BLOCKCHAIN IN HEALTHCARE TODAY

ORIGINAL CLINICAL RESEARCH

GAHBT: Genetic-Based Hashing Algorithm for Managing and Validating Health Data Integrity in Blockchain Technology

Fozia Hanif[1], Urooj Waheed[2], Rehan Shams[3], Aisha Shareef[1]

[1]Department of Mathematics, University of Karachi, Karachi, Pakistan; [2]Department of Computer Science, DHA Suffa University, Karachi, Pakistan; [3]Department of Telecommunication Engineering, Sir Syed University of Engineering and Technology, Karachi, Pakistan

Corresponding Author: Fozia Hanif, Email: ms_khans2011@hotmail.com

Keywords: blockchain, hashing, decentralized technology, genetics, health data distribution, security, surveillance

Abstract

A method for managing, securing, and validating health data distribution records using a genetic-based hashing algorithm in a decentralized environment is presented in this research report. The rationale for choosing blockchain is to secure the transaction of health data and protect these data from manipulated fraudulent movement and corruption by a contributor to the chain, or any individual. Our approach uses technology that provides an efficient surveillance measure, including transparency of records, immunity from fraud, and protection from tampering, as well as sustaining the order of data. For medical research, the results here provide a genetic-based hashing algorithm for data security, which has lower computational complexity, low space coverage, higher security and integrity, and a high avalanche effect. The simulation will show the validity, immunity, and integrity of the data record. The technique modified in this secure decentralized network is a cryptographic hashing algorithm for 512 bits. In this study, a genetic algorithm (GA) is used to generate a key that must be used in the encryption and decryption of medical data. A GA is a metaheuristic approach inspired by the laws of genetics; and it is generally used to generate high-quality solutions for complex problems. Applications of GAs are possible in medical fields, such as radiology, oncology, cardiology, endocrinology, surgery, oncology, and radiotherapy in healthcare management.

Submitted: 24 November 2022; Accepted: 17 January 2023; Published 03 February 2023

FEB Inc. blockchain technology, as applied to healthcare, offers secure (immutable) storage of private sensitive electronic data (e.g., patient diagnosis and medication) that can be shared among healthcare providers, institutions, and other disciplines.[1] The challenge is to update a patient's medical record securely, while maintaining access for those involved in care.[2,3]

This access to the complete medical history by all participants in the patient's care is critical to achieve successful long-term management of conditions, such as HIV, cancer, diabetes, and cardiovascular diseases, to name just a few. However, coordinating the process of sharing data when a patient moves from one medical institution to another in a different city (or country) becomes complex.[4]

By providing the required tool that secures data access among authorized individuals, while excluding hackers,[5] without relying on a single trustworthy node, blockchain technology assures data surveillance and access to sensitive and private data. It provides guidance on healthcare data for the patient, as well as healthcare providers,[6,7] which is safe and secure when stored in the form of blocks of data that are secured and linked cryptographically.[8,9]

This study proposes an immune and robust system for handling electronic medical records or data. A blockchain platform is used as an access control technique in which a modified cryptographic method is used to maintain surveillance, as well as integrity, in order to gain access control of the sensitive medical information for sharing among the other nodes.

The modification in this method uses a cryptographic hash function to generate a new hash with the help of the traditional cryptographic technique. The system proposed here is supported by a hashing algorithm to improve the overall surveillance of data using a genetic algorithm (GA) approach for optimization.

A genetic-based hashing algorithm is used to manage and control the transaction node per time in order to improve data reliability. Additionally, with the introduction of a new hashing algorithm, this paper shows improved robustness. Testing the blockchain data-sharing system among the nodes reveals the uniqueness of controlling the access of records.

This paper is organized as follows: a literature review, highlighting the previous methodologies related to blockchain, a GA, and the crypto hash function. Next, methodology is described followed by simulations and results for the proposed algorithm. These include hashing the block, key generation procedure, and encryption/decryption algorithm.

Literature Review

Challenges confronted by the authors relate to enhancing the technique of blockchain-based cryptography. Today, cryptography is used in a variety of sectors to enhance safety and security. For example, it facilitates efforts by companies to secure their information without jeopardizing intimate details of customers, as well as the company.

It helps to understand the operation of CaaS (Containers-as-a-Service) to secure data in a system and the benefits CaaS provides to mobile and desktop computer users today and in the future.[10] In the same way, cryptography plays an essential role in blockchain technology to enhance and provide a safe, secure, and protected environment for sharing cryptographic data among the nodes of a network, without breaching the privacy of any of the information contributed by the nodes.

Blockchain technology was introduced by Satoshi Nakamoto for his well-known contribution to cryptocurrency, which is also called digital currency, i.e., Bitcoin.[11] In a blockchain network, a chain of blocks is linked and raised continuously by accumulating the transactions on the blocks so that a verified list of cryptographic data is formed.

The blockchain provides a detailed report of each transaction.[12] Many domains are involved in bloockchain due to its decentralized technology. These include education and health care.[13] The decentralized technology Ethereum is a blockchain technology used to create a secure verifying smart code, which is created by a peer-to-peer network and builds a new Ethereum-based token. These tokens can be used to power decentralized apps (dApps) with the help of smart contracts.[14] The smart contracts can be used according to the predefined approach for transferring records, and they allow access to a client's records by the medical experts.[15]

Information regarding the patient must be highly secure, which is possible with the implementation of smart contracts in different medical fields.[13] The application of the Internet of Things (IoT) to health care is achieved[16] through the application that deals with storing and transforming various formats. Examples include: images, text, and voice via internet. A secure architecture of healthcare multimedia data with the help of blockchain is presented by[17] comparison in terms of average packet delivery ratio, average latency, and average energy savings with other standard techniques.

A decentralized approach is used in this technique, which allows the data to be distributed, with each part of the distributed data shared with each participant involved in the network. A single block of data can be added to the blockchain using cryptography, with each block of data passing through the verification process. Mathematically, it follows an arrangement from the previous block by keeping the consent of the network decentralized.

The process of verification in the blockchain is called proof-of-work or mining.[18] Blockchain technology has certain advantages such as surveillance of data, immunity, and integrity of data, with no third-party interference. As discussed above, the security of medical records is a top priority in the healthcare industry. These advantages promote storing electronic medical data in the blockchain, and researchers have concluded that in the medical industry, the blockchain could be a feasible solution.[19,20]

Key Features in the Blockchain

Blockchain technology offers decentralization, transparency, and security and integrity of data.

Decentralization

Blockchain technology distributes data within the whole network rather than at a single point. Thus, no third party involved in the network can access and control the data. Blockchain shares and distributes the data with all nodes connected to the network. Put simply, data can only be handled and controlled by trustworthy entities.

Data Transparency

Accomplishing data transparency is a trust-based connection in any technology. Securing and tamper-proofing the data become an important component in any industry. As explained above, due to decentralization, blockchain data are not controlled by a single party. Data can be controlled by each node in the network. This explains why data become transparent and more secure from third-party involvement.

Citation: Blockchain in Healthcare Today 2023, **6**: 244 - http://dx.doi.org/10.30953/bhty.v6.244

Security and Integrity

The blockchain environment uses cryptography to enhance and provide surveillance and integrity to the nodes connected to the entire network. This paper uses a new cryptographic hashing algorithm on the hashes that are gathered in the blocks. The hashes give security to the blockchain and also provide integrity to the data.

Cryptographic hashes are one-way functions that produce the checksum (i.e., a small-sized block of data derived from another block of digital data for the purpose of detecting errors that might have been introduced) for the digital record that are excluded from the data distillation. These characteristics make blockchain decentralized, secure, and integrated cryptographically. Thus, it is a good option for the security and privacy of different types of records.

Today, blockchain technology is used in various fields because of its efficiency and high security of data. Blockchain technology is used in petroleum industries to improve supply chain management problems. The petroleum industry involves a global supply chain management in which international and local order, transportation, import/export, inventory, and information technology are included. In this type of supply chain, an organization is associated with suppliers, distributors, data compilers, and vendors, even with everyone involved. Hence, the blockchain environment is used to control and secure the data in this kind of industry.[21]

Blockchain used in the IoT works with decentralized and distributed data to collect, store, and strengthen transactions among IoT nodes. The system is related to blockchain-based IoT surveillance nodes and blockchain-based settlements, and it can be applied in aspects of the IoT ecosystem.[22]

The blockchain platform offers important and robust implementation in healthcare industries, as well as for the protected and tamper-proof electronic medical record. The system protects ledgers and grants complete access to the patient's medical history and treatment in the healthcare industry. With the help of the blockchain approach, it is easy to secure sensitive medical information. It provides additional surveillance services, accountability, authentication, and confidentiality.[23] The platform is flexible and offers considerable accommodation to surveillance in financial services industries and many other projects, including health care, supply chain management, cybersecurity, banking, data analytics, drug counterfeiting in pharmaceutical sectors, fintech, etc.

This paper aims to deal with the management and security of public data in user-oriented healthcare centers.[24]

Genetic Algorithm

GAs are generally used to develop a high-level solution to optimize any problem and detect problems depending on the operators stimulated by the biology concept such as machine learning and deep learning which is used in the classification of ECG signals. The GA was introduced by "John Holland" in 1960. It was inspired by the concept of "Darwin's Theory of Evolution."[25] The GA starts with a set of solutions representing chromosome-like data structure to obtain an optimal and potential solution for the problem. The main operators in the GA are selection, recombination, mutation, and crossover, which combine to achieve a new generation.

GA is an evolutionary procedure used to optimize problems, including shortest path, intrusion in wireless sensor networks (WSN), bandwidth utilization, and more. The reason behind using the GA in generating the key in underwater wireless sensor networks (UWSNs) is that cryptography through GA provides the lightweight complexity, which is the measure requirement within the UWSNs. The GA approach is random, which enhances cryptographic encryption and decryption. In addition, the GA starts with random results called chromosomes, which can be generated through many random procedures. These randomly generated results can be made more accurate by using different steps of the genetic procedure: fitness measure, crossover, and mutation. To get a more accurate result through GA, it is imperative to have a strong fitness function that applies to the initial random generation to measure its fitness. Fitness function identifies which chromosome can be used for the process of crossover.

In the crossover, two chromosomes produce two more fitted chromosomes that can be tested again using a fitness function. After getting better chromosomes from the crossover, we apply mutation to achieve global optima from local optima. In the proposed algorithm, we used the above-explained steps of the GA to generate half the part of the key for symmetric cryptography.

These traditional steps of GA have many variations according to the scenario and environment. We performed these steps by making the fitness function according to the suitable parameters related to the cryptographic approach's conditions. Today, GA is used and altered according to the requirements. These include a few concepts of GA that have been removed, altered, and for other new concepts introduced in fluid genetic algorithm (FGA) that have not only improve results and convergence control, but also can be used in a range of problems that involve multi-objects and multi-level issues.[26]

GAs are also used to strengthen the key in order to make the complete algorithm secure. As explained, in GA, data are generated randomly, genetic operators are applied, and then it is diffused by genetic and logical operators. In 2018, an algorithm was proposed for better results in terms of strengthening the key with less computational complexity than the two previous algorithms.[27]

Cryptographic Hash Function

The cryptographic hash function allows for mapping data of arbitrary size to a fixed-size bit string called a hash value. It is the fingerprint file (checksum), and it is used to verify that the files have not been tampered with or modified in any way not intended by the author.

The hash function is a one-dimensional function that cannot be reversed.[24] It is a mathematical representation of an algorithm. There are three main characteristics of an ideal hash: easy to calculate, computationally difficult to obtain alphanumeric content that has provided hash, and unlikely that 2-minute different texts can provide the identical hashes. The message acts as input data and is called a "message"; whereas the hash or hashing values act as an output called a "message digest value."

Methodology

The proposed method in this research study consists primarily of three components (i.e., blockchain, GA, and hashing). The proposed method uses secure blockchain technology, which depends on cryptographic hashing for the surveillance of peer-to-peer non-centered distributed ledgers. Furthermore, it uses a GA for the key generation of a robust system in order to protect the data from jeopardization or third-party involvement to avoid the breaching of nodes. The method described above is illustrated in Figure 1.

The Algorithm for GAHBT

START

Step 1: Hashing

Step 1.1: Annexing &Padding

Step 1.2: Append length

Step 2: Key Generation in two parts (by using a different method for each part)

Step 3: Encryption/Decryption

END

Padding and Appending Length

In the first step, a message of b-bits is padded to make it 16 bytes SHA (Secure Hash Algorithm), which is the multiple of 512. The first three "1" bits are appended to the input message, and then a series of off bits are padded, as given in Figure 2.

Hashing

Cryptographic hashes are one-way functions that take input of data and generate the result into a size of fixed length called a digest—also called checksum. Hash functions are not able to decrypt as they are one-way functions; hence, they cannot be encrypted back to form the original message. Many algorithms are available to generate a hash, such as MD5 (message-digest algorithm), SHA1 (Secure Hash Algorithm 1), and SHA2 (Secure Hash Algorithm 2).

This proposed model gives a modified hashing algorithm to generate a hash. The given algorithm creates 128 bits unique message output called a "digest."

Consider a message of length b-bits that needs to digest. For the hashing of a b-bits message, the proposed algorithm passes through different steps described in the algorithm. The model stages will use some assisting functions that comprise buffer and auxiliary functions that have already been initialized.

Algorithm for Generating HASH

START

Step 1: Split the blocks into 16 bits

Step 2: Assign initial values (in hex) to x, y, z

Step 3: Apply characteristics functions for value refining

$$A(x, y, z) = rot^{20}(\neg x \oplus s^{11} y \wedge z)$$
$$B(x, y, z) = rot^{10}(\neg y \wedge s^{14}(z \vee \neg x))$$
$$C(x, y, z) = x \oplus y\ rot^{5} z)$$
$$D(x, y, z) = (x \wedge y) \oplus (\neg z \vee x)$$

Step 4: The table of 64 elements will be established using the sin function

$$abs(sin(i + 1) \times 2^{16})$$

END

Splitting Blocks

After appending the length, the 512 bits message is divided into 16 (Block) words with the length of 32 bits each. (16 × 32). Say M [0, … , N-1] where N is a multiple of 16.

Initializing the Values

Values are initialized by taking the first 32 bits of the fractional parts of the square roots of the first 17 prime numbers. And every prime positioned hex value will be used to initialize the process and store them in (x, y, z).

Characteristic Functions

Now, for the further process, the proposed study introduces some characteristic functions to refine the initial values and make the results more random and secure. These functions are designed to take three words of 16 bits and give the output of 16 bits. Functions are given as follows from (1) to (4):

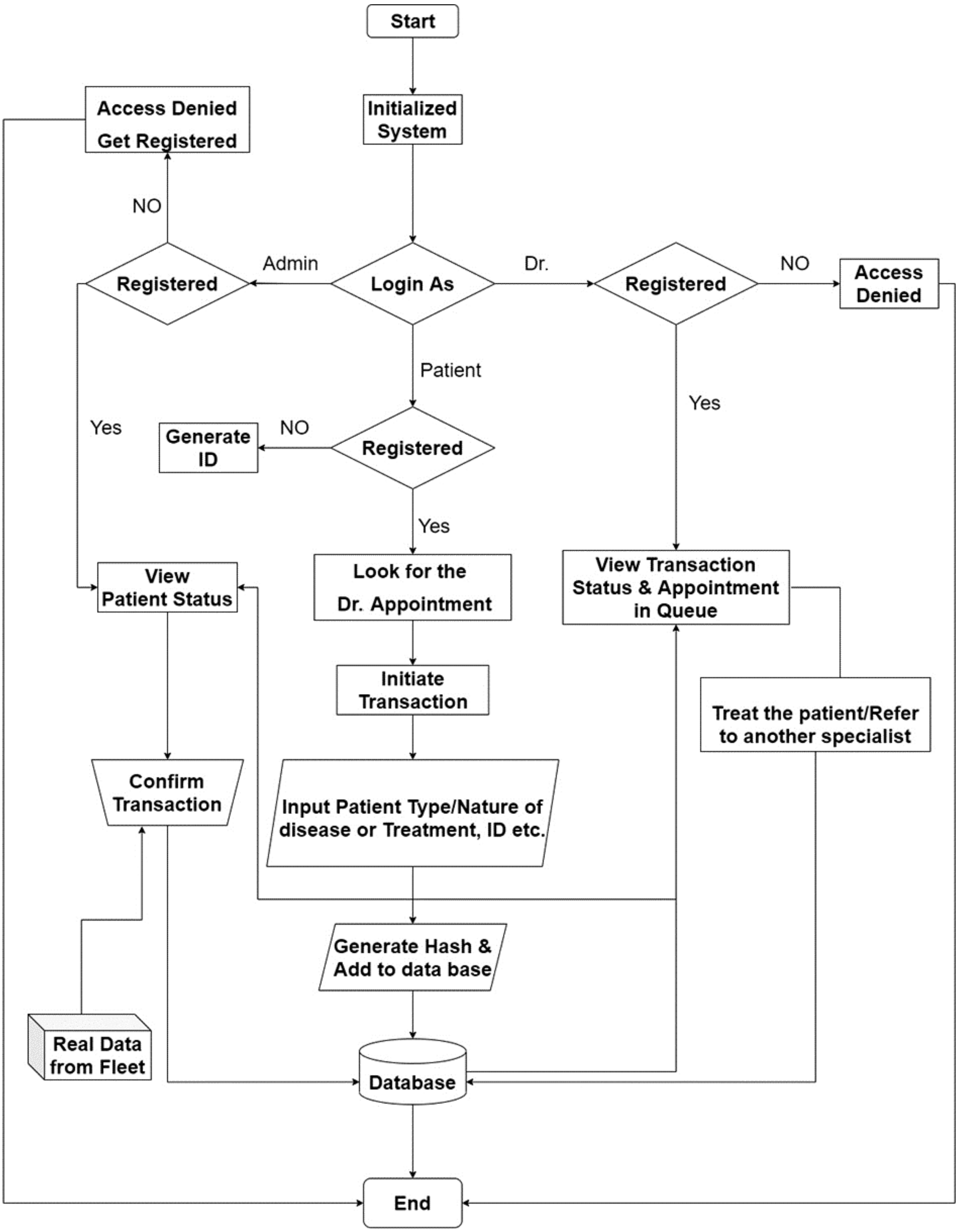

Fig. 1. Process flow of the proposed model of GAHBT (genetic-based hashing algorithm for managing and validating health data integrity in blockchain technology).

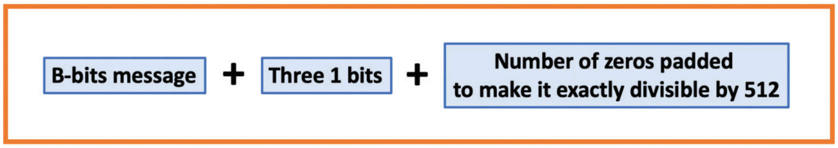

Fig. 2. Illustration of the padding procedure. The b-bit message is based on 64 bits. After that, it is appended under modulo 512 (divisible by 16 for working in hexadecimal). These two steps are taken from Professor Rivest's study[28] to construct the entered message input for digestion by annexing and padding the bits.

$$A\,(x, y, z) = rot^{20}\,(\neg\, x \oplus s^{11} y \wedge z) \quad (1)$$
$$B\,(x, y, z) = rot^{10}\,(\neg\, y \wedge s^{14}\,(z \vee \neg\, x)) \quad (2)$$
$$C\,(x, y, z) = x \oplus y\; rot^{5}\, z) \quad (3)$$
$$D\,(x, y, z) = (x \wedge y)\; \oplus (\neg\, z \vee x) \quad (4)$$

Applications of the above characteristic functions will take place randomly. The selection of the above functions will be decided with the help of a random position generator.

The Table

In the proposed procedure table of 64 elements will be established using the sin function:

$$abs\,(sin\,(i + 1) \times 2^{16}) \quad (5)$$

Let us consider that the table has values K [1, … , 64]. K[i] represents the 1st element of the generated table.

Key Generation Using GA

A key of 256 bits is required for this algorithm. In this study for randomization, we divided the key into two parts named K1 & K2. The left half portion of the key K1 (128 bits) is developed from the (GA), which is an evolutionary algorithm based on the idea of natural selection.[29] Whereas, the right half portion of the key K2 (128 bits) will be taken from the hash generated in section 3C.

Algorithm for Key Generation using GA

START

Step 1: Initial Random Population Generation (bin/hex)

Step 2: Crossover (OX1)

Step 2.1: Create two random crossover points in the parent

Step 2.2: Copy the segment between them from the first parent to the first offspring

Step 2.3: At the second crossover point in the second parent, copy unused numbers from the second parent to the first child, wrapping around the list.

Step 2.4: Repeat for the second child with the parent's role reversed.

Step 3: Fitness Function

$$Fitness\ Function\ = 1 - \frac{Gap\ value}{entropy \times run\ value}$$

$$Entropy\ = log_{2}\left(\frac{2^{n}}{C_{r}^{n}}\right)$$

Step 4: Threshold

Step 5: Mutation

Step 5.1: Select chromosomes

Step 5.2: Apply scramble mutation

Step 5.3: Apply inversion mutation

Step 5.4: Apply swap mutation

END

Steps of Proposed GA

The basic procedure of GA usually involves conventional steps: the process starts with an initial random population composed of various chromosomes. The chromosomes are taken in a binary number system or hexadecimal number system. In the proposed algorithm, the key is generated by using the generation of random population, calculation of fitness function, crossover, and mutation. The process and detail of the GA in different applications may vary according to the scenario of the considered problem.

In the next section, an explanation of the proposed algorithm is discussed in detail.

Initial Random Population Generation

In this step, the random population of 128 bits is generated by pseudo-random numbers in a binary number system that is called chromosomes. The generation of a pseudo-random number is based on a linear congruential process.[30]

Crossover

It is a special operator of GAs that helps differentiate between GAs and other algorithms. There are various types to operate crossover (i.e., one-point crossover, two points crossover, multipoint crossover, random point crossover,

Citation: Blockchain in Healthcare Today 2023, **6**: 244 - http://dx.doi.org/10.30953/bhty.v6.244

uniform crossover, whole arithmetic recombination, and Davis's Order crossover (OX1)).

In this study, Davis's Order crossover (OX1) is used. OX1 is a permutation-dependent crossover that communicates with the data information about relative ordering to children. The working steps of Davis's Order crossover are as follows:

1. Select a random substring from the parents.
2. Copy the parent substring in the initial offspring.
3. Now, from the second point crossover in the second chromosome, start copying the remaining inexperienced bits from the second parent to the first offspring as it covered the substring created in Step 1.
4. Redo the above process for the second child by reversing the parent role.

Fitness Functions

A GA is an optimization approach based on natural selection. The aim of generating this simulated/artificial data is to minimize its differentiation of statistical inference from the actual data. In GAs, the computation of an optimized individual is called fitness. Therefore, to calculate the fitness of the nascent generated random population (chromosomes) in the previous step, a fitness function is required. In this study, the proposed fitness function is given to calculate the fitness of chromosomes.

$$Fitness\ Function\ =\ 1-\frac{Gap\ value}{entropy \times run\ value} \tag{6}$$

$$Entropy\ =\ log_2\left(\frac{2^n}{C_r^n}\right) \tag{7}$$

Run Value

A run-in (6) is described as a series of growing values or a sequence of reducing values. The variety of growing or reducing values is the run's duration. It is a non-parametric test used to check the randomness of a pattern.

Entropy

Entropy in (7) is used to compute diversity or randomness in generated data. Full entropy in data represents that the data are completely random and there is no significant pattern present in the data, and it is not able to recognize easily. For encryption and hashing functions, highly entropic data are considered to make algorithms unpredictable or even secret to ensure the safety and security of procedure. In the same way, the gap value is used to calculate the randomness of data by computing the gaps between the digits that appear in the repetition of particular digits. Notably, this study counts the maximum number of zeros or ones that appear in chromosomes. The run test is a non-parametric statistical test of randomness. It tests the value of a run that is up or down or above or below the mean by comparing actual and expected values.

Threshold

When the fitness function provides values close to 1, it can be assumed to be fitted values. The threshold for this study is set at 90%. The best fit value will come from the set of stored fitted values.

Mutation

The mutation operation is critical to the success of GAs since it diversifies the search directions and avoids convergence to local optima. The mutation is a major step in GAs. It is used for preserving genetic diversity among various chromosomes (generations)[28] very helpful to attain the global optima. After crossover, mutation changes at least one bit in the chromosomes to reflect the results of surrounding GA's.[31] In this step, all those chromosomes with a higher threshold value in the crossover will be selected to undergo the mutation process.

This paper uses a combination of mutation processes for the best results.[32] Steps are given as follows:

1. Chromosomes are chosen according to the set criterion.
2. First of all, scramble mutation is applied. In this type of mutation, two substrings are chosen randomly, and the highlighted values are shuffled or scrambled randomly. The rest of the experienced strings remain the same.
3. After scrambled mutation, inversion mutation takes place. It selects one random substring and inverts it.
4. After obtaining an inverted chromosome uses, swap mutation is applied. For this, two positions are generated randomly, and their values are swapped.

This is how one round of mutation takes place. After the mutation process, the fitness function is applied to each processed chromosome and stores the best-fitted values that pass through the set criteria. Those values that fall outside the set criteria of fitness will get into the process again. All steps are repeatedly applied for about 500 rounds until more suitable values are obtained than the previously stored values.

In case of getting a less fitted value of the chromosome after 500 rounds, the previously best-fitted value is taken as the finishing part of the key. In the other case, when the fitness value after calculating the fitness function is better than the stored one, take it as the first half portion of the key. The complete process is illustrated in the flow chart in Figure 3.

Encryption/Decryption

Encryption

The proposed algorithm divides data into equal-length blocks and then encrypts each block with the help of a

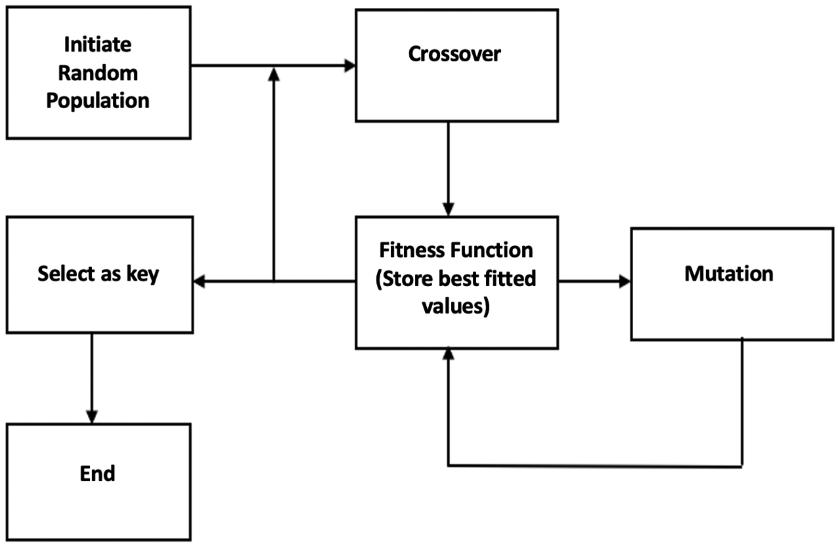

Fig. 3. Flowchart showing the steps of GA (genetic algorithm).

Fig. 4. Encryption procedure by using the first part of the key.

mathematical series of functions called the key of 256 bits. The proposed algorithm will construct the key in two parts, each of which is 128 bits. Initially, the key is broken into eight parts, each of which is 32 bits; therefore, eight blocks of 32 bits are formed. Say (K1, K2, K3, … , K8).

Now consider the four alternate parts of blocks that are K11 = K1, K3, K5, and K7, and together, they are 128 bits, and the remaining 128 bits of the key are K2 = K2, K4, K6, K8.

In this method, the encryption of the M length plain text message will be done by using the key (K) of length 256 bits as shown in Figure 4. And detail is given as follows:

Encryption of B1

1. Consider the first part of the plain text B1. Get ASCII codes of each character of the first part plain text message (B1) and store them in an ASCII array.
2. Convert the result from Step 1 into a binary number system.
3. Padding the result from Step 2 to make it 128 bits.
4. Apply circular shift on the result from Step 3.
5. Break the result from Step 4 into four equal parts, B11, B12, B13, and B14 which are 32 bits each.
6. Now generate a random number of four digits (say r1) and assemble the parts of the result of 5. according to the position generated and store the value.
7. Generate a random position (say r2) by a pseudo-random number generator and store that number and rotate the resultant of 6. according to the generated number.
8. Apply XOR between key (K11), which is 128 bits, and the resultant of 7.
9. Now break the resultant into two parts, each of 64 bits, and find out the hamming distance, say (d).
10. Take d as a position and rotate the result from Step 7.
11. Now find the least prime factorial of hamming distance d say f and apply f time right circular shift on Step 9.
12. Take complement of all prime positions of Step 10.
13. Convert the result into ASCII values and consider it as the right portion of encrypted plain text.

Encryption of B2

Now consider the remaining half plain text and half key; both of them are 128 bits. In this part of the encryption, the process goes through a variety of matrix operations like transpose, permutation, row, and column mixing and leading diagonal shifting on the plain text message to guard and protect the plain text.

1. Consider the plain text and calculate ASCII values of corresponding characters.
2. Convert (1) in the binary number system.
3. Make a matrix of order 8 × 8 of the plain text of 128 bits resultant from 2.
4. Construct a matrix of the same order (8 × 8) using 128 bits key.
5. Start with the column mixing with the help of a random number generated from 1 to 8 with the help of a pseudo-random number generator (say r3) and store this value.

6. Apply the column mixing on the result from Step 5.
7. Take transpose of 6. Start from the last column.
8. Apply XOR with the key matrix to get the matrix of the same order (8 × 8) and bits (128).
9. Now we have a matrix of order 8 × 8. Break the matrix into two equal parts, each of 4 × 8. Say D1 and D2. Find the Hamming distance of corresponding values (say h1, h2, …, h32). Add all the hamming distances to get a single value of h ≤ 256.
10. Find g.c.d of h and 256.
11. Randomly generate a prime number (say r4) and apply the right circular shift/rotation on Step 8, g.c.d times and obtain another matrix of order 8 × 8.
12. Repeat Steps 5 to 11. Till the random number of times. (Generate a pseudo-random number and store it as r5).
13. Apply XOR between key matrix and resultant of 11 to get the matrix of order 8 × 8 of 128 bits.
14. Convert the obtained matrix elements into ASCII values and concatenate them with the first part of encrypted plain text.

Decryption

- Cyphertext C of 265 bits
- Divide it into two halves, C1 and C2, respectively.

Decryption of B1

1. Consider the right portion of C that is C1 as mentioned above of 128 bits.
2. Take complement of all prime positions of C1.
3. Apply left circular shift on resultant of 2. f times., where f is the least prime factorial of hamming distance. (That is previously saved)
4. Rotate the resultant at the d position of 3 by f times.
5. Apply XOR between the first part of the key (K11) and the result of 4.
6. Break the results from 5 into four halves, each of 32 bits.
7. Rearrange the blocks using the reverse of r2 stored already.
8. Apply right circular shift.
9. Convert into ASCII codes.

Decryption of B2

1. Consider C2 and convert it into binary numbers.
2. Convert it into matrices of order 8 × 8.
3. Apply XOR operation with the key matrix that is K22 to get the original matrix of order 8 × 8.
4. Apply left circular shift using (r4) as position according to the g.c.d times.
5. Again, apply XOR with the K22 to get the matrix of the same order (8 × 8) and bits (128).
6. Take reverse transpose of the resultant 5 according to the first row.
7. Take permutation of the result in such a way that the first column will become the last two rows and so on.
8. With the help of a pseudo-random number (r3), mix the columns 8 × 8.
9. Repeat Steps 3 to 8 according to the stored r5 random number times.
10. Consider the matrix and convert it into plaintext.

Algorithm for Encryption/Description

START

Step 1: Represent the message as an integer from 0 to N-1 length

Step 2: Divide the plain text message into two halves, B1 and B2.

Where B1 & B2 will be used with the K11 & K22, respectively.

Step 3: Divide the key obtained in section 3.6 into 8 parts. (K1, K2, K3, … , K8).

Make K11 by combining the odd key parts and make K22 by combining even key parts.

Step 4: K11 and K22 are both 128 bits, and together they make 256 bits.

Step 5: Both parts of plane text B1 and B2 will be encrypted by K11 and K22, respectively.

END

Simulations

Experimental environment

For the implementation of the proposed blockchain model, implementation has been done by considering the single health record of size 30.82 MP that consists of a standard size of image and text that is intraoral photography of size 1.64 MB, orthodontic cephalogram of size 1.40 MB, skin lesion photography of size 22.17 MB, dental panoramic X-ray of size 0.85MB.[33] The reason behind considering the single record is to avoid the delay in processing when different records are grouped into a block. The block size is considered 1MB for the representation of the health record query. The block is divided into two parts: the body and header, with the header storing the metadata information that is the previous block's hash, timestamp, Merkle root hash value, block number, and version;[34] whereas the body stores the information of health record for the experiment. The header size is 80 bytes. In the experiment, we have used proposed hashing algorithm that generates a unique 256-bit output for a given input.[35] All the experiments for blockchain system have been performed with increasing number of health records (4,000, 5,000, 6,000, 7,000, 8,000, and 9,000) and increasing number of hospitals (10, 20, 30, 40, 50 and 100). We gradually increase the number of records by keeping the number of hospitals constant, which is 10. Similarly, we gradually increase the number of hospitals by keeping the number of records constant, based on 4,000.[36] All the simulations have been done on the MATLAB simulator.

Results

Figure 5 represents the execution time for upgrading the health records using a minimal blockchain model from [GBHA], client/server, and the model development in this paper. It represents the graph has a linear relationship between blockchain systems and client/server.[37,38]

It represents that the time consumed in execution for the client network server and blockchain models is in linear relation with the health records. The time taken by the client network is less than the time consumed by the blockchain due to the working consensus mechanism of the blockchain for the acceptance and duplication of data. The data records are needed to upload on the ledger that must be forwarded to the medical centers that will have a copy of the ledger record for acceptance. For consensus, every associated node will transmit the hash of blocks to the other peers present in the network before appending the hash to the record ledger. Whereas, in the client-server model, the record of data is upgraded to where the information of the patient is saved.

The time consumed in execution for the client-server is less compared to the blockchain system designed for updated health record ledger. On average client/server algorithm takes 8.5 times less time than the blockchain.

Figure 6 represents the working efficiency of client/server and blockchain models in representing data quantity transmitted to upgrade health data versus increment/appraisal in the health record. It shows that the amount of data transmitted in the blockchain is more compared to the client/server network because every upgraded record request is forwarded to the other nodes in the network.

Figure 6 represents the relationship between data transfer and the number of health records in the client model and blockchain models. This is repeated in Figure 2. It has been shown in the graph that the data transmission through blockchain is more compared to the client-server system because every updated health record is forwarded to the rest of the network peer node by creating more data transmission. Additionally, all peer nodes forwarded the block's data, including the hash, to the rest peers in the network and boosted the transmission of data, and generated the request. In the blockchain, the execution time is due because, for the data, the query is forwarded to all nodes in the network.

In Figure 7, the execution time of blockchain and client/server models has been represented for the querying health data from the database with an increase in the number of health records. The less execution time of the blockchain compared to the client-server can be seen from the graph because first of all, the data have to be retrieved from the database where the record is present, whereas, in blockchain, the data are recovered from the local copy of the ledger present in all nodes.

Mass data transmission uses a client-server approach to query health data from databases as the number of health records increases. From the databases, the transmission of health records, as they increase in number, is represented in Figure 4 based on the client-server model for querying healthy data. From the figure, the data transmitted by the blockchain network is greater compared to the client-server network.

Figure 8 represents the transmission of data by the client-server and blockchain network. To upgrade the health data, the number of medical centers is increased by blockchain and client-server models. It represents the

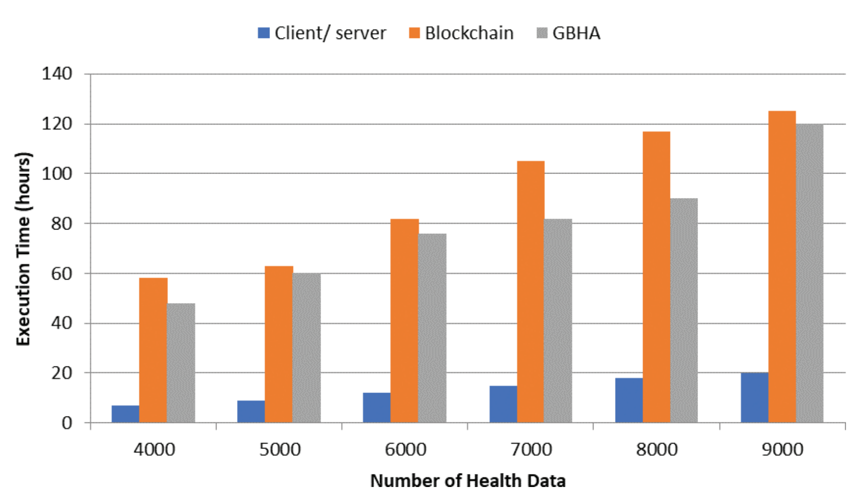

Fig. 5. Relation between the execution time and health records in working consensus mechanism.

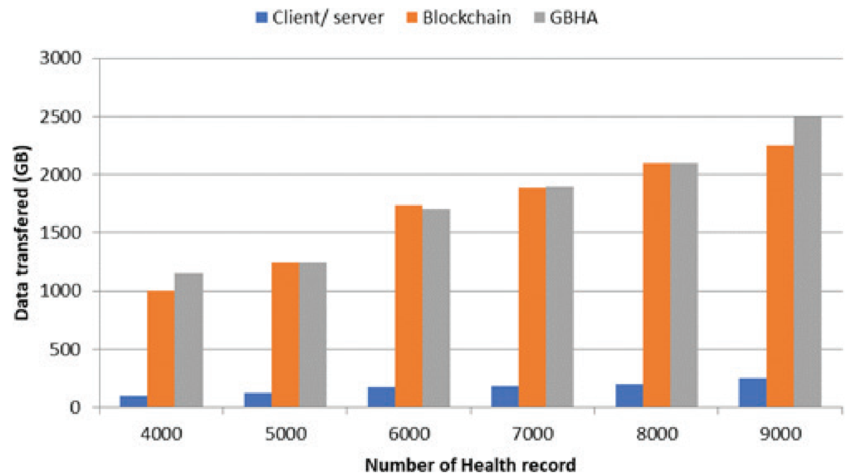

Fig. 6. Relation between data transfer and the number of health records.

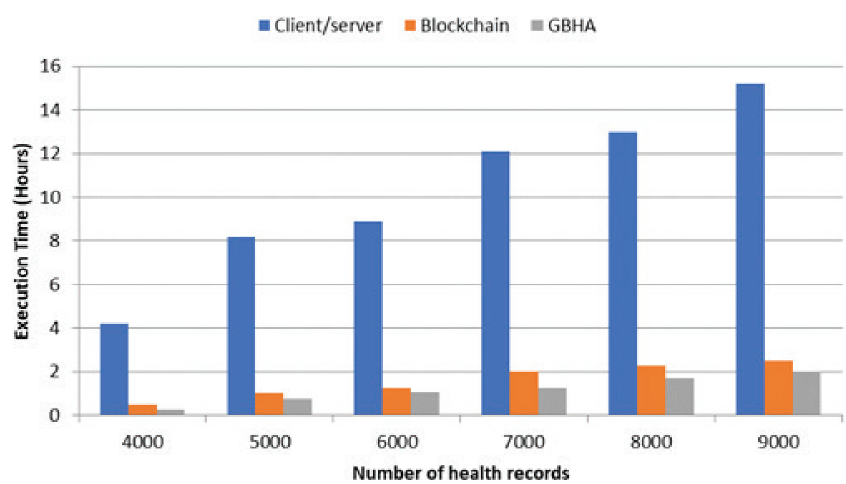

Fig. 7. Relation between querying health records and execution time.

time consumed by blockchain more than the client-server approach because of the consensus protocol that the former uses.

Figure 9 presents details regarding the time for execution for querying the electronic health data from the data bank with an increase in the number of medical centers. The graph represents that the execution time of blockchain is significantly less than the client-server network for a health data query.

Figure 10 represents the query of health data transmission by client-server and blockchain models as the number of medical centers increases. It represents the constant behavior of data transmission due to the number of data queried irrespective of the number of medical centers. The required query of the medical center will be performed, and the transmitted data using blockchain for the query of health data increases as the number of medical centers increases.

Conclusion

The purpose of the research is to present a secure decentralized ledger in a database that manages the healthcare distribution records using a secure and immune genetic-based hashing algorithm in blockchain technology. The system proposed in this paper uses a new cryptographic hashing algorithm to provide fundamental security. It raises the system's immunity to make it more efficient and tamper-proof and protects it from numerous attacks. When it turns to the comparison, the study in this paper provides higher scalability, immunity, data integrity, and robustness than the other blockchain systems, which enables sharing of secure sensitive healthcare data among various network nodes. Additionally, the proposed algorithm can be safely utilized effectively in sharing private and sensitive healthcare information in different environments like healthcare institutes, clinics, hospitals, and patients' families.

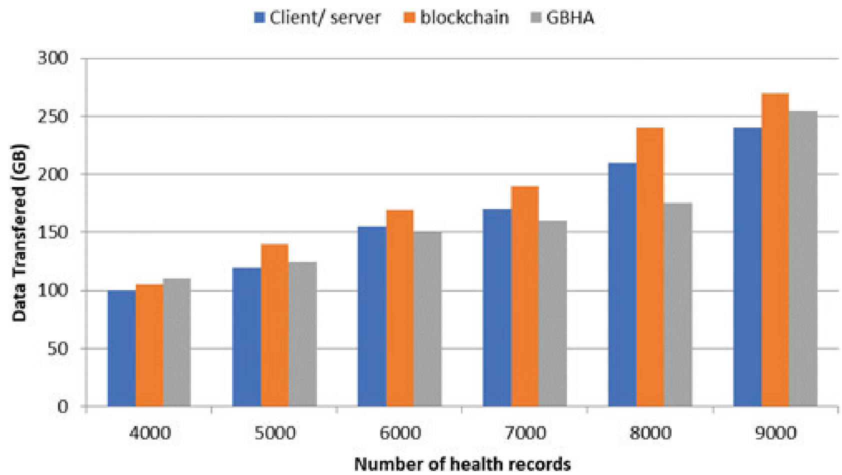

Fig. 8. Relation between the number of health records and data transfer in GBs in the working consensus mechanism.

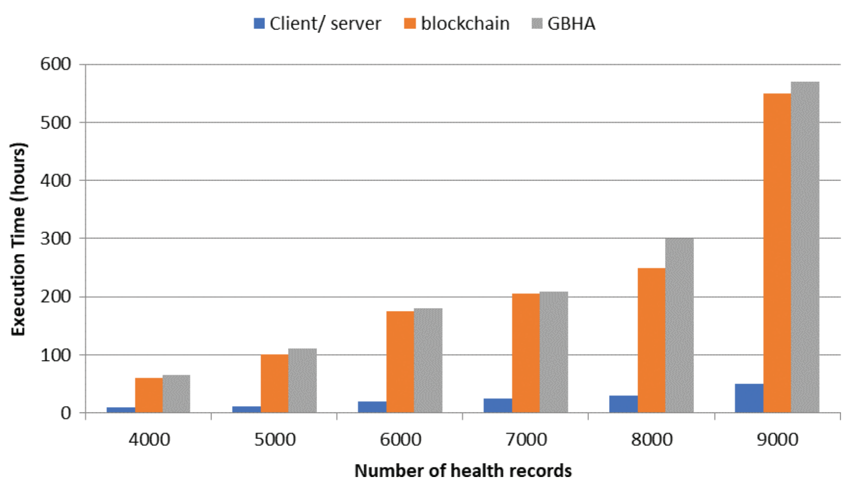

Fig. 9. Relation between transmission of the number of health records and execution time as the number of medical centers increases.

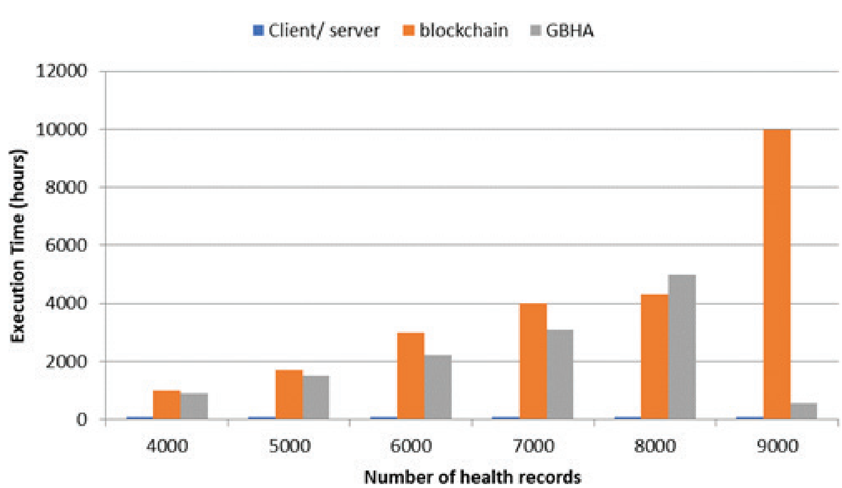

Fig. 10. Relation between transmission of the number of health records and execution time as the number of medical centers increases.

Acknowledgments

NA.

Conflicts of Interest

The authors declare no potential conflict of interests at this time.

Funding Statement

No funding supported the preparation of this article.

Contributors

Dr. Fozia has developed the methodology, whereas Ms. Urooj has done the simulation. Ms. Aisa has done the write-up of the manuscript, and Rehan has done the overall supervision of the paper.

References

1. Cartwright-Smith L, Gray E, Thorpe JH. Health information ownership: legal theories and policy implications. Vand J Ent Tech L. 2016;19:207.
2. Campe J, Kitchen LM, Porterfield M, Sandeen B. Environmental assessment consolidated communications Squadron Facility Nellis Air Force Base, NV. Omaha, NE. Department of the Air Force 99 CES/CEV. 2005. https://apps.dtic.mil/sti/pdfs/ADA633780.pdf
3. Meng W, Tischhauser EW, Wang Q, Wang Y, Han J. When intrusion detection meets blockchain technology: a review. IEEE Access. 2018;6:10179–88. https://doi.org/10.1109/ACCESS.2018.2799854
4. Yang JJ, Li JQ, Niu Y. A hybrid solution for privacy preserving medical data sharing in the cloud environment. Fut Gen Comp Syst. 2015;43:74–86. https://doi.org/10.1016/j.future.2014.06.004
5. Al Omar A, Rahman MS, Basu A, Kiyomoto S. Medibchain: a blockchain based privacy preserving platform for healthcare data. International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Springer, Cham, December 2017, pp. 534–43.
6. Schneider J, Blostein A, Lee B, Kent S, Groer I, Beardsley E. Profiles in innovation: blockchain–putting theory into practice. Goldman Sachs. 2016. https://pgcoin.tech/wp-content/uploads/2018/06/blockchain-paper.pdf
7. Dorri A, Steger M, Kanhere SS, Jurdak R. Blockchain: a distributed solution to automotive security and privacy. IEEE Commun Mag. 2017;55(12):119–25. https://doi.org/10.1109/MCOM.2017.1700879
8. Radanović I, Likić R. Opportunities for use of blockchain technology in medicine. Appl Health Econ Health Policy. 2018;16(5):583–90. https://doi.org/10.1007/s40258-018-0412-8
9. Fairley P. Blockchain world-feeding the blockchain beast if bitcoin ever does go mainstream, the electricity needed to sustain it will be enormous. IEEE Spectr. 2017;54(10):36–59. https://doi.org/10.1109/MSPEC.2017.8048837
10. Rahimi N, Reed JJ, Gupta B. On the significance of cryptography as a service. J Inform Sec. 2018;9(4):242–56. https://doi.org/10.4236/jis.2018.94017
11. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. Decentral Bus Rev. 2008;21260. https://bitcoin.org/bitcoin.pdf
12. Sun J, Ren L, Wang S, Yao X. A blockchain-based framework for electronic medical records sharing with fine-grained access control. PLoS One. 2020;15(10):e0239946. https://doi.org/10.1371/journal.pone.0239946
13. Nishi FK, Khan MM, Alsufyani A, Bourouis S, Gupta P, Saini DK. Electronic healthcare data record security using blockchain and smart contract. J Sensors. Vol. 2022. https://doi.org/10.1155/2022/7299185
14. Gharat A, Aher P, Chaudhari P, Alte B. A framework for secure storage and sharing of electronic health records using blockchain technology. ITM Web of Conferences, Vol. 40, EDP Sciences, p. 03037.
15. Sreeraj R, Singh A, Anbarasu V. Preserving EMR records using blockchain. Ann Rom Soc Cell Biol. 2021;25(6):5344–50.
16. Rathee G, Sharma A, Saini H, Kumar R, Iqbal R. A hybrid framework for multimedia data processing in IoT-healthcare using blockchain technology. Multimedia Tools Appl. 2020;79(15):9711–33. https://doi.org/10.1007/s11042-019-07835-3
17. Sharma A, Tomar R, Chilamkurti N, Kim BG. Blockchain based smart contracts for internet of medical things in e-healthcare. Electronics. 2020;9(10):1609. https://doi.org/10.3390/electronics9101609
18. Hussein AF, ArunKumar N, Ramirez-Gonzalez G, Abdulhay E, Tavares JMR, de Albuquerque VHC. A medical records managing and securing blockchain based system supported by a genetic algorithm and discrete wavelet transform. Cogn Syst Res. 2018;52:1–11. https://doi.org/10.1016/j.cogsys.2018.05.004
19. Boonstra A, Versluis A, Vos JF. Implementing electronic health records in hospitals: a systematic literature review. BMC Health Serv Res. 2014;14(1):1–24. https://doi.org/10.1186/1472-6963-14-370
20. Gunter TD, Terry NP. The emergence of national electronic health record architectures in the United States and Australia: models, costs, and questions. J Med Internet Res. 2005;7(1):e383. https://doi.org/10.2196/jmir.7.1.e3
21. Chima CM. Supply-chain management issues in the oil and gas industry. J Bus Econ Res. 2021;5(6). https://doi.org/10.1051/itmconf/20214003037
22. Kshetri N. Can blockchain strengthen the internet of things? IT Prof. 2017;19(4):68–72. https://doi.org/10.1109/MITP.2017.3051335
23. Ekblaw A, Azaria A, Halamka JD, Lippman A. A case study for blockchain in healthcare: "MedRec" prototype for electronic health records and medical research data. Proceedings of IEEE Open & Big Data Conference, Vol. 13, August 2016. https://www.healthit.gov/sites/default/files/5-56-onc_blockchainchallenge_mitwhitepaper.pdf
24. Mettler M. Blockchain technology in healthcare: the revolution starts here. 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), IEEE, September 2016. https://ieeexplore.ieee.org/document/7749510
25. Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press; 1992. https://mitpress.mit.edu/9780262581110/adaptation-in-natural-and-artificial-systems/
26. Jafari-Marandi R, Smith BK. Fluid genetic algorithm (FGA). J Comp Design Eng. 2017;4(2):158–67. https://doi.org/10.1016/j.jcde.2017.03.001
27. Nazeer MI, Mallah GA, Shaikh NA, Bhatra R, Memon RA, Mangrio MI. Implication of genetic algorithm in cryptography to enhance security. Int J Adv Comp Sci Appl. 2018;9(6): 371–379. https://doi.org/10.14569/IJACSA.2018.090651
28. Hasn AA. A literature survey on the usage of genetic algorithms in recent cryptography researches. 2015.
29. Zheng Z, Xie S, Dai HN, Chen X, Wang H. Blockchain challenges and opportunities: a survey. Int J Web Grid Serv. 2018;14:352–75. https://doi.org/10.1504/IJWGS.2018.095647

30. Deng LY, Lu HHS, Chen TB. 64-Bit and 128-bit DX random number generators. Computing. 2010;89(1):27–43. https://doi.org/10.1007/s00607-010-0097-9
31. Kumar A, Chatterjee K. An efficient stream cipher using genetic algorithm. 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE, March 2016, pp. 2322–6. https://ieeexplore.ieee.org/abstract/document/7566557
32. Hong TP, Wang HS, Chen WC. Simultaneously applying multiple mutation operators in genetic algorithms. J Heuristics. 2000;6(4):439–55. https://doi.org/10.1023/A:1009642825198
33. Garcia Ruiz M, Garcia Chaves A, Ruiz Ibañez C, et al. mantisGRID: a grid platform for DICOM medical images management in Colombia and Latin America. J Digital Imaging. 2011;24(2):271–83. https://doi.org/10.1007/s10278-009-9265-x
34. Ismail L, Materwala H. A review of blockchain architecture and consensus protocols: use cases, challenges, and solutions. Symmetry. 2019;11(10):1198. https://doi.org/10.3390/sym11101198
35. Ismail L, Materwala H. Blockchain paradigm for healthcare: performance evaluation. Symmetry. 2020;12(8):1200. https://doi.org/10.3390/sym12081200
36. Yang S, Orlvoa Y, Lipe A, Boren M, Hincapie-Castillo JM, Park H, et al. Trends in the Management of Headache Disorders in US Emergency Departments: Analysis of 2007–2018 National Hospital Ambulatory Medical Care Survey Data. JCM. 2022;5. https://www.mdpi.com/2077-0383/11/5/1401#
37. Shi S, He D, Li L, Kumar N, Khan MK, Choo KKR. Applications of blockchain in ensuring the security and privacy of electronic health record systems: a survey. Comp Secur. 2020;97:101966. https://doi.org/10.1016/j.cose.2020.101966
38. Rivest RL, Agre B, Bailey DV, et al. MIT Computer Science & Artificial Intelligence Laboratory. Cited on 6. 2008. https://people.csail.mit.edu/rivest/pubs/Riv08c.slides.pdf