medical records (EMR) to the blockchain network. EMR typically includes a patient's medical history, diagnoses, treatments, and other relevant health information. Therefore, the EMR is encrypted to ensure its privacy and security.
6. **The patient gives consent with digital certificate to view and update EMR: The p**atient specifies their consent preferences regarding who can view and update their EMR. The patient may provide explicit consent for specific healthcare providers, such as doctors, nurses and pharmacists, to access their EMR using their digital certificate.
7. **Doctor, Nurse and Pharmacist Access to Patient EMR:** Doctors registered in the system can view and update the EMR of their patients. It allows doctors to securely access and updates their patients' medical records to provide appropriate healthcare services. However, nurses and pharmacists registered in the system can view patient EMRs butcannot update the information. It ensures that nurses and pharmacists have access to relevant patient information for quality care, but they do not have the authority to change the EMR.

Overall, this concept establishes a secure and transparent system for registering and authenticating healthcare professionals and patients, issuing digital certificates for authentication, creating and maintaining a blockchain network for secure storage of professional information and EMR, and enabling authorized access to patient EMR based on roles and permissions. It ensures the healthcare data's privacy, security, and integrity while allowing authorized users to access and update the necessary information for quality healthcare services.

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

The section describes the SECURE-BLOCK system for registering doctors, nurses, pharmacists, and patients in a healthcare setting, using an authentication server to verify their identities. The registration process involves providing personal information such as name, email id, and role (i.e., doctor, nurse, pharmacist, or patient), which is then verified by the authentication server to ensure that the users are authentic.

A HmacSHA256 hash value is generated and provided to the users during the registration process. This hash value is based on the email id provided during registration. The users can later use this hash value to login and access the system.

In more detail, the registration process would typically involve the following steps:

1. **Personal Information Collection:** The doctors, nurses, pharmacists, and patients would provide their personal information, such as name and email id, to the system during the registration process. They would also specify their roles, i.e., whether they register as a doctor, nurse, pharmacist, or patient.
2. **Authentication Server Verification:** The authentication server would verify the provided personal information to ensure that it is valid and authentic. It may involve checking the email id against existing records, validating the format of the email id, and ensuring that the roles specified are valid in the system.

3. **HmacSHA256 Hash Generation:** Once the personal information is verified, the authentication server will generate a HmacSHA256 hash value based on the email id provided during registration. HmacSHA256 is a cryptographic hash function that produces a fixed-size hash value, which can be used for secure authentication.
4. **Hash Value Provision:** The generated hash value is then provided to the users, i.e., the doctors, nurses, pharmacists, and patients, as part of the registration process. The users would typically receive this hash value through a secure channel, such as email.
5. **Login and Access:** In subsequent logins, the users would provide their email id and the hash value generated during registration to the system. The system would then verify the hash value against the stored hash value for the corresponding email id in its records. If the hash values match, the users will be granted access to the system, allowing them to log in and use it according to their specified roles.

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, secretKey

Output : OuterHash

// Step 1: Pad the secret key if necessary

Step 1 : If secretKey.length> 64, Then

Step 2 : secretKey = SHA256(secretKey)

Step 3 : If secretKey.length< 64, Then

Step 4 : secretKey = padKey(secretKey)

// Step 2: Generate the inner and outer padding keys

Step 5 : InnerPaddingKey = xor(secretKey, 0x36)

Step 6 : OuterPaddingKey = xor(secretKey, 0x5C)

// Step 3: Calculate the inner hash

Step 7 : InnerHash = SHA256(concatenate(InnerPaddingKey, Plaintext))

// Step 4: Calculate the outer hash

Step 8 : OuterHash = SHA256(concatenate(OuterPaddingKey, InnerHash))

// Step 5: Return the final hash

Step 9 : return OuterHash

In Algorithm 2, the term 'secretKey' denotes a key used in the HMACSHA256 algorithm for generating a secure hash of a plaintext. The key is to bind the Plaintext to the hash, ensuring its authenticity and integrity. It serves as a shared secret between the sender and recipient of the Plaintext and is combined with the Plaintext to produce the hash. Hence, maintaining the confidentiality and secure communication of the secret key between the sender and recipient is crucial for preserving the security of the hash. Within the context of the HMACSHA256 algorithm, the secret key generates two padding keys, which are subsequently employed in calculating the inner and outer hashes that comprise the final HMACSHA256 hash.

Algorithm 2 incorporates a 'SHA256' function that computes the SHA-256 hash of a given input, a 'concatenate' function that merges two strings, a 'xor' function that executes a bitwise exclusive-or operation, and a 'padKey' function that pads the secret key with zeros or other specified padding values to achieve a length of 64 bytes if necessary. Furthermore, the 'HMACSHA256' function mandates the provision of 'plaintext' and 'secret_Key' as inputs and generates the resulting HMACSHA256 hash as output.

In Algorithm 2, the hexadecimal notations' 0x36' and '0x5C' represent the binary values '0011 0110' and '0101 1100', respectively. These values are utilized as inner and outer padding keys within the HMACSHA256 algorithm. The inner padding key is obtained by applying a bitwise exclusive-or ('xor') operation between the secret key and '0x36'. Similarly, the outer padding key is generated by performing a bitwise exclusive-or operation between the secret key and '0x5C'. These inner and outer padding keys are then employed to calculate the inner and outer hashes, which ultimately constitute the final HMACSHA256 hash. Utilizing these padding keys enhances the security of the HMACSHA256 algorithm by incorporating the Plaintext and the secret key into the computation of the final hash.

Overall, the registration process with the authentication server and the provision of a hash value based on email id adds a layer of security to the system, helping to ensure that only authorized users can register and access the system.

3.1.2 Issuance of Digital Certificates:

After successful registration, a certification authority (CA) generates digital certificates for the registered users. The digital certificates serve as proof of identity for the registered users in the system. In addition, they contain the user's email ID and cryptographic keys (RSA+ public key, RSA+ private key, and BC+ secret key) used for encryption and decryption purposes. The RSA+ public key and BC+ secret key encrypt messages intended for the user. In contrast, the RSA+ private key and BC+ secret key are used for decrypting messages and performing other authorized actions in the system.

The hybrid encryption technique capitalizes on the respective strengths of symmetric and asymmetric encryption, striking a balance between speed and security. Figure 2 illustrates the proposed hybrid cryptography architecture, showcasing the improved approach's implementation.

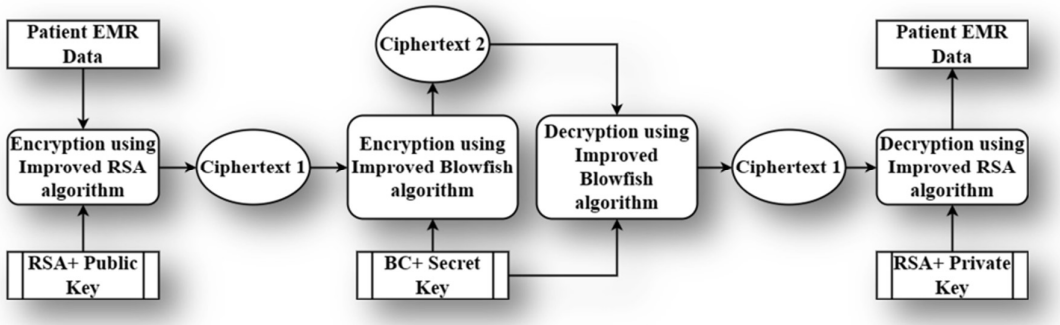

Figure 2: Improved hybrid cryptography

The generation, encryption, and decryption processes for RSA+ public key, RSA+ private key, and BC+ secret key are complex cryptographic operations that provide security and confidentiality in the system. Therefore, proper management and protection of these keys are crucial to ensure the integrity and confidentiality of user data and communications in the system. This section explains in detail the RSA+ public key, RSA+ private key and BC+ secret key generation, encryption and decryption.

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

The widely employed RSA encryption algorithm is commonly utilized for secure data transmission. However, to enhance its security, one approach is to augment the key size. Nevertheless, this method also results in increased encryption and decryption times. The RSA+ algorithm addresses this challenge by breaking data into smaller segments and incorporating two additional prime numbers into the traditional RSA algorithm. As a result, it results in a more robust encryption process with faster encryption and decryption times, rendering it, particularly fitting for utilization in blockchain systems.

The RSA encryption algorithm has limitations, such as slow encryption and decryption processes and limited key sizes, which can render it susceptible to intruder attacks. The proposed EMR sharing system employs the RSA+ algorithm, which enhances the complexity of the encryption process and accelerates encryption and decryption timesto overcome these shortcomings.

The RSA+ algorithm is imperative for enhancing the security and efficiency of data transmission in systems like blockchain, where secure storage and transmission of large

volumes of data are essential. The advantages of utilizing the RSA+ algorithm include heightened security due to larger key sizes and faster encryption and decryption times, making it particularly suitable for implementation in large-scale systems. Algorithm 3 outlines the asymmetric key cryptography based on the RSA+ algorithm.

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

Input : Plaintext

Output : EncryptedPlaintext_1 (after encryption), Plaintext (after decryption)

/* Improved RSA Public Key and Private Key generation */

Step 1 : X, Y, Z, W = Choose any four large prime numbers

Step 2 : M = X * Y * Z * W

Step 3 : N = X * Y

Step 4 : S = (X-1) * (Y-1) * (Z-1) * (W-1)

Step 5 : PK = Choose PK such that GCD(PK, S) = 1

Step 6 : SK = Choose SK such that SK = PK^{-1} * mod(S)

Step 7 : PublicKey = Pair of PK and M

Step 8 : PrivateKey = Pair of SK and N

/* Encryption */

Step 9 : PK, M = Extract PK and M from PublicKey

Step 10 : T[] = Split Plaintext into n chunks

Step 11 : EncryptedPlaintext_1 = ""

Step 12 : for i = 0 to n

Step 13 : BA[] = Convert T[i] to byte array

Step 14 : T2 = BA^{PK} mod (M)

Step 15 : EncryptedPlaintext_1 = EncryptedPlaintext_1+""+T2

Step 16 : end for

/* Decryption */

Step 17 : SK, N = Extract SK and N from PrivateKey

Step 18 : T[] = Split EncryptedPlaintext_1 into n chunks

Step 19 : Plaintext = ""

Step 20 : for i = 0 to n

Step 21 : ET = T[i]

Step 22 : BA[] = ET^{SK} mod (N)

Step 23 : T1 = Convert BA to string

Step 24 : Plaintext = Plaintext + "" + T1

Step 25 : end for

Algorithm 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 is a symmetric key block cipher algorithm that operates on 64-bit blocks of data with a maximum key length of 448 bits. However, as with any cryptographic algorithm, the need to continuously improve its security has arisen due to the evolving landscape of potential attacks. An advanced version of Blowfish cryptography known as Improved Blowfish cryptography (BC+) has been proposedto address this. BC+ involves modifications or variations to the standard Blowfish algorithm, such as adding extra rounds or changing block or key sizes, to enhance its security. The key advantage of BC+ is its increased level of security compared to the standard Blowfish algorithm, as it is designed to better protect against new or unknown attacks. In addition, it makes BC+ suitable for various applications. The working process of BC+ is discussed in Algorithm 4.

Algorithm 4: Improved Blowfish Cryptography (BC+)

Input : Input_Plain_text

Output : Output_Cipher_text, Output_Decrypted_Plain_text

/* Encryption */

Step 1 : Initialize the key_schedule using the provided key, which can be either a string or a binary key

Step 2 : Divide the Input_Plain_text into blocks of N bytes (N * 8 bits), where N is the new block_size

Step 3 : For each block, perform an XOR operation with the P_box and then separate it into two halves (left and right)

Step 4 : Conduct M rounds of substitution and permutation on the two halves using the key_schedule, where M represents the number of additional rounds

Step 5 : Swap the left and right halves, and XOR the result with the P_box

Step 6 : Repeat steps 4 and 5 for every block of Input_Plain_text

Step 7 : Assemble the Cipher_text blocks and return the final result as Output_Cipher_text

/* Decryption */

Step 8 : Initialize the key_schedule using the provided key, which can be either a string or a binary key

Step 9 : Divide the Output_Cipher_text into blocks of N bytes (N * 8 bits), where N is the block_size

Step 10 : For each block, perform an XOR operation with the P_box and then separate it into two halves (left and right)

Step 11 : Conduct 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

Step 12 : Swap the left and right halves, and XOR the result with the P_box

Step 13 : Repeat steps 4 and 5 for every block of Output_Cipher_text

Step 14 : Assemble the Decrypted_Plain_text blocks and return the final result as Output_Decrypted_Plain_text

The advantage of the Enhanced Blowfish Cryptography (BC+) algorithm, as described in Algorithm 4, lies in its heightened security. Including additional rounds amplifies the number

of transformations the Plaintext undergoes, making it significantly more challenging for potential attackers to detect patterns or uncover the key. The algorithm's security is further enhanced with longer keys, as the number of possible keys an attacker would need to try increases exponentially. Modifying the block size allows the algorithm to operate on larger data units, which can result in improved efficiency and potentially higher security. Overall, increasing the number of rounds and using a larger block size can significantly enhance the overall security level of the algorithm.

3.1.3 The Role of Blockchain in SECURE-BLOCK system:

In the SECURE-BLOCK system, the blockchain plays a crucial role in ensuring the security and integrity of the EMR. The main responsibilities of the blockchain in this system are:

1. **Decentralization:** Blockchain allows for a decentralized and distributed network, eliminating the need for a central authority to control and manage EMR data, ensuring transparency and trust in the system.
2. **Security:** Blockchain employs an improved hybrid cryptographic technique to secure EMR data, protecting it from unauthorized access, tampering, and data breaches, ensuring the confidentiality and integrity of patient information.
3. **Transparency:** Blockchain provides a transparent and auditable record of all transactions and changes to EMR data, allowing for traceability and accountability.
4. **Consensus:** Blockchain uses a consensus algorithm, such as Proof-of-Work (PoW), to ensure that all transactions are verified by multiple nodes in the network, ensuring transaction integrity and preventing fraud.
5. **Smart Contracts:** Blockchain utilizes smart contracts, which are self-executing agreements, to enforce access control mechanisms, permissions, and consent preferences of patients, ensuring that only authorized parties can access and update EMR data.
6. **Digital Certificates:** Blockchain employs certification authority to issue digital certificates to healthcare professionals and patients, serving as proof of identity and for encryption and decryption purposes.
7. **Immutable Ledger:** Blockchain maintains an immutable ledger of all transactions, changes, and updates to EMR data, providing a tamper-proof history of events.
8. **Efficiency:** Blockchain allows for efficient and secure sharing of EMR data among authorized parties, reducing the need for redundant data entry and ensuring data accuracy and consistency.
9. **Patient Empowerment:** Blockchain enables patients to control their EMR data by giving explicit consent using digital certificates. It allows them to specify who can access and update their information, ensuring patient privacy and autonomy.
10. **Trustworthiness:** Blockchain enhances trust among stakeholders, including patients, healthcare professionals, and others, by providing a secure, transparent, and auditable system for EMR sharing, ensuring data privacy, security, and integrity.
11. **Disaster Recovery:** Blockchain stores EMR data across a distributed network of nodes, ensuring data redundancy and resilience to single-point failures, providing enhanced disaster recovery capabilities.

12. **Trust and Reputation:** Blockchain enables healthcare entities to establish trust and reputation based on their verified digital certificates and transaction history. It promotes trust among stakeholders and reduces the risk of fraudulent activities.

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) (Professional details of Doctor, Nurse and Pharmacist, EMR details of the patient), blockchain
Output	**:**	An event that announces that a new user detail has been added to the blockchain
Step 1	**:**	Previous_Block_Hash← 0
Step 2	**:**	Size_of_the_Blockchain← 0
Step 3	**:**	For each block B in the blockchain
Step 4	**:**	Size_of_the_Blockchain++
Step 5	**:**	Previous_Block_Hash← Hash of B
Step 6	**:**	End For
Step 7	**:**	Block_number←Size_of_the_Blockchain + 1
Step 8	**:**	Time-stamp ← Get current time
Step 9	**:**	Nonce ← Generate a random number
Step 10	**:**	Hash ← Generate hash for Block_number, Timestamp, Nonce and EUD // **Algorithm 2**
Step 11	**:**	If Size_of_the_Blockchain == 0 Then
Step 12	**:**	Genesis_Block← Block (Block_number, Timestamp, Nonce, Hash, Previous_Block_Hash)
Step 13	**:**	Upload Genesis_Block to Blockchain
Step 14	**:**	Else
Step 15	**:**	Succeeding_Block← Block (Block_number, Timestamp, Nonce, Hash, Previous_Block_Hash)
Step 16	**:**	Upload Succeeding_Block to Blockchain
Step 17	**:**	End If

Algorithm 5 outlines the process of adding encrypted user details (EUD), such as the Professional details of doctors, nurses and pharmacists, and EMR details of patients to a blockchain. The algorithm begins by initializing variables such as Previous_Block_Hash and Size_of_the_Blockchain to keep track of the existing blockchain. It then iterates through each block in the blockchain, incrementing the Size_of_the_Blockchain variable and updating the Previous_Block_Hash with the current block's hash. Finally, the algorithm generates a hash for the new block by combining the Block_number, Time-stamp, Nonce, and EUD. If the blockchain is empty (Size_of_the_Blockchain equals 0), a Genesis_Block is created with the relevant information and uploaded to the blockchain. Otherwise, a Succeeding_Block is created and uploaded. This process ensures that the user details are securely added to the blockchain, and the event of new user details being added is announced. By leveraging blockchain technology, this algorithm enhances the security and transparency of the SECURE-BLOCK system for managing 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 containing consent preferences and access permissions, doctor, nurse, and pharmacist credentials for authentication, Patient EMR data
Output	**:**	Authorized access to view and/or update patient EMR based on consent preferences and access permissions.
Step 1	**:**	The patient provides explicit consent preferences for who can view and update their EMR using their digital certificate.
Step 2	**:**	Doctor, nurse, and pharmacist credentials are authenticated to ensure they are registered in the system and have appropriate access permissions.
Step 3	**:**	The patient's digital certificate is verified to ensure it is valid and matches the patient's consent preferences.
Step 4	**:**	If the doctor's credentials are authenticated, and the digital certificate indicates that the doctor has access permissions, the doctor is granted access to view and update the patient's EMR.

Step 5 : Suppose the nurse's credentials are authenticated and the digital certificate indicates that the nurse has access permissions. In that case, the nurse can view the patient's EMR but not update the information.

Step 6 : Suppose the pharmacist's credentials are authenticated and the digital certificate indicates that the pharmacist has access permissions. In that case, the pharmacist can view the patient's EMR but not update the information.

Step 7 : If the credentials of any user are not authenticated, or the digital certificate does not indicate access permissions, access to patient EMR is denied.

Step 8 : Any attempted changes to the patient's EMR are logged for auditing purposes.

Step 9 : The algorithm continues to monitor access and updates to the patient's EMR to ensure ongoing compliance with consent preferences and access permissions.

Step 10 : The algorithm terminates when the authorized users complete their tasks or log out of the system.

Algorithm 6 is a digital certificate-based access control algorithm designed for patient EMR. It takes inputs such as the patient's digital certificate containing consent preferences and access permissions, credentials of doctors, nurses, and pharmacists for authentication, and the patient's EMR data. The algorithm's outputallows access to view and/or update the patient's EMR based on consent preferences and access permissions.

The algorithm starts with the patient providing explicit consent preferences for who can view and update their EMR using their digital certificate. The credentials of doctors, nurses, and pharmacists are then authenticated to ensure they are registered in the system and have appropriate access permissions. Finally, the patient's digital certificate is verified to ensure it is valid and matches the patient's consent preferences.

If the doctor's credentials are authenticated, and the digital certificate indicates that the doctor has access permissions, the doctor is granted access to view and update the patient's EMR. Likewise, suppose the nurse's credentials are authenticated and the digital certificate indicates that the nurse has access permissions. In that case, the nurse can view the patient's EMR but not update the information. Similarly, suppose the pharmacist's credentials are authenticated and the digital certificate indicates that the pharmacist has access permissions. In that case, the pharmacist can view the patient's EMR but not update the information.

If the credentials of any user are not authenticated, or the digital certificate does not indicate access permissions, access to the patient's EMR is denied. Any attempted changes to the patient's EMR are logged for auditing purposes. The algorithm continues to monitor access and updates to the patient's EMR to ensure ongoing compliance with consent preferences and access permissions. The algorithm terminates when the authorized users complete their tasks or log out of the system.

Overall, the concept described in this section establishes a secure and transparent system for registering and authenticating healthcare professionals and patients, issuing digital

certificates for authentication, creating and maintaining a blockchain network for secure storage of professional information and EMR, and enabling authorized access to patient EMR based on roles and permissions. This system ensures healthcare data privacy, security, and integrity while allowing authorized users to access and update the necessary information for quality healthcare services. Through digital certificates and blockchain technology, this concept provides a robust and efficient solution for managing patient EMR and promoting data privacy, security, and integrity in healthcare settings.

4. Experimental Results and Discussions

The efficiency of the SECURE-BLOCK system was examined in this section. A blockchain built with Java was utilized in this experiment. This experiment has a data string containing anything you could think of, containing smart contracts in the Ethereum style. The performance of the SECURE-BLOCK system was evaluated by estimating the time taken for hash generation and the time taken for encryption and decryption, which served as the primary evaluation criterion.

The time duration (in milliseconds) between the before and after hash generation in the blockchain network, denoted as Hash Generation Time (HGT), is defined as follows in Eq. (1):

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

BH represents the current time in milliseconds before hash generation, and AH represents the current time in milliseconds after hash generation. In Table 1, a comparison is made between the hash generation time of various algorithms used in the blockchain network, including Shynu et al.'s algorithm [16], Abunadi et al.'s BSF-EHR algorithm [17], Abunadi et al.'s BBPM algorithm [18], and the SECURE-BLOCK system.

Table 2: Hash generation time comparison

Number of Blocks	Shynu et al. [16]	Abunadi et al. [17]	BBPM [18]	SECURE-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

Compared to other approaches, the SECURE-BLOCK solution accelerates the generation of hashes, as depicted in Figure 3. It is due to the utilization of the HmacSHA256 lightweight hash generation technique in the SECURE-BLOCK system, which outperforms other algorithms in speed. As the number of blocks increases, the HGT (hash generation time) also rises.

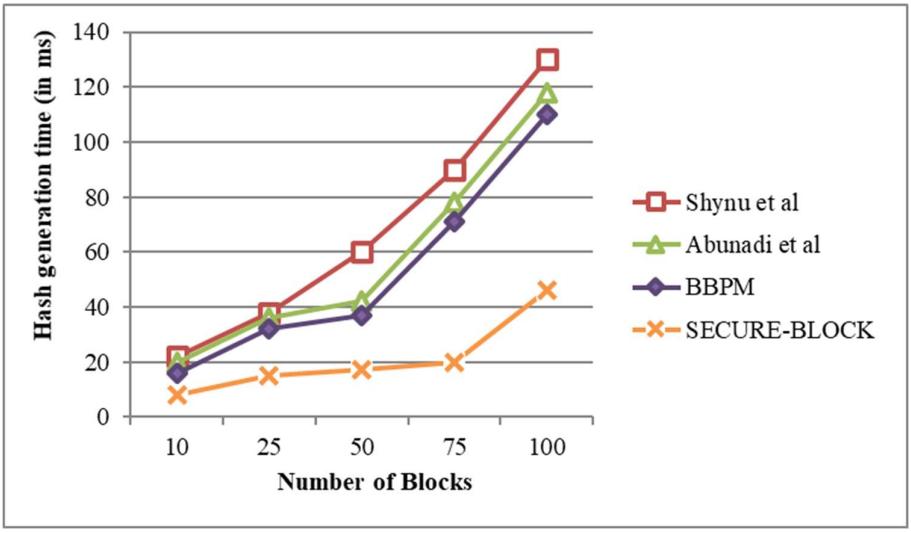

Figure 4: Hash generation time comparison

Moreover, Table 2 compares the times required for encryption and decryption using two hybrid cryptography combinations, AES - RSA and RSA - ECC, as documented in reference [19], with the proposed cryptography technique (RSA+ - BC+).

Table 2: Comparative analysis of encryption and decryption times (in seconds) for the RSA+ - BC+ cryptography technique with two existing hybrid cryptography combinations

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

As per the data presented in Table 2, the proposed RSA+ - BC+ cryptography technique enables users to encrypt data using robust hybrid algorithms with minimal time requirements for encryption and decryption across various data sizes. Figure 5 visually depicts the comparison between encryption and decryption times.

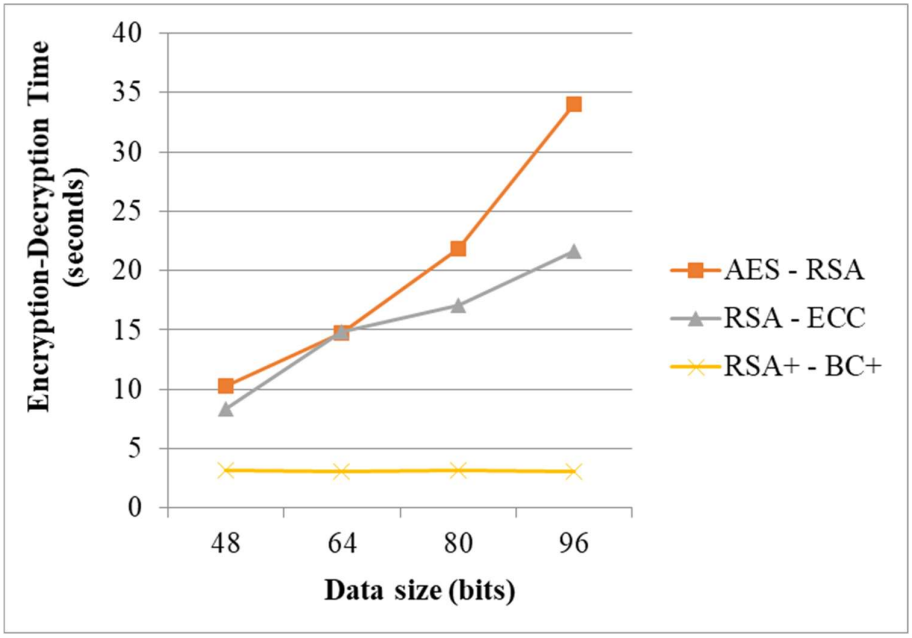

Figure 5: Comparison of data size versus encryption-decryption time for RSA+ - BC+ cryptography technique with two existing hybrid cryptography combinations

Overall, the SECURE-BLOCK system demonstrates faster hash generation times than other algorithms due to its utilization of the HmacSHA256 lightweight hash-generating technique. The proposed RSA+ - BC+ cryptography technique also offers efficient encryption and decryption times for different data sizes.

5. CONCLUSION

In conclusion, the proposed SECURE-BLOCK system presents a novel and robust approach to securing Electronic Medical Records (EMRs) through blockchain technology, cryptographic techniques, and access controls. The system enhances security, privacy, and trustworthiness in EMR sharing, addressing the challenges of existing centralized systems. The experimental results indicate that the SECURE-BLOCK system generates hashes faster and uses RSA+ with BC+ cryptography, which takes less time for encryption and decryption than other cryptography combinations. These results demonstrate the efficiency and effectiveness of SECURE-BLOCK in providing secure EMR sharing. However, further research and development are needed to explore the scalability, interoperability, and real-world implementation of the SECURE-BLOCK system. Future work could focus on refining the system's performance, conducting extensive testing in real-world healthcare settings, and addressing regulatory and legal challenges associated with blockchain technology in healthcare. Additionally, ongoing advancements in blockchain and cryptography could be leveraged to continually enhance the security and privacy of EMR systems, ultimately benefiting patients, healthcare professionals, and stakeholders in the healthcare industry.

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] I. 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.