SN Computer Science (2022) 3: 403
https://doi.org/10.1007/s42979-022-01284-w

ORIGINAL RESEARCH

Secure Covid-19 Electronic Health Records Management for Telediagnosis and Travel Ticket Assistant System Using Cryptographic Approaches

Vinod Ramesh Falmari[3] · M. Brindha[1] · Seokbum Ko[2]

Received: 12 June 2021 / Accepted: 1 July 2022 / Published online: 28 July 2022

Abstract

Presently, the whole world is suffering from the Covid-19 pandemic. In this harmful situation, using information and Internet technology is mandatory for the government and medical practitioners. After the lockdown, the government needs to take important decisions to allow passengers to travel through air, rail, and land. In the present situation, people need to get a medical report from the hospitals to travel through various modes of transport. In this regard, the Covid-19 history of the passengers plays an important role in issuing tickets to the passengers. Hence, in this paper, a novel authentication method using InterPlanetary File System (IPFS) is suggested to retrieve the Covid-19 history of all passengers to determine whether to issue tickets and allow people to travel through various modes of transport. The government can share the Covid-19 status of passengers with the ticket issuing authority. The medical practitioners can share medical reports and medical images of such people for telediagnosis. To provide security, a novel privacy-preserving storage and sharing of Covid-19 records using secure authentication and image cryptosystem are proposed using chaos, cryptographic hash (SHA-256), Paillier cryptosystem, and IPFS. Security analysis shows that the system can withstand various kinds of attacks.

Keywords IPFS · Chaos · Covid-19 · Tele-diagnosis · Image cryptosystem · Electronic health records

Introduction

Nowadays, the whole world is struggling with the Covid-19 pandemic. The medical experts are acting like warriors in this battle. It is required to maintain the Covid-19 records database for the smooth functioning of all strategic activities. Covid-19 database includes all the patients' personal details, travel history, contact history, diagnosis reports containing corresponding medical images like X-ray, MRI, PET scans, etc. Travel history includes the mode of travel like a car, train, bus, flight, etc. Contact history includes the details of the person's friends, colleagues, family members, etc., who are in physical contact. The government can easily trace out the suspected people from the travel history and contact history. From this information, the government can take strategic decisions like quarantine, lock-down, sanitization, etc., in concerned areas like working offices, residential colonies, airports, railway stations, bus stands. This ultimately results in taking control over the patient as well society to prevent further worst effects. In countries like India, the government has categorized the whole country into three different zones—Red, Orange, and Green. Depending on these color codes assigned to the districts, somewhat relaxation or more strictness is applied. In the present and future days, to come back to normal routine life, the government needs to make important decisions to allow passengers to travel through air, rail, and land. In the present situation, people need to get a medical report from the hospitals to travel through various modes of transport. It is observed that to get the Covid-19 report, so many people are standing in a long queue near hospitals. To manage the crowd and provide essential things to them like drinking water, food,

✉ M. Brindha
brindham@nitt.edu

Vinod Ramesh Falmari
vinodfalmari@gmail.com

Seokbum Ko
seokbum.ko@usask.ca

[1] National Institute of Technology, Tiruchirappalli, Tamil Nadu 620015, India

[2] University of Saskatchewan, Saskatoon, Canada

[3] Government Polytechnic Miraj, Sangli, Maharashtra 416410, India

etc., is another issue. Also, there is a chance of malpractices in issuing these medical certificates. A person can give a bribe to get a medical certificate. In this regard, the genuine Covid-19 history of the passengers plays an important role in issuing tickets to the passengers. Hence, an information technology (IT)-based Covid-19 status checking system is required everywhere. Through IT, a common database for Covid-19 records can be maintained. By accessing this database, the concerned authority can easily rectify the Covid-19 status for the person, e.g., railway authorities cannot issue a ticket to the person because of his Covid-19-positive status. The use of modern technology plays a key role in winning this battle. Because of digitization in the medical field, diagnosis speed, as well as performance, is increased. Also, due to rapid growth in Internet technology, Covid-19 patient records can be shared among concerned experts in one click. Quick sharing of information is advantageous to all concerned authorities. The medical experts can do telediagnosis from anywhere and anytime.

However, sharing such records openly through the Internet is not at all secure. Some people, like terrorists, always wait for such situations to employ their agenda. Cyber attacks are possible during sharing of such records. From a security point of view, confidentiality, integrity and availability are raised. Confidentiality means preserving privacy using encrypting data. Integrity refers to the non-modification of the data during transmission by an unauthorized user. Availability confirms that data are accessible to read and modify records anytime and anywhere for authenticated users like medical experts, government officers, and other agencies. So, a common shared storage server is needed to maintain the records. But, it is mandatory to store in encrypted form only. The commonly used storage servers are centralized. Single-point failure may happen. To avoid this, IPFS (InterPlanetary File System) is preferable. IPFS provides decentralized, tamper-proof, and immutable storage services.

The next subsequent sections of this paper are: in "Related Work", the related work is discussed; in "Preliminaries", the preliminaries required for the proposed system are briefly given. The framework and workflow of the proposed scheme are described in "Proposed System Design". Performance and security analysis is discussed in "Security and Performance Analysis". The conclusions are made in the last section.

Related Work

Digital imaging and Internet technology are useful in medical, government, and other concerned fields during these Covid-19 pandemic days. Telemedication, telesurgery, telediagnosis are some of the examples in the medical field. For the government, the Covid-19 patient's travel history and contact history are required to avoid further complicated situations and to handle law and order smoothly. The other fields like banks, travel agencies also need patients' status to do their activities. So, to implement the proposed system, the design of a storage system, as well as access through authentication protocol, is required. By considering the issue of confidentiality, data encryption is also needed. In consideration of the above points, here some common cryptographic approaches are reviewed.

In cryptography, various algorithms like DES—Data Encryption Standard, AES—Advanced Encryption Standard, secure cryptographic hashing, Digital Certificates, Digital Signature, El Gamal, RSA, Elliptic curve cryptography are provided which are used to design secured systems. Cryptographic hash functions are very useful in digital security systems like digital fingerprinting of messages, authentication of messages, and key derivation [1]. Hashing function maps data of any length to a fixed-length bit string. The properties of hash functions are—deterministic, one-way, irreversible, quick to compute, and produce avalanche effect [2, 3]. Also, it is a one-way function, i.e., a function which is not reversible. Because of this, these functions are working as an important tool in modern security systems [4]. In modern security applications, researchers are using different cryptographic hash functions like MD-5, SHA-256, RIPEMD-160, Whirlpool, bcrypt, BLAKE3, and many more. Among these, SHA-256 is more secure and popular. SHA-256 was developed by US National Security Agency (NSA) in 2001 [5]. The SHA-256 algorithm takes a message of arbitrary length as input and produces a 256-bit message digest as output.

Apart from these conventional cryptoalgorithms, homomorphic encryption is also designed. Using homomorphic computation, it is possible to do computations in an encrypted domain. Such encrypted domain is complex, costly, and preferably used in high-security applications like data aggregation, secure bidding systems, and cloud computing [6]. Homomorphism is the property that preserves the original properties and association among the information set after converting it into another information set. In cryptography, homomorphic functions allow doing computation without decrypting homomorphically encrypted information. The obtained result after the computations is in cipher form, and after decryption, it will be the same as the computation in plain domain [7]. Homomorphic cryptosystems are designed in two categories—fully and partially. Partial homomorphic cryptosystem does only restricted computations like addition and multiplication. Full homomorphic cryptosystem allows all computations. Craig Gentry [8] designed a full homomorphic system using lattice theory. Partial homomorphic algorithms like DGK, Paillier, and El Gamal are studied and proposed in [9]. Paillier provides addition and multiplication operations in a homomorphic manner. Paillier method is computationally comparable to RSA, and it uses modular maths. In reality, Paillier is useful in systems like secure distortion computation and secure transform [10, 11].

Now concerning authentication, the researchers have proposed various authentication techniques based on biometrics, PINs, smart cards, challenge-response, multi-factor, multi-level, etc. However, in reality, for users' easiness, authentication using UserId and Password is commonly used, and implementation cost is also less [12, 13]. Hence, companies and institutes often prefer password-based systems [14]. However, it is noticed that the passwords are crackable mostly due to users' inattentive habit [15]. In practice, most of the users often select delicate passwords [16, 17] and use the same passwords again and again for different services [18, 19]. The selected passwords are the most common words to their day-to-day world and also not difficult to memorize [20, 21]. The user credentials like UserId and Password are preserved in the Authentication Data Table (ADT) in the server. If the security of the server that stores ADT is low, hacker hacks the server and makes passive attacks on ADT [22]. Cryptographic hashing functions are irreversible and are commonly used in authentication techniques [23, 24]. In practice, passwords are hashed and then stored in ADT [25]. But, such direct hash values in ADT are not safe because of precomputed table attacks such as lookup and rainbow table attack [26]. To avoid these attacks, the salting of hash passwords is in practice. Salting is nothing but adding noise. But, salting passwords is also not safe against a dictionary attack. In this type, the matching is done with a guessed password and all the words from the dictionary. To prevent this attack, key stretching can be applied. In key stretching, delicate passwords are transformed into secure passwords. But, the key stretching technique is not well protected against Narrow-Pipe Attack [27]. Authentication based on Encrypted Negative Password (ENP) that uses a negative database is proposed by Luo et al. [28]. ENP is much stronger to protect from a lookup table and dictionary attack. However, the complexity of generating negative passwords is more. In the proposed password-based secure authentication the encryption keys are depending on the input credentials; hence, the overall security of the authentication system is strengthened. Also, authentication data table attack analysis with various metrics is performed and results in a robust and secure authentication system.

Considering the proposed system, the Covid-19 report consists of all the details like personal details, travel history, contact history, medical diagnosis reports, medical images like X-ray, and they are considered in image form (scanned documents) only. So, every record/report is converted to an image and used in the proposed architecture. But, as a cloud is a centralized system, if a single cloud model fails, it creates the problem of unavailability [29]. Nowadays, InterPlanetary File System (IPFS), which is a decentralized storage technology, is acting as an alternative for the cloud. Recently, using IPFS technology in combination with blockchain is coming in boom [29–31]. Chen et al. [31] proposed a P2P file system using IPFS and blockchain technology. Zheng et al. [30] proposed an IPFS-based blockchain storage system to distribute the load. Jin Sun et al. [29] used IPFS and blockchain for the storage system of medical records.

As IPFS is an external entity, the doubt of trust is always there. Hence, it is mandatory to encrypt data before outsourcing the storage on IPFS. Here, different image encryption techniques for encrypting Covid-19 medical images are reviewed.

Classical cryptosystems AES—Advanced Encryption Standard and DES—Data Encryption Standard which is block-based symmetric key systems and asymmetric methods like RSA—Rivest Shamir Adleman consume more time. On the other hand, while applying RSA to encrypt images, it creates many security issues as defined in [32]. The comparison of classical symmetric ciphers and similar ciphers is given in Table 1 as below:

However, because of the bulk capacity of the image, the above and similar methods take a very high time for encryption. Image cryptosystem based on chaotic maps [33–36] is popular because of easy implementation and effective security levels. These techniques use the Fridrich model. Fridrich used two stages—Confusion and Diffusion. Confusion is also known as Permutation. In confusion, pixels are shuffled, and in diffusion, pixels are modified. For increasing the security level of the image cryptosystem, permutation and diffusion are executed for $\mathcal{M}$ and $\mathcal{N}$ times, respectively. For permutation and diffusion, the required keystreams are produced using chaos-based functions. The different kinds of chaotic systems like 1-D, 2-D and more dimensions can be applied in different methodologies. To generate complicated, chaotic order, more dimensions are needed. Ultimately, the guessing of chaotic nature is very tough. On the other hand, the development cost of these higher dimension methodologies increases due to their high complexity [37]. The chaos functions such as Henon, Logistic, Arnold cat, Baker

Table 1 Comparison of symmetric ciphers

Sr. No	Scheme	Key size	Time complexity
1	DES	64	$O(2^{64})$
2	Tripple DES	128	$O(2^{128})$
3	AES	128, 192, 256	$O(2^{128}), O(2^{192}), O(2^{256})$
4	IDEA	128	$O(2^{128})$
5	KASUMI	128	$O(2^{128})$
6	SNOW	128	$O(2^{128})$
7	ZUC	128	$O(2^{128})$

are mostly useful in image cryptosystem. Apart from these standard maps, researchers also have used their logic to generate chaotic behavior values. Cao and Mao [38] proposed a 2-dimensional infinite collapse maps using *sin*() function. Mansouri and Wang [39] proposed a one-dimensional sine-powered chaotic map that uses a sine map and two control values. On the other hand, to decrease the rounds of confusion–diffusion and to achieve more security, bit-level permutation and diffusion techniques are used in [35, 36]. Rubik's cube movement logic is used to perform confusion of image pixels in [40]. During diffusion, the key generation process involves prime factorization. Zhou and Wang [41] used a 5-D conservative hyper-chaotic system to generate pseudo-random series in the confusion–diffusion phases. But, as the dimensions increase, the complexity also increases. Parallel computing is also applied in [42] to encrypt a group of images preemptively by utilizing the processors of a modern computer system. [43] proposed a method using chaos, and the generated keystream depends on an input image. Depending on methodology the comparison of some existing schemes is given in Table 2 as below:

As discussed, the above approaches and many more are used by researchers for keystream generation and image encryption; each methodology has advantages along with limitations. The proposed Covid-19 record cryptosystem is based on Fridrich architecture that uses keystreams generated from a 1-D logistic map for pixel confusion. The advantage of a 1-D logistic map is that it is faster and less complex than other high-dimensional chaotic maps. Cryptographic hash function SHA-256 is an irreversible function and deterministic. Using this SHA, a stream of unique values is easily generated through simple logic. Hence, in the proposed system the diffusion keystream is generated through SHA to increase the speed.

In the proposed system, the focus is given on checking the passengers' Covid-19 status for the travel ticket issuing authority and secure sharing of Covid-19 records among medical practitioners. The status checking system depends on the Covid-19 database. The Covid-19 database maintains the details of each Covid-19 patient and is accessible throughout the world using the Internet. The system uses two servers: cryptomatch server (CMS) and IPFS server. The CMS is a trusted entity and does the encryption and decryption of Covid-19 records. Additionally, it uploads and accesses encrypted records from IPFS. The IPFS stores the file and returns the unique hash value for that file. The uploaded file is downloaded using returned hash value and is used to further access the uploaded file. CMS maintains the Index Table for record-keeping. The index table contains four attributes—1. Patient Id; 2. IPFS returned hash value; 3. Covid-19 records encryption key value (Seed); and 4. Status. Patient Id can be a unique identification number like social security number or Aadhaar number (India). To check the status of the passenger, the *Status* field is accessed from Index Table, and only if the status is either *norecordfound* or *Recovered* then the ticket will be issued. Otherwise, the ticket will not be issued and immediately reported to the police. The proposed system is designed using password-based secure authentication, image cryptosystem, IPFS, cryptographic hashing, and Paillier cryptosystem. The key value (Seed) required for generating keystream is calculated from Patient Id. *Seed* is encrypted using the Paillier cryptosystem to provide higher security.

Table 2 Comparison of existing cryptosystems

Sr. No	Scheme	Methodology	Limitations
1	Fridrich [44]	Permutation-3D map, pixels and gray levels, Diffusion-Use Non-linear feedback shift register	Choice of cipher key depends on block size, requires padding, and increases transmission load
2	Lian et al. [45]	Corner-pixels confusion, Diffusion-XOR operation utilized	Fully dependent on one-time key
3	Zhu et al. [46]	Permutation - Bit-level, Arnold cat map, Diffusion-Logistic map	Practical after 5 rounds, High computational power, and Small key space
4	Zhu, Zang [46]	Permutation of upper 4 bits individual and lower 4 bits as single entity	Small keyspace, Vulnerable to differential attack
5	Teng et al. [35]	Binary bitplane division,confused only 4 higher bit-planes by coupled map lattice (CML) system and for diffusion logistic map used	High computational power, Small key space
6	Li et al. [33]	Pixel level, bit-level permutation , 5-D multi-wing hyper-chaotic system	High computational power, Small key space
7	Azimi et al. [47]	Pair chaotic coupled map and DNA encoding rule	Vulnerable to differential attacks

In the world, so many countries are developing countries. Because of the high population and lack of enough IT infrastructure, it is difficult to manage data securely and efficiently. As Covid-19 database contains most of the data in digital images (X-ray, CT scan, MRI, etc.) form. As data are huge, cloud technology can be the solution to manage such huge data. For rapid sharing of such private data among authenticated users, centralized control is required. However, if data are stored and transmitted in plain form there will be a privacy issue. Also, researchers have observed that the cloud is honest-but-curious in nature, which means it follows the defined protocol but it may trace data. Hence, to preserve the privacy of individuals and their personal data, the proposed solution is given. The motivation behind developing the proposed system is to provide secure and efficient sharing of data for travel agencies and medical experts. Considering the Covid-19 reports containing personal details, travel history, contact history, medical diagnosis reports, medical images like X-Ray, and other required things, the secure and efficient management of these records becomes essential. With the security concern, confidentiality, integrity and availability (CIA) are focused. Confidentiality is achieved through the encryption of records. Integrity assures no change in data during transmission by an unauthorized user. Availability means the facility to the authenticated users for the management of data. Considering these CIA points, the proposed system is designed through secure authentication, image cryptosystem, and secure data storage. Further, the future usage of the proposed system is like a person with a vaccination history allowed to travel and medical data can be used for further research in the medical field.

The highlights of the proposed work are:

- Covid-19 status checking system for travel ticket issuing authority is proposed.
- Rapid sharing of Covid-19 records among medical practitioners is done for telediagnosis.
- A novel password-based authentication protocol is designed to protect passwords in ADT against offline attacks.
- Random order keystream required for proposed image encryption system is generated using cryptographic hashing (SHA-256).
- A novel image cryptosystem is proposed for encrypting Covid-19 records.

Preliminaries

In this section, the fundamentals of the proposed system are briefly described. Paillier cryptosystem is used for encryption of *Seed*. The chaotic map is used to generate keystream used in confusion round of image encryption. The novel storage mechanism IPFS is also discussed briefly.

Paillier Cryptosystem

To understand Paillier method, assume U and V are two large prime numbers and assume $R = U \times V$. Assume *Encrypt*() and *Decrypt*() are the cryptofunctions with public key *e PK* and secret key *d SK*, respectively, and is given by (R, g), where g is a generator in $Z^{*}{}_{R^2}$. For two input values $k_1, k_2 \in Z_N$, the Paillier method has the following properties:

(1) Homomorphic add

$$Decrypt(Encrypt(k_1 + k_2)) \\ = Decrypt(Encrypt(k_1) * Encrypt(k_2) mod\, R^2) \tag{1}$$

(2) Homomorphic multiply

$$Decrypt(Encrypt(k_1 * k_2)) \\ = Decrypt(Encrypt(k_1)^{k_2} mod\, R^2) \tag{2}$$

Chaotic Function

A chaotic function or map is a mathematical function that produces chaotic order values. The input for the function is either discrete or continuous type value. In dynamical systems study, chaotic functions are frequently used. In the proposed system, a key is produced using a Logistic map.

Logistic Map

This map can generate chaotic behavior using:

$$d_{n+1} = \mu d_n(1 - d_n) \tag{3}$$

where d_n is a number between (0,1) which denotes the ratio of the existing population to the maximum possible population. In (3), μ is a control parameter that lies between (0,4] and even further decreased to [3.57,4] as that's the range of actual chaotic behavior.

InterPlanetary File System (IPFS)

IPFS technology is used to store and retrieve data, files (text, audio, video, image) in a distributed manner. IPFS is a distributed peer-to-peer (p2p) storage network. More specifically, it is a hypermedia distribution protocol in which contents are accessed as distributed identities. IPFS aims to make the web more secure, faster, and more open. Contents are accessible through peers, and the peers can be located anywhere in the network.

IPFS does the following whenever a file is uploaded:

1. For a given file, a unique cryptographic hash is returned. 2. IPFS removes duplicate data throughout the network. 3. Each node in the network stores only the data it is interested in, and required metadata is used for indexing purposes. 4. Whenever the user wants a file, the file's hash is given to ask the network to retrieve the content behind that file's hash.

As IPFS is a distributed storage system, it chunks a single file into multiples and distributes it among various nodes. So, complete file contents are not available at a single node. Moreover, IPFS uses transport encryption but not content encryption. This means that data are secure when being sent from one IPFS node to another. In the proposed system context, the complete file is not available at a single node; this is considered to be safer.

Proposed System Design

Considering the proposed system, the Covid-19 report consists of all the details like personal details, travel history, contact history, medical diagnosis reports, medical images like X-Ray, and they are considered in image form (scanned documents) only. So, every record/report is converted to an image and used in the proposed architecture.

The proposed system consists of various participants, cryptomatch server, IPFS, and it is developed using secure authentication protocol and image cryptosystem. The system architecture is shown in Fig. 1. The working details of the proposed system are described through the following subsequent subsections.

Participants of the Proposed System

Figure 2 shows the participants involved in the proposed system. There are two types of participants, i.e., primary and secondary. Primary participants are active participants handling the Covid-19 cases directly. Secondary participants are the passive participants who can view the status of any citizen from the Covid-19 database. Covid-19 patients are people undergoing treatment. Medical Experts are the doctors diagnosing Covid-19 patients. The role of the Research laboratory is to perform various required tests like PCR (polymerize chain reaction) tests and updating of records. The duty of the government is to trace out travel history, contact people, quarantine required people, etc.

Primary participants can create, view, and update the Covid-19 patients' records. The medical record consists of a sequence of images showing patients' reports and concerned medical images.

Secondary participants also play an important role in stopping Covid-19 from spreading. These types of participants include all ticket issuing authorities like railway, bus transport, flight service agencies as well as private vehicle transport service agencies. These participants have the right to view only the Covid-19 status of the person.

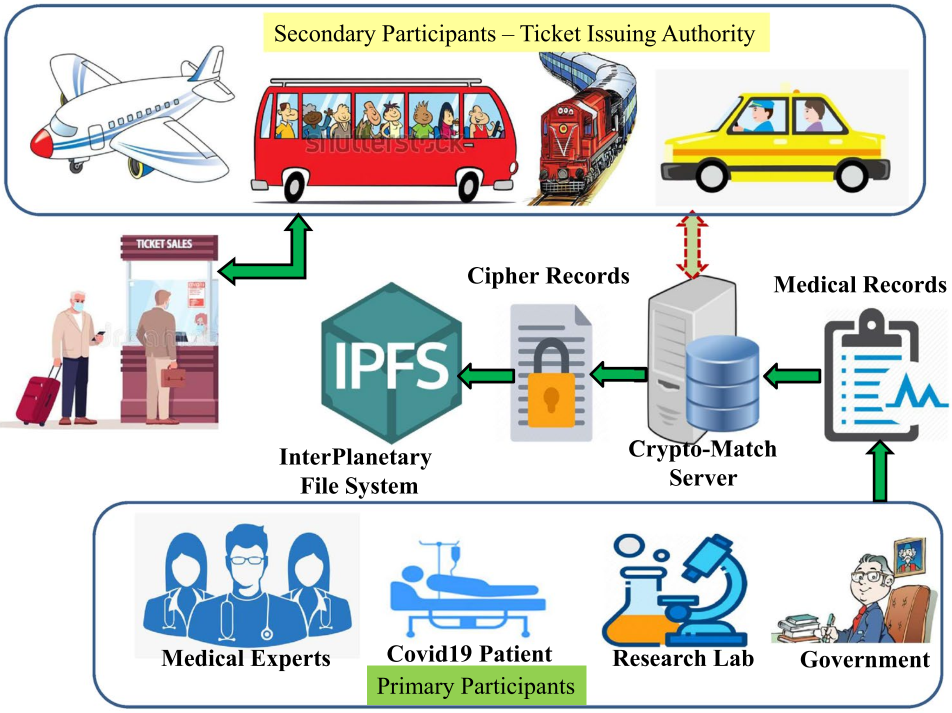


Fig. 1 System architecture

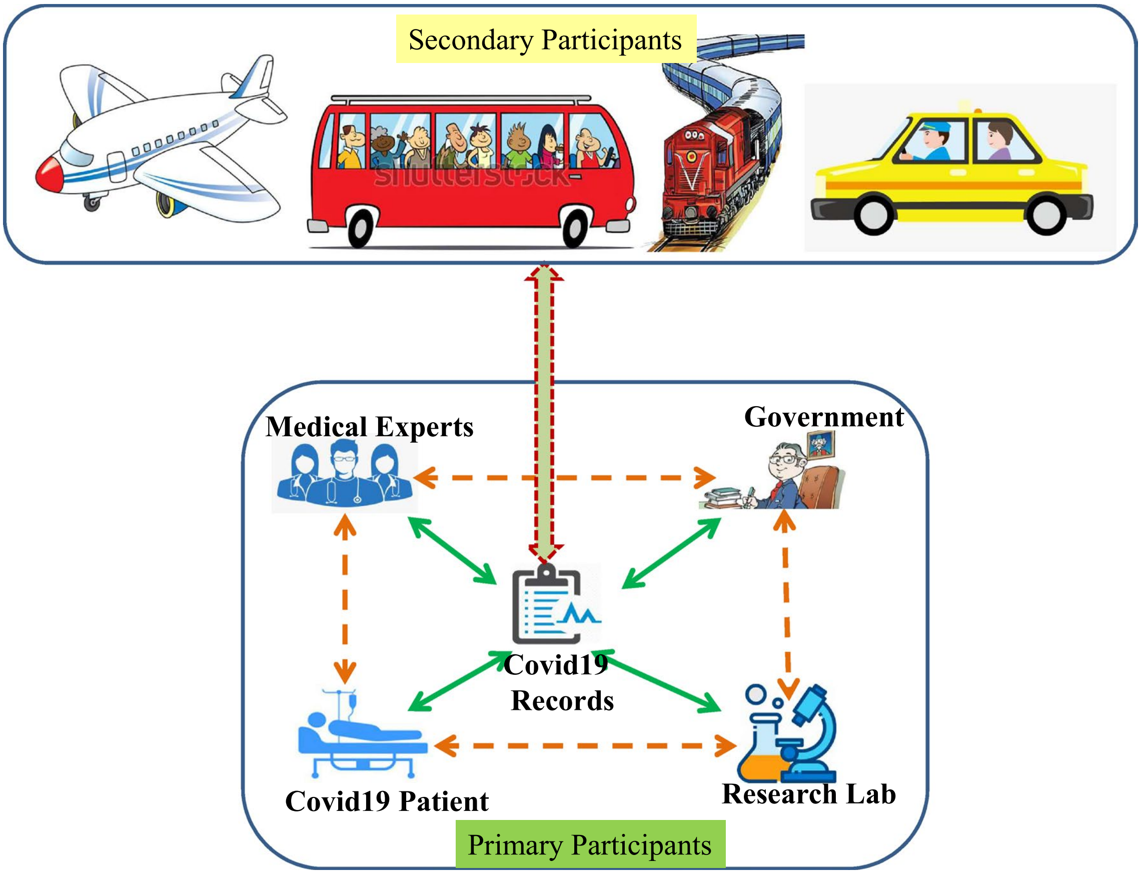


Fig. 2 Participants of the proposed system

A representative from these participants has to access the Covid-19 record and check the status of the citizen using a unique id like *AADHAAR* whether the person is infected or not. If not infected, then the corresponding person is allowed to continue their activities like traveling, joining the school, offices, etc. If the status is *active*, then they have to inform the government immediately to avoid further spreading.

Cryptomatch Server

Cryptomatch server (CMS) is the trusted entity. It permits authenticated users to access the system through authentication. Also, it does encryption and decryption of medical records. *CMS* also uploads the encrypted medical records to the IPFS and accesses the medical records as per the requirement. In addition, the *CMS* maintains the Index table and Authentication Data Table (ADT). The index table is used to keep track of all Covid-19 information. ADT maintains the user information, and the details are described in the next subsequent subsection. The attributes of the Index table are: 1. Patient Id; 2. Hash returned from IPFS for the encrypted medical records; 3. Paillier encrypted key value (Seed) and 4. Patient status—Active or Recovered. *Seed* value is calculated using,

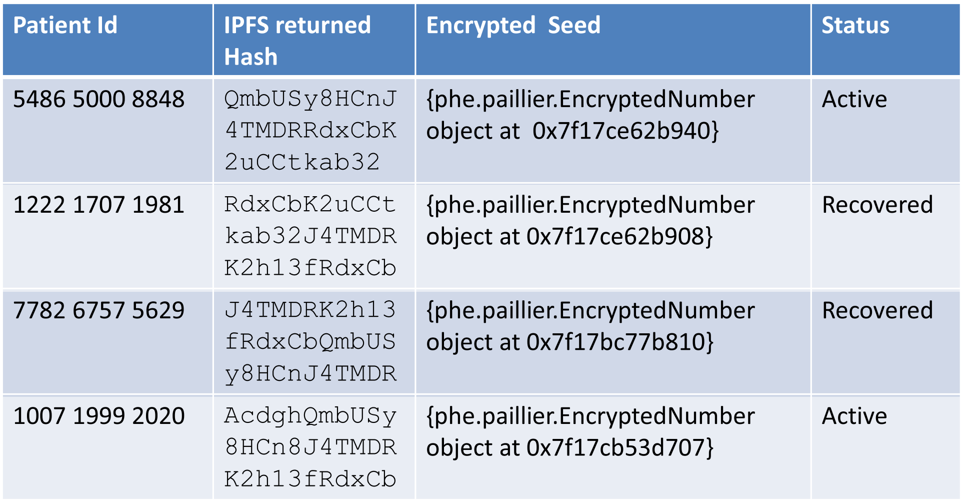
$$Seed = \sqrt{Hash(patientId) * c} \tag{4}$$

where c is the floating point constant derived from hash of the first image of the corresponding patient's Covid-19 record.

Figure 3 shows sample index table.

Proposed Authentication

Authentication is the first step to access any service through which the identity of authenticated users is verified. Here, for simplicity purposes, conventional user registration and authentication procedure are not considered. The focus is given on the contents of the Authentication Data Table (ADT). The attributes of ADT are UserId and Password. Concerning the security point of view, the contents of ADT should not be in plain format. In the proposed system, the ADT contains SHA-256 hashed UserId and AES-256 Encrypted Password.



Patient Id	IPFS returned Hash	Encrypted Seed	Status
5486 5000 8848	QmbUSy8HCnJ4TMDRRdxCbK2uCCtkab32	{phe.paillier.EncryptedNumber object at 0x7f17ce62b940}	Active
1222 1707 1981	RdxCbK2uCCtkab32J4TMDRK2h13fRdxCb	{phe.paillier.EncryptedNumber object at 0x7f17ce62b908}	Recovered
7782 6757 5629	J4TMDRK2h13fRdxCbQmbUSy8HCnJ4TMDR	{phe.paillier.EncryptedNumber object at 0x7f17bc77b810}	Recovered
1007 1999 2020	AcdghQmbUSy8HCn8J4TMDRK2h13fRdxCb	{phe.paillier.EncryptedNumber object at 0x7f17cb53d707}	Active

Fig. 3 Sample index table

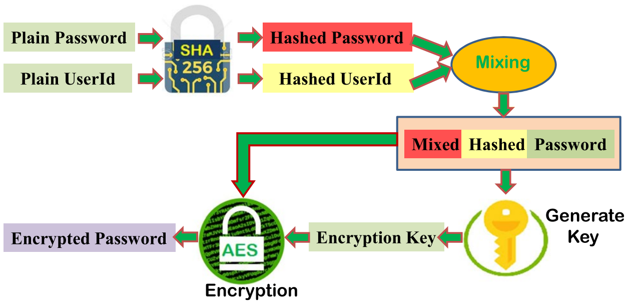


Fig. 4 Encrypted password generation

Encrypted Password Generation

Figure 4 shows the encrypted password generation method and is described through the following steps.

1. Initially, during the registration process, given UserId and Password are hashed using SHA-256.
2. Hashed UserId and Hashed Password are concatenated to get the 512-bit string.
3. Using logistic map 3, *ShufflingIndex*1 of size 512 is generated.
4. The obtained 512-bit hash string is shuffled according to *ShufflingIndex*1, and the result is called as *MixedHashPassword*.
5. Once again *ShufflingIndex*2 of size 512 is obtained by executing logistic map 3.
6. First 256 locations of the *ShufflingIndex*2 are considered as *KeyIndex*.
7. 256-bits of *MixedHashPassword* of the locations from *KeyIndex* are grouped. This selected 256-bit substring is acting as the AES-256 encryption key.
8. Using this encryption key, *MixedHashPassword* is encrypted and stored as an Encrypted password in ADT.

Table 3 illustrates the flow of encrypted password generation. Forsake and simplicity, and the intermediate values are represented in hex only.

Table 3 Illustration of encrypted password generation

Plain userId	Sachin
Plain password	Rajanikant2020
Hashed userId	964fa766292d50c62019ed7d33232559c0af511cddd5e55938aa845698f771b9
Hashed password	920d072d49008cbb3f32d0e830f4640ac4b7620271f8e3863b1da700c3f600f6
Concatenated hashed UserID and password	964fa766292d50c62019ed7d33232559c0af511cddd5e55938aa845698f771b9 920d072d49008cbb3f32d0e830f4640ac4b7620271f8e3863b1da700c3f600f6
Mixed hash password	777dab9bd21b506fca343ec2966a1019249d4134b60f7083adffc29ed42299d8 d0e3390f7c87e636c78f607600586c3980300558d02253f4f261d550a21bd50a
Encryption key	80f0703ce1f206d0ed488afd09399907b26798065033272e8214bce26514a4b3
Encrypted password	jYQzq1tCBEpqdlyzYtAwIu4WMMqlaQT+QbfWRvosVnH2Ai MEnrGxN+Vs6ArMo850TeU8IAImfzDzMGk3YKvi16En8/xTs LoQ4FMPbRccjsNluhzoRLRPKNfbgIyOxcZ56Fcr7jwXjEkgIB8 ZGg/NTj1ccj8D81vWWtbZxMYqpp6CwyM0Ef6FgbSvRc4ockglv

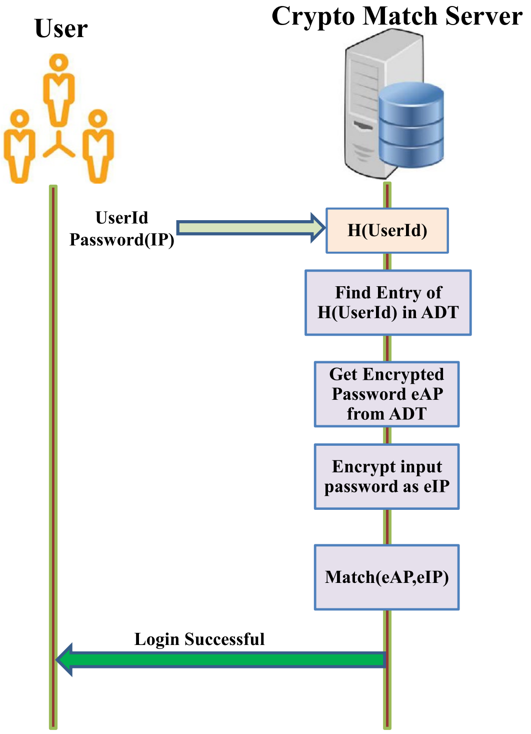

User
Crypto Match Server
UserId
Password(IP)
H(UserId)
Find Entry of H(UserId) in ADT
Get Encrypted Password eAP from ADT
Encrypt input password as eIP
Match(eAP,eIP)
Login Successful

Fig. 5 Authentication

Authentication

The authentication procedure is described in Fig. 5. The user is the participant (primary and secondary) of the system, and steps are given below:

1. Initially, user inputs the parameters *UserId*, *Password*. Then, these parameters are sent to the Cryptomatch server.
2. Cryptomatch server calculates hash value of *UserId* as *H*(*UserId*) using SHA-256 hashing technique.
3. Cryptomatch server searches ADT for input *H*(*UserId*). If desired *H*(*UserId*) is found, it reads the encrypted password from ADT as eAP. Otherwise, go to step 7.
4. Input password is encrypted as eIP using methodology described previously.
5. Cryptomatch server reads the encrypted password from ADT as eAP and compares eIP and eAP.
6. If eIP and eAP are the same, it returns a positive acknowledgment to the client.
7. Otherwise, it returns a negative acknowledgment to the client.

Encryption of Medical Record

Covid-19 Patient's medical record contains personal information, travel history, contact history, and medical reports, along with corresponding medical images. All these records are converted to images. Next, these images are encrypted using a novel image encryption system. The proposed methodology involves Random Keystream Generation using SHA-256. NIST Randomness Tests are performed to check the randomness of generated keystream. It also involves image encryption along with confusion and diffusion procedures.

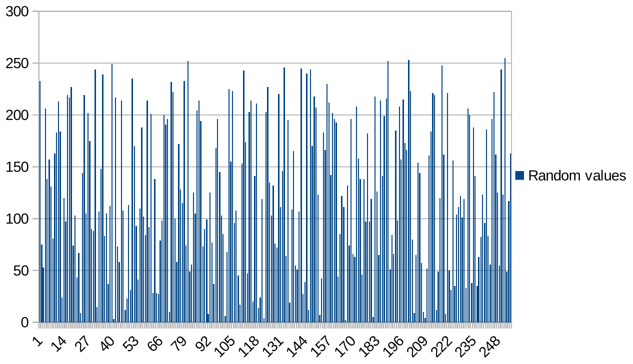


Fig. 6 Distribution of random values

Secure Hash Algorithm (SHA) 256 Based Random Keystream Generation

An n-bit cryptographic hashing function is a mapping of random size messages into n-bit hash values. It is one-way, i.e., not reversible and collision-resistant. These hashing functions are used in different security systems like digital signatures, blockchain technology, password security, authentication of the message, etc. In SHA-256, the message hash is generated in the following way:

(1) The message is appended with its size to obtain a message of multiple of 512 bits long.
(2) Then it is parsed into 512-bit message groups M_1, M_2 ,...,M_N. The message groups are processed one at a time: Starting with a fixed default hash value $Hash(0)$, serially

$$Hash(i) = Hash(i-1) + C_{Mi}(Hash(i-1)) \tag{5}$$

is computed. In (5), C is the SHA-256 compression function and + is word-wise modulo 232 addition. $Hash(N)$ is the hash of message M. The above hashing technique is used to produce keystream in following way

$$H_{i+1} = Hash(H_i) \tag{6}$$

given that $H_0 = Hash(Seed)$.

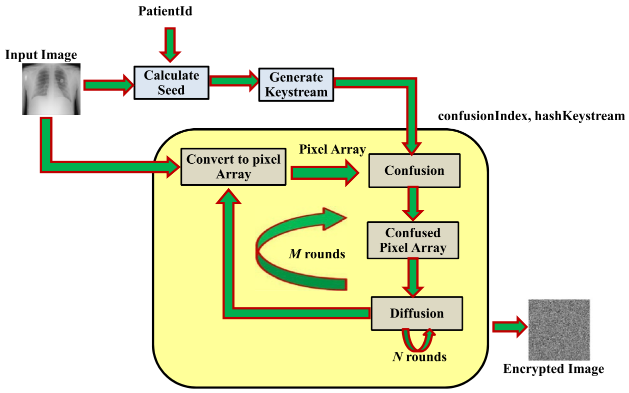
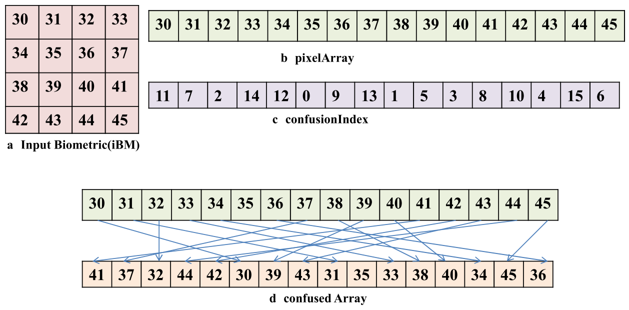


Fig. 7 Medical image encryption



30	31	32	33
34	35	36	37
38	39	40	41
42	43	44	45

a Input Biometric(iBM)

30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

b pixelArray

11	7	2	14	12	0	9	13	1	5	3	8	10	4	15	6

c confusionIndex

30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
41	37	32	44	42	30	39	43	31	35	33	38	40	34	45	36

d confused Array

Fig. 8 Confusion process

The function is tested to generate random keystream of length 256. For sample, here $H_0 = 35423421$ is taken and generated key value is applied a modulus of 256 and appended to keystream. Figure 6 shows the distribution of random keys.

Image Cryptosystem

Figure 7 shows the architecture of the proposed Covid-19 image cryptosystem. The proposed methodology uses Fridrich's confusion–diffusion architecture. Confusion means shuffling of pixels and diffusion means modifying pixel values. Chaotic maps are commonly used to generate keystream of chaotic order values for permutation and diffusion. In the proposed system, 1-D Logistic map (3) is used to produce a keystream of confusion. For generating keystream required for diffusion, a novel method using cryptographic hash function SHA-256 is implemented as described in (6).

Initially, plain image is converted into 1D *pixelArray*. Using Logistic map, *confusionIndex* is produced and pixels are shuffled accordingly into *confusedArray*. The *confusionIndex* is based on input values of μ and d_n which are calculated from *Seed*. The *pixelArray* is confused to obtain *confusedArray* as follows:

$$confusedArray \leftarrow confusion(pixelArray, confusionIndex) \tag{7}$$

Figure 8 illustrates the confusion process for given 4 × 4 input image.

Algorithm 1 ImageEncrypt

```
function ImageEncrypt($iImage, c, patientId$ )
    $iHash \leftarrow Hash(patientId)$
    $Seed \leftarrow sqrt(iHash * c)$
    $\mu, d_n, \leftarrow controlValues(Seed)$
    $\mathcal{M} \leftarrow no\_of\_confusionRrounds$
    $\mathcal{N} \leftarrow no\_of\_diffusionRounds$
    $hashKeyStream \leftarrow generateHashKeyStream(Seed)$
    $confusionIndex \leftarrow Logistic(\mu, d_n)$
    $pixelArray \leftarrow toArray(iImage)$
    $outert \leftarrow 1$
    while $outer \leq m$ do
        $confusedArray \leftarrow confusion(pixelArray, confusionIndex)$
        $inner \leftarrow 1$
        while $inner \leq n$ do
            $h \leftarrow hashKeyStream[inner]$
            $ci \leftarrow confusedArray[inner]$
            $ci1 \leftarrow confusedArray[inner - 1]$
            $diffusedArray \leftarrow (ci + ci1) \oplus h$
            $inner \leftarrow inner + 1$
        end while
        $outer \leftarrow outer + 1$
    end while
    $eImage \leftarrow toImage(diffusedArray)$
end function
```

Algorithm 2 ImageDecrypt

```
function ImageDecrypt($eImage, eSeed$ )
    $Seed \leftarrow Decrypt(eSeed)$
    $\mu, d_n, \leftarrow controlValues(Seed)$
    $\mathcal{M} \leftarrow no\_of\_confusionRounds$
    $\mathcal{N} \leftarrow no\_of\_diffusionRrounds$
    $hashKeyStream \leftarrow generateHashKeyStream(Seed)$
    $confusionIndex \leftarrow Logistic(\mu, d_n)$
    $pixelArray \leftarrow toArray(eImage)$
    $confusedArray \leftarrow pixelArray$
    $outert \leftarrow 1$
    while $outer \leq m$ do
        $inner \leftarrow 1$
        while $inner \leq n$ do
            $h \leftarrow hashKeyStream[inner]$
            $ci \leftarrow confusedArray[inner]$
            $ci1 \leftarrow confusedArray[inner - 1]$
            $diffusedArray \leftarrow (ci - ci1) \oplus h$
            $inner \leftarrow inner + 1$
        end while
        $confusedArray \leftarrow confusion(pixelArray, confusionIndex)$
        $outer \leftarrow outer + 1$
    end while
    $Image \leftarrow toImage(confusedArray)$
end function
```

Next, *hashKeyStream* is generated using 6 and *diffusedArray* is found as follows:

$$diffusedArray[i] \leftarrow (confusedArray[i] \oplus confusedArray[i-1]) \oplus hashKeyStream[i] \quad (8)$$

Algorithm-1 *ImageEncrypt* describes the logic of encryption. The functions used for image encryption are reversible. Algorithm-2 *ImageDecrypt* describes the logic of decryption. Figure 9 illustrates the diffusion process for given *confusedArray* and *hashKeyStream*.

Finally, *diffusedArray* is transformed into 2-D to get encrypted image *eImage*. Permutation and diffusion rounds are ran for $\mathcal{M}$ and $\mathcal{N}$ times to increase security level. The sample input images and corresponding encrypted images are shown in Fig. 10.

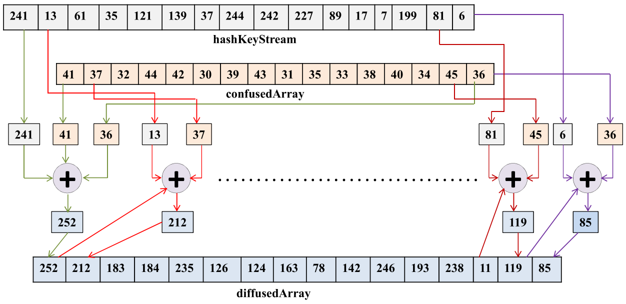
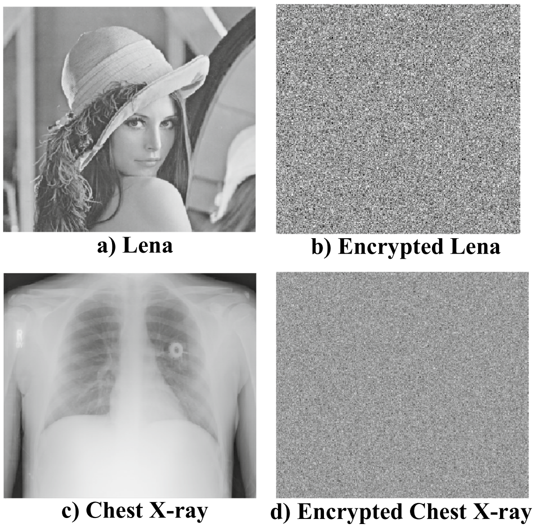


Fig. 9 Diffusion process

Uploading Files to IPFS

First, command *ipfs daemon* is used to connect the user's PC to the network. Next, after initialization of *ipfs daemon*,


a) Lena b) Encrypted Lena

c) Chest X-ray d) Encrypted Chest X-ray

Fig. 10 Sample input images and corresponding encrypted images

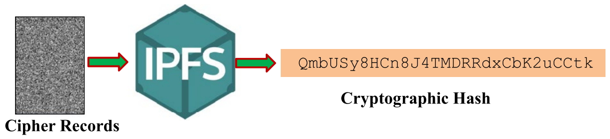


Fig. 11 Uploading file to IPFS

Table 4 NIST randomness test

S. No	Test	$\mathcal{P} - value$	Outcome
1	Frequency	0.786582	Passed
2	Runs	0.499825	Passed
3	Longest runs of ones	0.100325	Passed
4	Block frequency ($\hat{\mathcal{M}} = 20,000$)	0.657610	Passed
5	Non-overlapping templates ($\hat{\mathcal{M}} = 9$)	0.801707	Passed
6	Overlapping templates ($\hat{\mathcal{M}} = 9$)	0.110511	Passed
7	Serial ($\hat{\mathcal{M}} = 16$)	0.456287	Passed
8	Rank	0.527891	Passed
9	Approximate entropy ($\hat{\mathcal{M}} = 10$)	0.783215	Passed
10	Linear complexity ($\hat{\mathcal{M}} = 500$)	0.908556	Passed
11	Maurers universal	0.624582	Passed
12	Spectral DFT	0.705782	Passed
13	Cumulative sums	0.602451	Passed
14	Random excursions variant	0.526478	Passed
15	Random excursions	0.172565	Passed

Connection object is created. Using add(.) and get(.) methods of the *Connection* object files are uploaded and retrieved. When the user uploads the file to IPFS, it returns the hash of files and objects using the Multihash format of Base58 encoding of length 32 bytes. In the proposed system, all encrypted medical record images are stored in a folder, and then this folder is uploaded to IPFS. In response, IPFS returns the single hash value, which is stored in the Index table of the corresponding patient. Figure 11 shows the uploading of the file to IPFS.

Security and Performance Analysis

For the simulation of the proposed system, two personal computers cryptomatch server and IPFS server, are used. The cryptomatch server has an Intel Core i5-4570, 3.20GHz 4 processor, 8 GB RAM, and Ubuntu 16.04 LTS OS. The configuration of IPFS is a computer with Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz processor, 32 GB RAM, Ubuntu 18.04.3 LTS. For the implementation of the proposed system, XAMPP with Apache HTTP Server, MySQL ver 14.4 Distrib 5.7.29, and PHP are used. Encryption algorithms are coded in Python 3.6.3. Traditional image processing gray images from the USC SIPI database and medical images from NIH (National Institutes of Health) are taken to analyze the performance. Apart from these databases additional two Covid-19 databases are used for performance analysis are used namely

Table 5 Correlation analysis

S. No	Input	Correlation-input image			Correlation-encrypted image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
1	Lena	0.975966	0.986136	0.964092	– 0.003421	– 0.014752	0.006829
2	Girlface	0.983540	0.988184	0.972349	0.006011	– 0.030665	– 0.037308
3	Clown	0.981210	0.991339	0.975585	– 0.005675	0.022084	0.007481
4	Tank	0.882408	0.873433	0.846698	– 0.027938	– 0.018191	0.011577
5	Truck	0.961921	0.918158	0.907906	0.024098	– 0.010734	0.006271
6	Cameraman	0.905928	0.949359	0.884538	0.003158	– 0.008491	– 0.015892
7	Chest-1	0.998473	0.995108	0.993790	0.012284	– 0.012025	– 0.022124
8	Chest-2	0.998075	0.998046	0.996210	– 0.001896	0.017078	– 0.072145
9	b-f00163	0.970417	0.979807	0.996119	– 0.028031	0.011051	0.012567
10	b-f00175	0.925112	0.983456	0.975211	0.001224	0.011018	– 0.089010
11	b-f00181	0.606689	0.962907	0.369601	-0.003044	0.001870	0.060964
12	b-m00167	0.695605	0.978057	0.654010	– 0.001057	0.02106	– 0.062105
13	b-m00169	0.861112	0.993331	0.871523	0.001981	0.012022	0.020155
14	b-m00171	0.967657	0.986281	0.990064	0.017708	0.015631	– 0.015325
15	e-17524-1-1	0.998022	0.995676	0.996321	– 0.008117	0.017194	– 0.018572
16	e-17543-1-1	0.978211	0.979447	0.953223	0.009618	0.007018	– 0.001421
17	e-17605-1-1	0.872265	0.804989	0.678817	0.057196	– 0.010098	– 0.081245
18	e-17611-1-1	0.982431	0.996544	0.945067	0.009532	– 0.067598	– 0.085690
19	e-17637-1-1	0.988278	0.979665	0.979723	– 0.001265	0.081639	– 0.06743
20	e-17531-1-1	0.959874	0.990712	0.920832	– 0.009647	– 0.015094	– 0.007410

1.COVID-19 British Society of Thoracic Imaging database, 2. Eurorad COVID-19 cases. These databases are provided by The European Institute for Biomedical Imaging Research (EIBIR).

Here, the performance of the image cryptosystem is analyzed in the following subsections.

NIST (National Institute of Standards and Technology) Randomness Test

The NIST [48] has proposed a statistical test suite to test the randomness of the binary sequence using 15 distinct subtests. NIST uses the threshold level $\alpha = 0.01$ for randomness test. The $\mathcal{P}$-value is calculated for all binary sequences for each subtest individually. If obtained $\mathcal{P}$-value $\geq \alpha$ then a binary sequence is passed the randomness subtest. NIST gives two methods for interpreting test results [48]:

(i) Calculation of the pass rate $(\mathcal{P}_r)$ of passing sequences in the subtest. If $(\mathcal{P}_r)$ comes down outside the appropriate proportion given by

$$\overline{\mathcal{P}} \pm 3\sqrt{\frac{\overline{\mathcal{P}}(1-\overline{\mathcal{P}})}{\hat{m}}} \tag{9}$$

where $\overline{\mathcal{P}} = 1 - \alpha$, then it shows that the test sequence is not sufficiently random.

(ii) To calculate the $\mathcal{P}$-value distribution and to check $\mathcal{P}$-values are within the range [0, 1] and to verify whether it is uniform or not.

The range [0, 1] is partitioned into 10 equal subranges and $\mathcal{F}_{kq}$ denotes the frequency of $\mathcal{P}$-values falling within the k^{th} subrange. Let $\hat{\mathcal{M}}$ be the sample size and the $\mathcal{P} - value$ is calculated using,

$$\chi^2 = \sum_{kq=1}^{10} \frac{(\mathcal{F}_{kq} - \hat{\mathcal{M}}/10)^2}{\hat{\mathcal{M}}/10} \tag{10}$$

and

$$\mathcal{P}\text{-}value_u = igamc\left(\frac{9}{2}, \frac{\chi^2}{2}\right) \tag{11}$$

If $\mathcal{P}$-value$_u \geq 0.0001$, a sequence could be regarded as a uniform distribution.

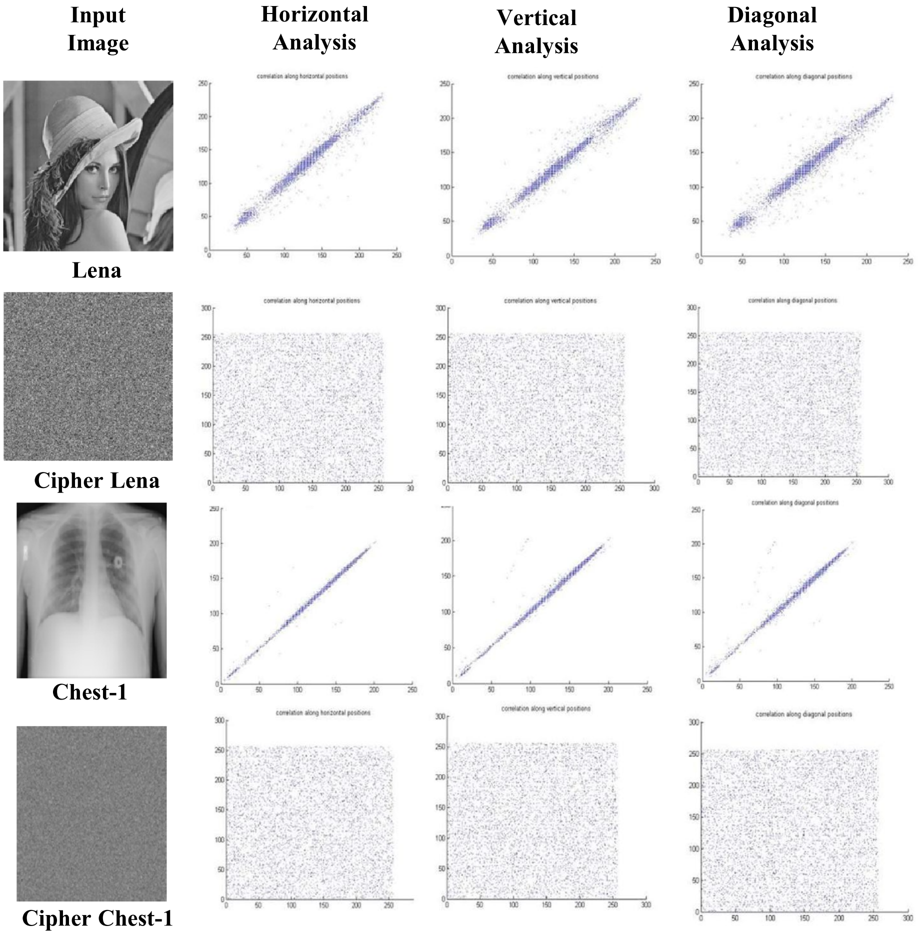


Fig. 12 Correlation analysis

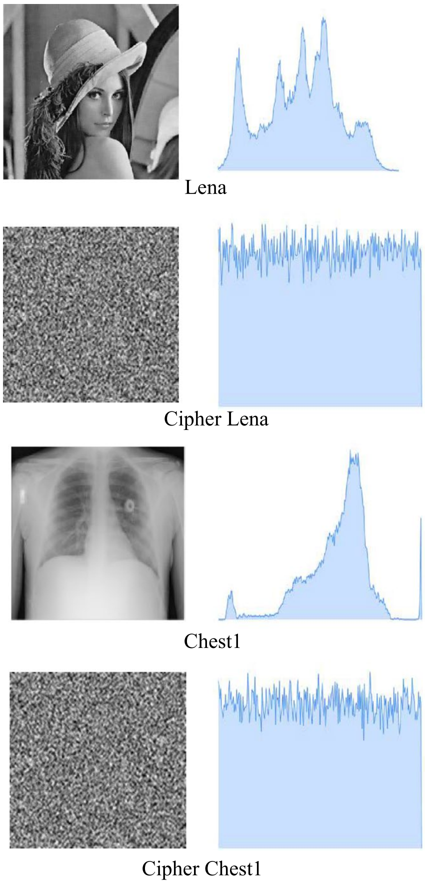


Fig. 13 Histogram analysis

For the experiment, the random number sequence of length 512 × 512 is generated using 6. Table 4 shows the results obtained from the randomness test. As seen from Table 4, the proposed hash keystream generation passes all tests. So, it can be utilized for image encryption.

Correlation Analysis

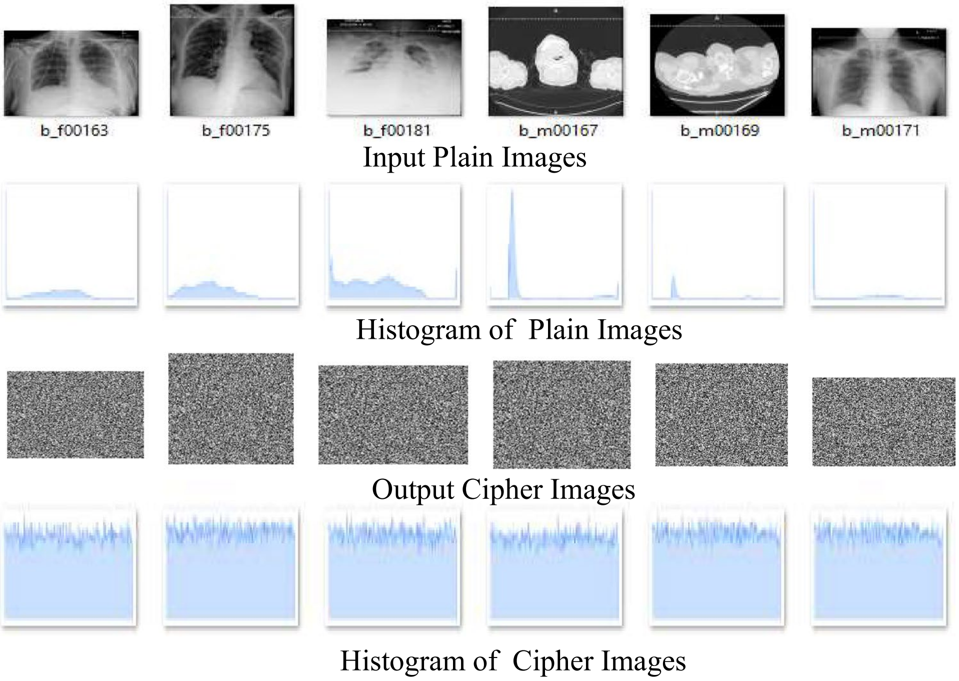
The correlation among connected pixels of the encrypted images is calculated through a random selection of 3000 samples of connected pixels from the input and cipher images separately. If correlation along vertically, horizontally, and diagonally is around zero, then the level of security of the cryptosystem is considered as good. Table 5 shows the comparison of correlation analysis of the proposed methodology for input and corresponding cipher images.

The distribution of connected pixels for the sample *lena* and *chest* − 1 and the corresponding encrypted images for the horizontal, vertical and diagonal directions are shown in Fig. 12. It shows that the pixel values of the ciphered image are spread uniformly in all directions.

Histogram Analysis

The distribution of pixels is described as a histogram. The histogram gives the statistical information about

Fig. 14 Histogram analysis-2 (COVID-19 British Society of Thoracic Imaging database)



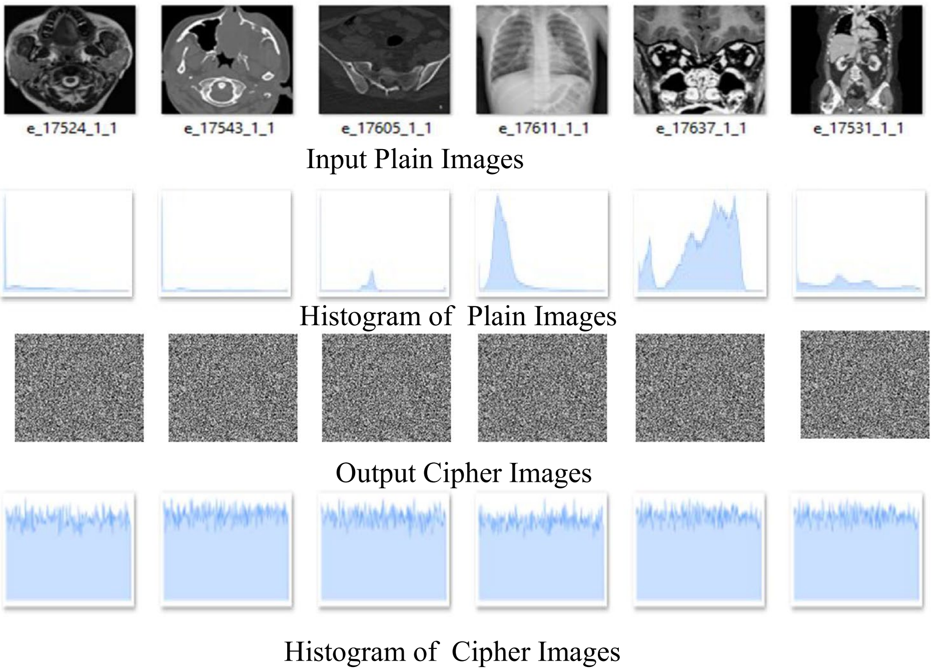


Fig. 15 Histogram analysis-3 (Eurorad COVID-19 cases)

Table 6 Differential analysis

Sr. No	Input Image	NPCR	UACI
1	Lena	99.6437	33.3762
2	Girlface	99.6087	33.0458
3	Clown	99.6125	33.3732
4	Tank	99.5950	33.5221
5	Truck	99.6117	33.3514
6	Cameraman	99.6555	33.3760
7	Chest-1	99.6117	33.4607
8	Chest-2	99.6164	33.4076
9	b-f00163	99.5055	33.5687
10	b-f00175	99.1243	34.0452
11	b-f00181	98.19014	32.6512
12	b-m00167	99.1290	33.1392
13	b-m00169	99.5560	32.8856
14	b-m00171	99.2316	33.6411
15	e-17524-1-1	99.3212	33.3451
16	e-17543-1-1	99.2350	33.5622
17	e-17605-1-1	99.1262	33.1322
18	e-17611-1-1	99.2904	33.5637
19	e-17637-1-1	99.1214	33.1200
20	e-17531-1-1	99.2743	33.5051

Table 7 Information entropy analysis

Sr. No	Input image	Entropy
1	Girlface2	7.9973
2	Lena	7.9972
3	Clown	7.9978
4	Tank	7.9975
5	Truck	7.9989
6	Cameraman	7.9948
7	Chest-1	7.9988
8	Chest-2	7.9998
9	b-f00163	7.9911
10	b-f00175	7.9968
11	b-f00181	7.9962
12	b-m00167	7.9994
13	b-m00169	7.9991
14	b-m00171	7.9959
15	e-17524-1-1	7.9988
16	e-17543-1-1	7.9978
17	e-17605-1-1	7.9992
18	e-17611-1-1	7.9991
19	e-17637-1-1	7.9993
20	e-17531-1-1	7.9990

the image. Histogram of input and encrypted *lena* and *chest* − 1 is shown in Fig. 13. Additionally, the histogram of sample COVID-19 British Society of Thoracic Imaging database and Eurorad COVID-19 cases is shown in Fig.14 and Fig.15, respectively.

It can be noticed that obtained histograms of both images are totally variant. So, it is hard to find information from statistics of pixels.

Differential Analysis

A number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are the testing methods to check the sensitivity of a little modification in the plain image as given by (12) and (14). Assume, I_1 is an input image and E_1 is the obtained encrypted image. I_2 is input image with a little modification and E_2 is the obtained encrypted image.

$$NPCR = \frac{\sum_{\alpha\beta} \psi(\alpha,\beta)}{width \times height} \tag{12}$$

$$\psi(\alpha,\beta) = \begin{cases} 0 & \text{if } E_1(\alpha,\beta) = E_2(\alpha,\beta) \\ 1 & \text{if } E_1(\alpha,\beta) \neq E_2(\alpha,\beta) \end{cases} \tag{13}$$

$$UACI = \frac{1}{width \times height} \sum_{\alpha=1}^{width} \sum_{\beta=1}^{height} \frac{|E_1(\alpha,\beta) - E_2(\alpha,\beta)|}{255} \tag{14}$$

The obtained results are shown in Table 6. From the results, it is concluded that a random one-bit change in the input image produces a major change in the encrypted image.

Analysis of Information Entropy

Information entropy measures randomness. For gray images, if pixels spread is uniform, then the highest entropy is eight. Assume $\mathcal{J}$ is the number of gray levels and $\mathcal{P}r(\mathcal{J}_i)$ is the probability of i^{th} gray level, then entropy is calculated using (15). Table 7 shows the entropy of encrypted images, and it is noticed that the values are nearer to standard entropy value.

$$\mathcal{E} = \sum_i \mathcal{P}r(\mathcal{J}_i) log_2 \left(\frac{1}{\mathcal{P}r(\mathcal{J}_i)} \right) \tag{15}$$

Analysis of Key Space

Keyspace is defined as the total number of keys used in the system. In the proposed system, a Logistic map with control values $\{\mu, d_n\}$ is used to generate a confusion index. Diffusion keystream is generated using (6), which depends on the value of $Seed$. $Seed$ is derived from a hash of Patient Id and constant c using (4). $\mathcal{M}$ and $\mathcal{N}$ are the permutation and diffusion rounds. Hence finally, key contains $\{\mu, d_n, Seed, c, m, n\}$. As per the IEEE floating-point number standard, 64-bit double datatype's computational precision is nearly 10^{-15} [49]. Hence, the obtained key space is $(10^{15})^6 = 10^{90} \approx 2^{286}$. If the

Table 8 Key sensitivity analysis

Sr. No	Parameter changed	NPCR
1	μ	99.6063
2	d_n	99.6555
3	$\mathcal{M}$	99.4638
4	$\mathcal{N}$	99.5583
5	$Seed$	99.6388

Fig. 16 Key sensitivity test

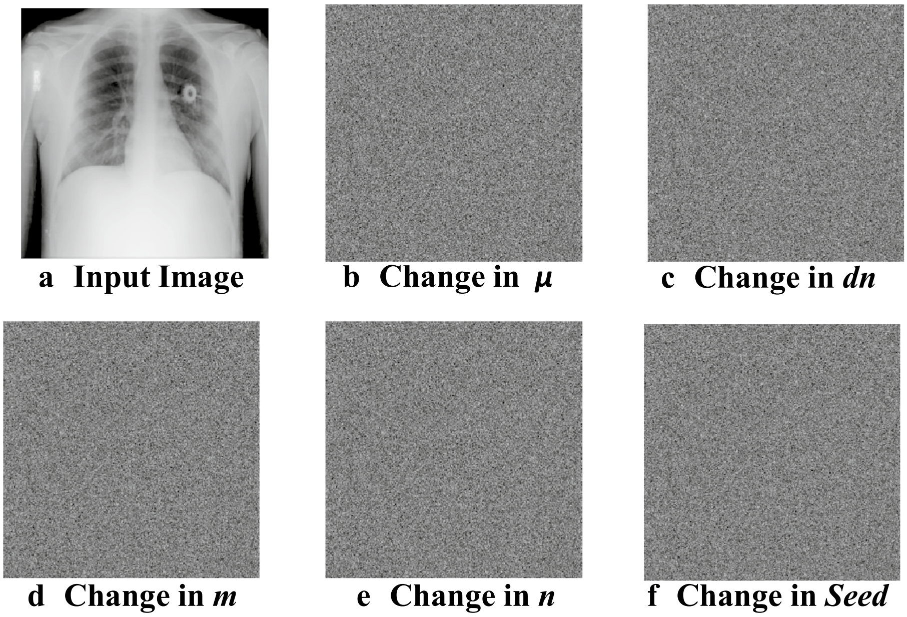


Table 9 Comparison of differential and statistical analysis of various systems

S No.	Scheme	NPCR%	UACI%	Entropy	Correlation-Cipher Image		
					Horizontal	Vertical	Diagonal
1	Abdorreza Babaei et al. [50]	99.6213	33.4197	7.9993	0.0014	0.0020	0.0012
2	Li et al. [33]	99.61	33.46	7.9972	– 0.0230	0.0019	– 0.0034
3	Xu et al. [36]	99.62	33.51	7.9974	0.00281	– 0.00183	– 0.000827
4	Hua [37]	99.6093	33.4476	7.902462	0.0013174	0.0006427	0.0019122
5	Jithin [51]	99.5700	33.8000	7.9994	-0.00116	0.00116	– 0.0043
6	Herbadji et al. [52]	99.6425	33.3827	7.9998	0.0013174	0.0006427	0.0019122
7	Proposed system	99.6437	33.3762	7.9998	– 0.004580	– 0.0015414	0.033475

Table 10 Hashed password table

Password	Password digest
Raj*2020	e9f88fd5a331b86e9b0493ad0462152298a21baa4b75741652e8ed1deadbfd6f
rediff@1981	83aaf816dbc3676335ddd54addf0ce66bdc5aa4bb44f9027493ad47906d3acb2
word@MS	dfb4a77c1b4ba0bfc65ced4d338434afc6afdc1e97f63adbefde4ec55fdee03a
indiA1947	44fe871629b617ff85534d6ded5c5930c6efe901b195100c046bc14f39535086

size of the keyspace of the cryptosystem is more than 2^{100}, then brute-force attacks are not feasible [33]. Hence, the obtained keyspace in the proposed system is enough worth to withstand attacks like brute force.

Analysis of Key Sensitivity

Here, the sensitivity of the system for small modifications in key parameters is analyzed. In the proposed encryption, $\{\mu, d_n, Seed, m, n\}$ are used as control key values to produce confusion index and diffusion keystream. Suppose I_1 is an input image, and $\mathcal{K}1$ is a set containing control key values used for encryption, and E_1 is the corresponding encrypted image. Assume a little modification is done in one of the key parameters of $\mathcal{K}1$ and new key $\mathcal{K}2$ is produced. Now, again I_1 is encrypted by $\mathcal{K}2$ and E_2 is obtained. Next, NPCR of E_1 and E_2 is obtained using (12). If obtained NPCR is around ideal NPCR, then cryptosystem is sensible for a little modification in key also.

In the proposed scheme, sample 'chest-1' image is taken as input image. At first, for the given key values, the input image is encrypted to obtain encrypted image E_1. Next, a little modification in key is done and new encrypted images E_2, E_3, E_4, E_5 are generated. The NPCR of E_1 and E_i (i={2,3,4,5}) are obtained. From Table 8, it is observed that NPCR values are around standard NPCR, which implies that the small change in any key-value produces a large change in the encrypted image. Figure 16 shows the corresponding encrypted images concerning changes in the key parameter value.

Table 11 Time complexity analysis

Sr. No	Cryptographic function	Time complexity
1	Paillier	$O(n^3)$
2	SHA	$O(n)$
3	Logistic map	$O(n)$
4	AES-256	$O(2^{256})$

Comparative Analysis of Performance

Here, the comparison of the proposed system and existing schemes is made concerning various metrics. The control values of chaotic maps are derived from the original plain image in [33]. However, the chaotic map used in [33] is five-dimensional. As dimensions increase, the complexity and cost also increase. In [36, 37], operations on bit-level are performed to increase the speed. But, the required keystreams are not derived either from the original image or from a random source which further shows the vulnerability to known/chosen plain text attacks. In [50, 51], Deoxyribonucleic Acid (DNA) sequences are used to derive the logic of pixel encoding. However, the logic is based on the rules, and this DNA rule book is required during decryption. This requires transmission of the DNA rule book to the receiver end. Herbadji et al. [52] have derived logic from the classical quadratic map to generate three chaotic sequences. Compared to the existing schemes, the proposed scheme utilizes two chaotic sequences to generate random keystreams that depend on plain images to withstand chosen/known-plaintext attacks without compromising speed.

Apart from the above comparison points, the comparison of differential and statistical analysis of different schemes is shown in Table 9. The security metrics used are NPCR, UACI, Entropy, and correlation analysis of adjacent pixels horizontally, vertically, and in diagonal directions. From Table 9, it is found that the proposed encryption scheme has achieved a better significant level in security and privacy concerning security metrics than other existing schemes.

Security of ADT

User credentials are stored in ADT. Assume that a hacker has already got access to the ADT. To reveal the passwords, the following attacks are analyzed.

(1) Bruteforce attack

Here, the attacker has to check all possible combinations of symbols to reveal the password. But, to make such attack is very hard.

(2) Dictionary attack

The attacker creates the dictionary consisting of familiar words or words of daily routine. Next, matching of words from the dictionary to ADT is done.

(3) Advanced Dictionary attack

In this attack, an attacker has to create a dictionary containing words that are predicted from users' nature of designing passwords.

(4) Lookup table attack

In this attack, initially, an attacker prepares the list of passwords. The list normally consists of regularly used passwords. Next, he constructs the lookup table containing tuples of hashed passwords and their passwords in the plain form of the prepared list. An attacker applies search operation among lookup table and ADT entries.

(5) Rainbow table attack

Rainbow tables consist of hash chains that differ from hash tables. By alternating hash function and reduction function, a series of alternating passwords and hash values are produced. Only the initial and the endmost plain value produced is entered in the rainbow table.

In all the above attacks except brute force, the attacker has to find a value in ADT with the already calculated hash value. Consider a table having plain value and its hash attributes which are already constructed as shown in Table 10. Now, an attacker has to search in the table for a given hash value. If a given match is found, a password is cracked.

In the proposed scheme, the ADT contains a hash of UserId and AES-256 encrypted hashed passwords. The encryption key is the group of symbols chosen from a particular location of the $mixed\,hash\,string$. The mixed hash string is constructed by hashed UserId and hashed Password. The length of the mixed hash string is 512, and the length of the encryption key is 256. So there are total of ${}^{512}C_{256}$ possible keys, and the attacker must know the original UserId and Password. As it is very hard to guess the original UserId and original Password, the privacy of the ADT is well preserved.

Analysis of Time Complexity

The complexity of the proposed system is based on the complexity of the Paillier cryptosystem, Secure Hash Algorithm (SHA), logistic map, the complexity of AES-256, and the complexity of the proposed image cryptosystem. For n digit number space, the time complexities are given in Table 11.

Considering the above-defined complexities, the complexity of different phases of the proposed system is given below:

(1) Seed encryption/decryption

For encryption and decryption, Paillier cryptosystem is used and the complexity is $O(n^3)$.

(2) Encrypted password generation

During new user registration, for the given UserId and Password, SHA is calculated, shuffled using a logistic map, encryption key is generated using a logistic map, and finally encrypted using AES256. Hence, the time complexity for this phase is given by: $2 \times O(n) + 2 \times O(n) + O(2^{256})$.

(3) User Authentication

For the given UserId, input encrypted password is generated as given in step 2, i.e., $2 \times O(n) + 2 \times O(n) + O(2^{256})$. Next, at CMS, an encrypted password is extracted from the ADT, and complexity is given by $O(logN)$, where N is the number of records in the ADT. Then, the complexity of matching two encrypted passwords is given by $O(1)$. Hence, the total complexity of this phase is: $2 \times O(n) + 2 \times O(n) + O(2^{256}) + O(logN) + O(1)$.

(4) Image Encryption and decryption

The complexities of image encryption and decryption are equal. Let P be the plain image of width w and height h. Also, let m be the number of confusion rounds and let n be the number of diffusion rounds. Then, the complexity of proposed image cryptosystem is given by:$T(encryption) = T(decryption) = O(mnwh)$.

Analysis of Known/Chosen Plain Text Attack

In chaotic encryption, the attacker mainly gives focus to finding the intermediate values instead of finding the chaos function arguments used for key [49]. To obtain these values, the proposed encryption is broken into four sections. If *iImage* is an input image is converted into 1D *pixelArray*. If *confusionIndex* is obtained from logistic map using (3) then 1^{st} section of confusion is given by,

$$cImage = confusion(pixelArray, confusionIndex) \tag{16}$$

In diffusion, the 2^{nd} section of the cipher, the *cImage* is XORed with the *hashKeyStream* which is generated using SHA-256 (6). Diffusion is represented as follows:

$$diffusedArray = diffusion(confusedArray, hashKeyStream) \tag{17}$$

In the 3rd section, *diffusedArray* is again diffused for $\mathcal{N}$ times as follows:

$$diffusedArray' = diffusion(confusedArray, hashKeyStream)^{\mathcal{N}} \tag{18}$$

Lastly, *diffusedArray′* is again confused for $\mathcal{M}$ times as:

$$cImage' = confusion(diffusedArray', confusionIndex) \tag{19}$$

Finally, to get encrypted image

$$eImage = confusion(E^n, confusionIndex)^{\mathcal{M}} \tag{20}$$

where E is given by,

$$E = diffusion(diffusedArray, hashKeyStream) \tag{21}$$

In (17), the used *hashKeyStream* is generated from (6). The control parameters of (6) depends on *Seed* which is calculated from *patientId* and constant key value *c*. Hence, *hashKeyStream* is dependent on *patientId* and *c*. In addition, *c* is derived from hash of first image of corresponding patient's Covid-19 record. Hence, it is very hard for an hacker to obtain value of *Seed* without getting the input values *patientId* and his record.

Differential Cryptanalysis

Differential cryptanalysis [53] is a cryptanalysis technique that attempts to determine the difference between enciphered plain images. These plain images are typically varied by a single bit. By analyzing a plain image and its encrypted image, the analysis demonstrates that all confusion matrices can be successfully recovered from an encrypted image. Because the proposed algorithm is a one-round encryption process, consider r = 1 when analyzing the proposed system using differential cryptanalysis. Consider plain images as P_1 and P_2 and the acquired encrypted images E_1^1 and E_2^1. The differential image is given by:

$$\delta E^r = E_1^1 \oplus E_2^1 \tag{22}$$

Let $F_C^r()$ and $F_D()$ are linear functions representing the confusion and diffusion processes. Then, (22) is expanded to:

$$\begin{aligned}\delta E^r &= (F_C^r(P_1, eK_1) \oplus F_D(eK_1)) \oplus (F_C^r(P_2, eK_2) \oplus F_D(eK_2))\\ &= F_C^r(P_1 \oplus P_2) \oplus F_C^r(eK_1 \oplus eK_2) \oplus F_D(eK_1 \oplus eK_2)\\ &= F_C^r(\delta P) \oplus F_C^r(\delta eK) \oplus F_D(\delta eK)\end{aligned} \tag{23}$$

Equation (23) shows the relationship between the differential plain and encrypted images. The differential encrypted δE image totally depends on the confusion key stream $F_C^r(eK_1)$, $F_C^r(eK_2)$ as well as diffusion key stream $F_D(eK_1)$, $F_D(eK_2)$ for every encryption.

In some other schemes, if there is no relationship between the plain image and the key stream in substitution and permutation processes, the differential encrypted image E^r is completely irrelevant to the key streams.

$$\begin{aligned}\delta E^r &= (F_C^r(P_1, eK) \oplus F_D(eK)) \oplus (F_C^r(P_2, eK) \oplus F_D(eK))\\ &= F_C^r(P_1 \oplus P_2) \oplus F_C^r(eK \oplus eK) \oplus F_D(eK \oplus eK)\\ &= F_C^r(\delta P)\end{aligned} \tag{24}$$

From (24), the following observations are made:

(i) The encrypted image difference results from the outcome of plain image difference P and a sequence of confusion process F_C^r. For instance, if P_2 is a blank image, i.e., all pixels are ZERO then the differential encrypted image depends on P_1 and the confusion function.

(ii)If the attacker selects a special value for P_1, the attacker may be able to determine the confusion function. Once an attacker has encrypted E_2^r, confusion function and its plain image P_2 then any plain image P_1 can be determined from encrypted image E_1^r.

However, in the proposed cryptosystem, there is a correlation between different E^r. Here, confusion and diffusion keys for each encryption round are different, reducing the possibility of breaking the proposed algorithm. Furthermore, because the confusion method is related to plain image, it will complicate the cryptanalysis process.

Conclusion

In this paper, Covid-19 status checking for issuing travel tickets using privacy-preserving storage and sharing of Covid-19 records through secure authentication is proposed. The proposed system is developed using cryptographic hash function SHA-256, Chaotic map, Paillier cryptosystem, and InterPlanetary File System (IPFS). In secure authentication, the contents of the Authentication Data Table (ADT) are well protected against a brute-force attack, dictionary attack, advanced dictionary attack, lookup table attack, and rainbow table attack because of hashed UserIds and AES-256 encrypted password. The encryption key is unique for each user and is derived from hashed input UserId and Password. Novel image encryption is also developed as a part of the proposed system. To withstand known plain text/chosen plain text attacks and to increase the level of security, the control values of the keystreams required during confusion–diffusion of the proposed cryptosystem are derived from input Patient Id and Covid-19 record. From comparison analysis, it is found that the proposed image encryption scheme has achieved significant security as well as privacy.

In the future, the focus will be given to increasing the speed of the cryptosystem by applying parallel computing in confusion and diffusion. Also, nowadays, the biometric recognition system is becoming popular in all sectors, including banking, identity verification in the airport, and authentication systems. Biometric information of the user such as face template, fingerprint template, iris template can be encrypted using the proposed system to provide a higher level of security.

Declarations

Conflict of Interests All authors declare that they do not have any conflict of interest.

Informed Consent In this study, we used traditional image processing gray images from the USC SIPI database, and medical images from NIH (National Institutes of Health) are taken to analyze the performance. Apart from these databases additional two Covid-19 databases are used for performance analysis, namely 1.COVID-19 British Society of Thoracic Imaging database, 2. Eurorad COVID-19 cases. These databases are provided by The European Institute for Biomedical Imaging Research (EIBIR).

Ethical Approval In this study, we have not used any humans or animals as material for experiments. All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with 1964 Helsinki Declaration and its later amendments or comparable ethical standards.

References

1. Dobbertin H, Bosselaers A, Preneel B.RIPEMD-160: A strengthened version of RIPEMD. in *International Workshop on Fast Software Encryption* (Springer, 1996), pp. 71–82
2. Halevi S, Krawczyk H. Randomized hashing and digital signatures (2006)
3. Al-Kuwari S, Davenport JH, Bradford RJ. Cryptographic hash functions: recent design trends and security notions. IACR Cryptolo ePrint Arch. 2011;2011:565.
4. Schneier B. Cryptanalysis of MD5 and SHA: Time for a New Standard. Computerworld **19** (2004)
5. Standard AE. National Bureau of Standards. Washington,Federal information processing standard (fips) publication 197. DC: US Department of Commerce; 2001.
6. Carlton RA. Secure Integer Comparisons Using the Homomorphic Properties of Prime Power Subgroups. Electronic Thesis and Dissertation Repository (2017)
7. Damgard I, Geisler M, Kroigard M. Homomorphic encryption and secure comparison. Int J Appl Cryptogr. 2008;1(1):22.
8. Gentry C. et al., Fully homomorphic encryption using ideal lattices. in *Stoc*, vol. 9 (2009), vol. 9, pp. 169–178
9. Paillier P. Public-key cryptosystems based on composite degree residuosity classes. in *International Conference on the Theory and Applications of Cryptographic Techniques* (Springer, 1999), pp. 223–238
10. Hsu CY, Lu CS, Pei SC. Homomorphic encryption-based secure SIFT for privacy-preserving feature extraction. in *Media Watermarking, Security, and Forensics III*, vol. 7880 (International Society for Optics and Photonics, 2011), vol. 7880, p. 788005
11. Zhang L, Jung T, Liu K, Li XY, Ding X, Gu J, Liu Y. Pic: Enable large-scale privacy preserving content-based image search on cloud. IEEE Trans Parallel Distrib Syst. 2017;28(11):3258.
12. Bonneau J, Herley C, Van Oorschot PC, Stajano F. Passwords and the evolution of imperfect authentication. Commun ACM. 2015;58(7):78.
13. Gokhale MAS, Waghmare VS. The shoulder surfing resistant graphical password authentication technique. Proc Comput Sci. 2016;79:490.
14. Ma J, Yang W, Luo M, Li N. A study of probabilistic password models. in *2014 IEEE Symposium on Security and Privacy* (IEEE, 2014), pp. 689–704
15. Adams A, Sasse MA. Users are not the enemy. Commun ACM. 1999;42(12):41.
16. Li Y, Wang H, Sun K. Personal information in passwords and its security implications. IEEE Trans Inf Forensics Secur. 2017;12(10):2320.
17. Spafford EH. Opus: Preventing weak password choices. Comput Secur. 1992;11(3):273.
18. Sun HM, Chen YH, Lin YH. oPass: A user authentication protocol resistant to password stealing and password reuse attacks. IEEE Trans Inf Forensics Secur. 2011;7(2):651.
19. Shay R, Komanduri S, Durity AL, Huh PS, Mazurek ML, Segreti SM, Ur B, Bauer L, Christin N, Cranor LF. Designing password policies for strength and usability.ACM Transactions on Information and System Security (TISSEC). 2016;18(4):13.
20. Andriotis P, Tryfonas T, Oikonomou G. Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method. in *International Conference on Human Aspects of Information Security, Privacy, and Trust* (Springer, 2014), pp. 115–126
21. Zviran M, Haga WJ. Password security: an empirical study. J Manag Inf Syst. 1999;15(4):161.
22. Song R. Advanced smart card based password authentication protocol. Comput Stand Interfaces. 2010;32(5–6):321.
23. Wegman MN, Carter JL. New classes and applications of hash functions. in *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)* (IEEE, 1979), pp. 175–182
24. Preneel B, Govaerts R, Vandewalle J. Hash functions for information authentication. in *CompEuro 1992 Proceedings Computer Systems and Software Engineering* (IEEE, 1992), pp. 475–480
25. Ah Kioon MC, Wang ZS, Deb Das S. Security analysis of MD5 algorithm in password storage. in *Applied Mechanics and Materials*, vol. 347 (Trans Tech Publ, 2013), vol. 347, pp. 2706–2711

26. Oechslin P. Making a faster cryptanalytic time-memory trade-off. in *Annual International Cryptology Conference* (Springer, 2003), pp. 617–630
27. Kelsey J, Schneier B, Hall C, Wagner D. Secure applications of low-entropy keys. in *International Workshop on Information Security* (Springer, 1997), pp. 121–134
28. Luo W, Hu Y, Jiang H, Wang J. Authentication by encrypted negative password. IEEE Trans Inf Forensics Secur. 2018;14(1):114.
29. Sun J, Yao X, Wang S,. Wu Y. Blockchain-based Secure Storage and Access Scheme For Electronic Medical Records in IPFS. IEEE Access (2020)
30. Zheng Q, Li Y, Chen P, Dong X. An innovative IPFS-based storage model for blockchain. in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)* (IEEE, 2018), pp. 704–708
31. Chen Y, Li H, Li K, Zhang J. An improved P2P file system scheme based on IPFS and Blockchain. in *2017 IEEE International Conference on Big Data (Big Data)* (IEEE, 2017), pp. 2652–2657
32. Zeng L, Liu R, Zhang LY, Liu Y, Wong KW. Cryptanalyzing an image encryption algorithm based on scrambling and Veginère cipher. Multimed Tools Appl. 2016;75(10):5439.
33. Li Y, Wang C, Chen H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Opt Lasers Eng. 2017;90:238.
34. Pareek N, Patidar V, Sud K. Cryptography using multiple one-dimensional chaotic maps. Commun Nonlinear Sci Numer Simul. 2005;10(7):715.
35. Teng L, Wang X. A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive. Opt Commun. 2012;285(20):4048.
36. Xu L, Li Z, Li J, Hua W. A novel bit-level image encryption algorithm based on chaotic maps. Opt Lasers Eng. 2016;78:17.
37. Hua Z, Zhou Y. Image encryption using 2D Logistic-adjusted-Sine map. Inf Sci. 2016;339:237.
38. Cao W, Mao Y, Zhou Y. Designing a 2D infinite collapse map for image encryption. Signal Processing p. 107457 (2020)
39. Mansouri A, Wang X. A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. Information Sciences (2020)
40. Vidhya R, Brindha M. A chaos based image encryption algorithm using Rubik’s cube and prime factorization process (CIERPF). Journal of King Saud University-Computer and Information Sciences (2020)
41. Zhou M, Wang C. A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks. Signal Processing p. 107484 (2020)
42. Song W, Zheng Y, Fu C, Shan P. A Novel Batch Image Encryption Algorithm Using Parallel Computing. Inf Sci. 2020;2:2.
43. Falmari VR, Brindha M. Privacy preserving cloud based secure digital locker using Paillier based difference function and chaos based cryptosystem. J Inf Secur Appl. 2020;53:102513.
44. Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcation Chaos. 1998;8(06):1259.
45. Lian S, Sun J, Wang Z. A block cipher based on a suitable use of the chaotic standard map. Solitons Fractals. 2005;26(1):117.
46. Zl Zhu, Zhang W, Wong Kw YuH. A chaos-based symmetric image encryption scheme using a bit-level permutation. Inf Sci. 2011;181(6):1171.
47. Azimi Z, Ahadpour S. Color image encryption based on DNA encoding and pair coupled chaotic maps. Multimed Tools Appl. 2020;79(3):1727.
48. Rukhin A, Soto J, Nechvatal J, Smid M, Barker E. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Booz-allen and hamilton inc mclean va: Tech. rep; 2001.
49. Murugan B, Gounder AGN. Image encryption scheme based on block-based confusion and multiple levels of diffusion. IET Comput Vision. 2016;10(6):593.
50. Babaei A, Motameni H, Enayatifar R. A new permutation-diffusion-based image encryption technique using cellular automata and DNA sequence. Optik. 2020;203: 164000.
51. Jithin K, Sankar S. Colour image encryption algorithm combining, Arnold map, DNA sequence operation, and a Mandelbrot set. J Inf Secur Appl. 2020;50:102428.
52. Herbadji D, Belmeguenai A, Derouiche N, Liu H. Colour image encryption scheme based on enhanced quadratic chaotic map. IET Image Proc. 2019;14(1):40.
53. Chen L, Ma B, Zhao X, Wang S. Differential cryptanalysis of a novel image encryption algorithm based on chaos and Line map. Nonlinear Dyn. 2017;87(3):1797.

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