Tree parity machine guided patients' privileged based secure sharing of electronic medical record: cybersecurity for telehealth during COVID-19

Arindam Sarkar¹ · Moumita Sarkar²

Received: 15 July 2020 / Revised: 23 October 2020 / Accepted: 10 February 2021 /

Published online: 21 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In the COVID-19 pandemic, telehealth plays a significant role in the e-healthcare. E-health security risks have also risen significantly with the rise in the use of telehealth. This paper addresses one of e-health's key concerns, namely security. Secret sharing is a cryptographic method to ensure reliable and secure access to information. To eliminate the constraint that in the existing secret sharing schemes, this paper presents Tree Parity Machine (TPM) guided patients' privileged based secure sharing. This is a new secret sharing technique that generates the shares using a simple mask based operation. This work considers addressing the challenges presents in the original secret sharing scheme. This proposed technique enhances the security of the existing scheme. This research introduces a concept of privileged share in which among k number of shares one share should come from a specific recipient (patient) to whom a special privilege is given to recreate the original information. In the absence of this privileged share, the original information cannot be reconstructed. This technique also offers TPM based exchange of secret shares to prevent Man-In-The-Middle-Attack (MITM). Here, two neural networks receive common inputs and exchange their outputs. In some steps, it leads to full synchronization by setting the discrete weights according to the specific rule of learning. This synchronized weight is used as a common secret session key for transmitting the secret shares. The proposed method has been found to produce attractive results that show that the scheme achieves a great degree of protection, reliability, and efficiency and also comparable to the existing secret sharing scheme.

Keywords Telehealth · Security · COVID-19 · Artificial Neural Networks (ANN) · Secret share

---

✉ Arindam Sarkar
arindam.vb@gmail.com

Moumita Sarkar
kar.moumi@gmail.com

¹ Department of Computer Science and Electronics, Ramakrishna Mission Vidyamandira, Belur Math, Howrah, 711202, West Bengal, India

² Department of Computer Science, The University of Burdwan, Golapbag, Burdwan, 713104, West Bengal, India

1 Introduction

Presently, as the world deals with an unforeseen COVID-19 pandemic [47], the recent trend of social distancing [7] is to work from home with lockdowns and travel limitations. In December 2019, the ‘future’ of telehealth began [46]. Global COVID-19 is making the first telehealth trial for millions around the world. Telehealth bridges the gap between individuals, clinicians, and healthcare systems, allowing everyone, especially symptomatic patients’, to remain at their home [20] and interact remotely with doctors, helping to minimize virus transmission to large populations and frontline medical workers.

Telehealth must be one of the alternative systems for e-healthcare and promoting health decision-making within the framework of improving service delivery [43]. In COVID-19 there is a need for proper medical diagnosis [5] at the right time so that the patients get exact and concrete medical advice from physicians, which improves the curable probabilities. A patient can pursue experienced opinions remotely through telehealth from different expert physicians. It enables doctors and patients’, via smartphones or computers with a webcam, to connect 24/7. In improving the health systems response to COVID-19, World Health Organization (WHO) cited [44] telehealth as an important service. In crises in public health people may use immersive audio and video applications for a broader variety of programs through their clinicians. Such services provide not only video visits but also text, e-mail, and smartphone apps and can extend them to include wearable technology and chatbots [37]. These services can be utilized to assist patients’ with COVID-19 and those requiring additional clinical regular services in synchronous and asynchronous ways.

Secret sharing or secret division is a technique by which a secret is shared between a group of participants. A piece or share of the secret is obtained by every participant. A subset of a predetermined number of individuals will cooperate and disclose the secret data during the secret recovery stage. The existing secret sharing scheme present in the literature has several drawbacks. They do not support privilege based reconstruction of information. Due to these patients’ information can be misused. Also, the existing scheme can experience a MITM attack at the time of exchanging the secret shares. To eliminate the limitation present in the existing scheme, this paper presents Tree Parity Machine (TPM) guided patients’ privileged based secure sharing of information. In this proposed scheme patient has been treated as a privileged recipient. It is not possible to intentional sharing of patients’ information and clinical signals because the original information can’t be regenerated excluding the amalgamation of the patient’s secret share. The proposed secret sharing technique offers a scheme where any k (threshold) number of information shares including the share of the privileged recipient (patient) can only be able to reconstruct the original information. No confidential data of the patient be revealed without any permission of the patient. Here, data are shared secretly among n number of recipients. So, there’s no risk of losing or corrupting entire data due to medium or key corruption.

It is, necessary to search for innovative methods of secured and low costs protocol for the generation/exchange of keys. This is a big challenge to resist the attacker, although he/she may partake in the ability to follow the algorithm framework. The process of synchronization of the TPM provides an ability to address immense exchange problems. So, there is a need to develop a secured key exchange protocol with minimum parameter exchange to resist the MITM attack. In this proposed technique sender’s and receiver’s TPM accepts common input and different random weight vector and exchanges their output. It results in complete synchronization after some steps by setting weights according to the common learning rules. In this way, the sender’s and receiver’s TPM can produce a key that is difficult to infer by the attacker even though the algorithm is known to the attacker. To leap towards

data secrecy, the proposed technique plays a pivotal role in the secured online transmission of information.

Hence, this proposed scheme can be incorporated with any existing telehealth system as a highly secured online information transmission module. This would give a new dimension to treat the patients carefully without disclosing their sensitive information.

The rest of this paper is structured accordingly. The background of the work is described in Section 2. Section 3 presents related works. Section 4 deals with the proposed methodology. Section 5 discusses the proposed technique using an example. Section 6 represents the time complexity analysis of the proposed technique. Section 7 deals with results and analysis. In Section 8, the conclusion and future scope are given and references are given at the end.

2 Background

Telehealth aims to be an efficient and secure way to prevent COVID-19 from spreading during this global pandemic. At a time when social isolation is one of the most important steps in the fight against the COVID-19 pandemic, individual visits are the second, third, or last choice. The COVID-19 is a serious acute respiratory disease caused by the form of coronavirus arising in December 2019 in the city of Wuhan, China and rapidly spreading around the world [19, 48]. The COVID-19 pandemic, which not only threatens public health but also affects other facets of human life, in particular the global economy, is currently affecting many of the countries [27]. Different steps to deal with and manage COVID-19 have been considered by various countries. One of the best ways to cope with and monitor the COVID-19 pandemic is by telehealth systems [26]. Due to the high risk of transmission of the disease through personal contact, the regulation of the COVID-19 through minimizing direct contact is possible by telehealth. One of telehealth ‘significant applications is to monitor the patients’ following the hospital release [45], which can also be used for COVID-19 patients’. The quality of telehealth and e-healthcare also has to do with data security and privacy. Confidentiality and patients’ data security must be considered vital, and telehealth scholars have addressed this issue as a major challenge [40]. Data security and privacy are the key issues to be addressed before communicating patients’ data to doctors.

The US legislation providing data protection and security measures for the protection of health information is HIPAA (Health Insurance Portability and Accountability Act, 1996). As per HIPAA, human-operated ransomware attacks toward e-healthcare organizations and critical infrastructure escalated during the COVID-19 pandemic [10, 15]. Hundreds of attacks on health institutions have occurred [9]. To help track individuals who have been confirmed to have contracted COVID 19, Apple and Google develop contact tracking technology, but the Electronic Frontier Foundation (EFF) has cautioned that cybercriminals are misusing the program in its current form [11]. The WHO is one of the leading organizations battling COVID-19 and has proven to be an enticing target for hackers and hacktivists who have escalated the attack during a COVID-19 pandemic by the organization. WHO reported last month that by spoofing, hackers attempted to access its network as well as its partners. The Federal Bureau of Investigation (FBI) has reported that cybercriminals are trying to steal money from buyers of personal protective equipment (PPE) and medical equipment from the state and health sector [12, 17]. INTERPOL has informed hospitals during the current 2019 pandemic that there will be continual ransomware attacks [16]. Although some ransomware groups have publicly declared that they will stop the attacks on the front-line healthcare providers coping with COVID-19, others continue to strike. The Federal Office of Inquiry issued a notice that tried to take advantage of the outbreak of the

COVID-19 pandemic following an uptick in Business Email Compromise (BEC) attacks [13, 14]. Fraudsters have already launched new and COVID-19-themed activities almost exclusively in 2019, as per a new study released by them. COVID-19 is associated with 80% of all threats reported by the company. In recent times, over half a million e-mails, 300,000 malicious URLs, and more than 200,000 malicious e-mail attachments have been analyzed. During the COVID-19 crisis, teleconference systems such as Zoom have proven popular among companies and clients when working from home. In recent days, however, some zoom protection concerns have been discovered that have raised doubts about the suitability of a device for medical use [18].

3 Related works

Gupta et al. [8] proposed a safe mechanism by means of the neural cryptography to share the hidden shares of an image between parties involved. A simple multi-secret image share scheme has been proposed by Chattopadhyay et al. [6] that uses a stable hash function and Boolean operations. Nag et al. [32] have designing a framework for multi-secret image sharing based on XOR when t is not limited only to 2 or n . In order to generate r (where r is the number of qualify subgroups) they have used n alters of the same size as confidential image size to distribute images publicly using XOR operations. Alkhodaidi et al. [3] also proposed a technique which adds the technique of image steganography to the counting-based secret-sharing method to obtain more relevant security. The article suggests that LSB image steganography be redistributed to integrate share keys in the colored cover image. Blesswin et al. [21] introduced the Enhanced Semantic Visual Secret Sharing (ESVSS) Scheme, which uses two color cover images to relay a gray-scale secret image to the recipient. The hidden picture is rebuilt at the end of the receiver by digitally piling the shares together. Shivani et al. [39] proposed a scheme for exchange the medical images with XOR based Multi Secret Sharing scheme for storage and forward-looking telemedicine. This scheme is n out of n multi-secret sharing systems that are capable of concurrently transmitting n secret images. Only after some calculations for all n shares and a master component could all secrets be exposed. Nabiyev et al. [31] have proposed a secret share based medical image transmission. There are some limitations to the current secret sharing scheme present in the literature. They do not endorse information reconstruction based on privileges. It is possible to misuse the data of these patients because of this. Also, at the time of exchanging the secret shares, the current system will experience an MITM attack. Nayak et al. [34] have proposed a technique for storing patients' data after encrypting it using Advanced Encryption Standard and then superimposed within a retinal image. Also, Nayak et al. [33, 34] have done a relative analysis regarding the concurrent storage of patients' diagnosed images in both frequency and spatial domains. Different types of signals displaying the rhythm of the heart, signals generated by the brain, etc are captured electronically, compressed, and encrypted. Subsequently, the compressed ciphertext is embedded in patients' medical images at the LSB level. Acharya et al. [1, 41, 42] proposed some techniques for transmitting data inside the image keeping the security aspect intact. Researchers have discussed also the chaos-based encryption technique. Raeiatibanadkooki et al. [38] have suggested a chaotic Huffman code for compression and a wavelet transformation for encryption of ECG signal. The main objective is to squeeze and encrypt the ECG without the loss. Lin [23] has described a chaotic encoding method of EEG signal along with a view towards logistic maps and experimental forms of decomposition. Lin et al. [24] developed a new encryption scheme. However, the security analysis was missed out. Ahmad et al. [2] have

suggested a security investigation to the robustness of the encryption. But a comprehensive analysis was lacking in that method. Telecare diagnosis system has emerged as a treatment tool in the e-health community. Researchers are putting their best efforts to provide a more flexible system for society. Mulyadi et al. [29] have proposed how to improve the accuracy of a 12-lead ECG using waveform segmentation of ECG graphs. Liaquat et al. [22] have designed a framework to establish relationships between various cardiac patients' attributes using unsupervised learning techniques. In this technique, a K-means is applied to derive the hidden relationships among various patients' attributes. Capua et al. [4] have suggested a technique to monitor and assess the ECG signals in real-time. ECG signals are sensed through the sensor and processed by a personal digital assistant (PDA). The PDA may detect and diagnose the ECG signals, and call the emergency personnel in case of abnormalities detected. Patients' can also visualize the ECG clinical signals inside the PDA interface. Murillo-Escobar et al. [30] have proposed chaotic function-based enciphering on the ECG / EEG signals. They have proposed a system on the logistic map based encryption. Although they have tried to impose encryption hardness using diffusion of a chaotic system, still the issue of Man-In-The-Middle (MITM) attacks has not been addressed properly. Here, the clinical ECG / BP / EEG signal of the patients' that could be captured by the intruders for fake or inconsistent purposes. Secured online transmission of the confidential patients' data/signals is major issue. Implementation of their proposed technique has not been appropriately explained on the real-life application systems. Moreover, live sensing of the patients' signals and data compression behavior was missing in their work.

4 Proposed methodology

In this proposed technique, a robust session key is generated through the synchronization of two TPMs. In different sessions, different session keys are being used for secured transmission. A mask is generated to generating the encrypted share of the information. Then each share is ciphered through the neural session key. Finally, the neural encryption key is encrypted using the privileged recipient's RSA public key.

Proposed Algorithm

Requirement: *Patient's EMR*

Input: *Total no. of shares (n), no. of threshold (k)*

Output: *n number of secret shares of EMR*

{/* Session key generation*/}

EncryptionKey[ ] = Call NeuralSessionKey () ;

{/*Secret Mask generation*/}

Mask[n_{C_{k-1}}][n] = Call Mask_Generation(

Mask[n_{C_{k-1}}][n] , n , k);

{/*Secret Shares Generation*/} *Share[n][n_{C_{k-1}}] = Call SecretShare_Generation(*
Mask[n][n_{C_{k-1}}] , EMR[p][q]);

{/*Encryption of Partially open shares*/}

EncryptedShare[i][j] = Call ShareEncryption

(Share[n][ ], EncryptionKey[ ]);

{/*RSA Encryption of Encrypted Key with Privilege Recipient's Public key*/}

PrivilegeEncryptedKey[ ] = Call RSA (

EncryptionKey[ ],

PrivilegeReceipient'spublicKey);

Report n transmittable shares

4.1 Mask matrix formation

This algorithm for generating masks requires two inputs. One is the number of shares (n) and the other parameter is the value of the threshold. The $Mask[n_{C_{k-1}}][n]$ is determined depending on the value of (n) and (k). In the mask generator, the number of recipients, i.e. (n) and the minimum number of shares, i.e. threshold value (k), are placed. The length of each mask is $n_{C_{n-k}}$. The proposed work is focused on masking which uses *AND*-ing for share generation and *OR*-ing predefined minimum number of shares to reconstruct the original. The beauty of the mask generation algorithm is that it follows the property of lossless join decomposition with the effect of data integrity.

A secret information must be transmitted as a sequence of bits 0s and 1s. This proposed work depends on the masking of the predefined n number of shares and then on successful *OR*-ing with the predefined k shares (including privileged shares), where $2 \leq k \leq n$ is needed to regenerate the originally transmitted information. Shares are created by a mask and each share has some missing bits that can be recovered by no less than that exact ($k-1$) share. The mask size is $n_{C_{k-1}}$ in this technique since certain combinations of bits should be $n_{C_{k-1}}$ in any bit location.

This mask is *AND*-ed with the secret information to form the partially opened shares. Thus, 1 in the mask retains the data at that particular position whereas 0 in the mask eliminates the secret bit. This procedure compressed the share by eliminating some secret bits which can be recovered at the time of secret reconstruction by *OR*-ing any k number of shares with the help of the mask. So, n number of masks can form n number of compressed shares. Mask having different 1 and 0 combinations generates different shares. The length of each mask can be measured as $n_{C_{k-1}}$ i.e. $\frac{n!}{(k-1)! \times (n-k-1)!}$.

Total number of missing and secret data bits are $n - 1_{C_{k-2}}$ and $n - 1_{C_{n-k}}$ respectively. So, the size of each mask is $n_{C_{k-1}}$ and the total number of 0's are $n - 1_{C_{k-2}}$ and the number of 1's are $n - 1_{C_{n-k}}$. In the compressed share total number of bits is $n - 1_{C_{n-k}}$ i.e. total number of 1's present in the share because 0's are discarded.

The complete mask generation algorithm is given as follows.

Algorithm *int Mask_Generation*
 ($Mask[n_{C_{k-1}}][n]$, n , k)

Input: $Mask[n_{C_{k-1}}][n]$, n , k

Output: Mask for each share.

Method: Generate a mask matrix of dimension $n_{C_{k-1}} \times n$ having $(n - k + 1)$ number of 1's and $(k - 1)$ number of 0's in each row. Arranged the row in a predefined order.

Transpose this mask matrix so that each row of this matrix can represent an individual mask for each share. Now, the new matrix dimension will be $n \times n_{C_{k-1}}$ after transpose.

Each row size will be $n_{C_{k-1}}$

1. $P = n_{C_k}$, $Ceiling\_Bar$, S , $NO1 = 3$, : Integer
 $k1 = 0$, $k2 = 0$, Mid , i , j
2. $BIN[n]$, $Index[P]$, $Index2[P]$: Integer array
3. $Mid = Floor((0 + (P - 1) / 2))$;
4. For $i = Ceiling\_Bar$ to 1 do
5. $BIN[n] \leftarrow Decimal2Binary\_Convert(i)$;
6. Check if $One\_Count(BIN[n]) = NO1$
7. $Index[k1 + +] \leftarrow i$;
8. end if

```

9. end for
10.  $Index[P] \leftarrow ASC_{SORT}(Index[0, Mid]) +$ 
    $ASC_{SORT}(Index[Mid + 1, P - 1]);$ 
// '+' is the concatenation o
11.  $i = 0;$ 
12.  $j = Mid + 1;$ 
13. While ( $i \leq Mid$  &&  $j \leq P - 1$ ) do
14.    $Index2[k2 + +] \leftarrow Index[i];$ 
15.    $Index2[k2 + +] \leftarrow Index[j];$ 
16.   increment  $i;$ 
17.   increment  $j;$ 
18. end while
19. For  $S = 0$  to  $(P - 1)$  do
    $Mask[S][ ] \leftarrow Decimal2Binary\_Convert$ 
20.    $(Index[Index2[$ 
      $S]]);$ 
21. end for
22.  $Mask[n][P] \leftarrow [Mask[P][n]]^T$ 
23. Return  $Mask[n][P]$ 

```

4.2 Masking of patients' information

Here a secret record is shared among some n recipients and some of them (k) can reconstruct the record if and only if among k number of shares one of them should be the privileged recipient. But less k number of recipients or without privileged recipient, record can't be reconstructed. At first, secret record is divided into some blocks and then each block is divided and shared among n number of recipients. In this way each block record is constructed by k number of recipients and at last, block records are combined to construct secret record. The algorithm of generating secret shares is represented as follows.

SecretShare_Generation(
 $Mask[n][n_{C_{k-1}}], EMR[p][q]$
Input: $Mask[n][n_{C_{k-1}}], EMR[p][q]$

Output: n number of Secret shares

Method: Generate n number of secret shares by AND-ing secret $Mask[n][n_{C_{k-1}}]$ with the input image.

{/* AND-ing of information with secret mask to generate Secret shares*/}

```

1. Set  $p = 1, q = 1$ 
2. for  $i = 1$  to  $n$  do
3.   for  $j = 1$  to  $n_{C_{k-1}}$  do
4.      $Share[i][j] = Mask[i][j] \&$ 
        $EMR[p][q]$ 
5.     increment  $j, p, q$ 
6.   end for
7.   increment  $i$ 
8. end for

```

4.3 TPM based encryption of partially open shares

A special Artificial Neural Network architecture called TPM (Tree Parity Machine) is used at the patient and caregiver end. TPM based synchronization is a modern source of public key exchange schemes that have lower memory and machine time complexities. TPM can build a mutual secret key between two parties. In keeping with all of these, the attention is on sharing secrets on a public platform with less computational power. With TPM, both parties have the same input vector, produce an output bit, and are learned based on the output bit. The weight vectors of both networks will become identical after full synchronization. These same weights serve as a key common to both parties.

TPM is composed of M no. of input neurons for each H no. of hidden neurons. TPM has an only neuron in its output. TPM works with binary input, $xinput_{u,v} \in \{-1, +1\}$. The mapping between input and output is described by the discrete weight value between $-Lrange$ and $+Lrange$, $weight_{u,v} \in \{-Lrange, -Lrange + 1, \dots, +Lrange\}$. In TPM u -th hidden unit is described by the index $u = 1, \dots, H$ and that of $v = 1, \dots, M$ denotes input neuron corresponding to the u -th hidden neuron of the TPM. Each hidden unit calculates its output by performing the weighted sum over the present state of inputs on that particular hidden unit which is given by (1).

$$\begin{aligned} hidden_u &= \frac{1}{\sqrt{M}} WEIGHT_u.XINPUT_u \\ &= \frac{1}{\sqrt{M}} \sum_{v=1}^M weight_{u,v} xinput_{u,v} \end{aligned} \quad (1)$$

$signum(hidden_u)$ define the output $\sigma output_u$ of the u -th hidden unit. (in (2)),

$$\sigma output_u = signum(hidden_u) \quad (2)$$

If $hidden_u = 0$ then $\sigma output_u$ is set to -1 to make the output in binary form. If $hidden_u > 0$ then $\sigma output_u$ is mapped to $+1$, which represents that the hidden unit is active. If $\sigma output_u = -1$ then it denotes that the hidden unit is inactive (in (3)).

$$signum(hidden_u) = \begin{cases} -1 & \text{if } hidden_u \leq 0 \\ +1 & \text{if } hidden_u > 0 \end{cases} \quad (3)$$

The product of the hidden neurons denotes the ultimate result of TPM. This is represented by $\tau output$ is (in (4)),

$$\tau output = \prod_{u=1}^H \sigma output_u \quad (4)$$

The value of $\tau output$ is mapped in the following way (in (5)),

$$\tau output = \begin{cases} -1 & \text{if } \sigma output_u = -1, \text{ is odd} \\ +1 & \text{if } \sigma output_u = -1, \text{ is even} \end{cases} \quad (5)$$

$\tau output = \sigma output_1$, if only one hidden unit ($H = 1$) is there. $\tau output$ value can be the same for 2^{H-1} different ($\sigma output_1, \sigma output_2, \dots, \sigma output_H$) representations.

If the output of two parties disagree, $\tau output^{TPM-A} \neq \tau output^{TPM-B}$, then no update is allowed on the weights. Otherwise, follow the following rules:

TPM be trained from each other using Hebbian learning rule (in (6)).

$$\begin{aligned} weight_{u,v}^+ = & \\ & fn(weight_{u,v} + xinput_{u,v} \tau output \Theta(\sigma output_u \tau output) \\ & \Theta(\tau output^{TPM-A} \tau output^{TPM-B})) \end{aligned} \quad (6)$$

In the Anti-Hebbian learning rule, both TPM is learned with the reverse of their output (in (7)).

$$\begin{aligned} weight_{u,v}^+ = & \\ & fn(weight_{u,v} - xinput_{u,v} \tau output \Theta(\sigma output_u \tau output) \\ & \Theta(\tau output^{TPM-A} \tau output^{TPM-B})) \end{aligned} \quad (7)$$

If the set value of the output is not imperative for tuning given that it is similar for all participating TPM then random-walk learning rule is used (in (8)).

$$\begin{aligned} weight_{u,v}^+ = & \\ & fn(weight_{u,v} + xinput_{u,v} \Theta(\sigma output_u \tau output) \\ & \Theta(\tau output^{TPM-A} \tau output^{TPM-B})) \end{aligned} \quad (8)$$

If $X = Y$ then $\Theta(X, Y) = 1$ Otherwise, If $X \neq Y$ then $\Theta(X, Y) = 0$ Only weights are updated which are in hidden units with $\sigma_{output_u} = \tau_{output}$. $fn(weight)$ is used for each learning rule (in (9)).

$$fn(weight) = \begin{cases} \text{signum}(weight)Lrange & \text{for } |weight| > Lrange \\ weight & \text{otherwise} \end{cases} \quad (9)$$

This mutual synchronization algorithm helps to generate an encryption key over a public channel. This generated encryption key is used to encrypt the partially open shares. Following algorithm represents the steps of encryption of partially open shares with the synchronized neural session key.

$$ShareEncryption(Share[n][n_{C_{k-1}}], \\ EncryptionKey[\ ])$$

Input: $Share[n][n_{C_{k-1}}]$, $EncryptionKey[ ]$

Output: n number of encrypted shares.

Method: Perform XOR operation between shares and neural synchronize key.
 {/Encryption of Partially open shares*/}

```

1. for  $i = 1$  to  $n$  do

```

2. *for* $j = 1$ *to* $n_{C_{k-1}}$ *do*

3. $EncryptedShare[i][j] = Share[i][j]$
 $XOR EncryptedKey[j]$

4. *increment j*5. *end for*6. *increment i*
$$7. \text{ EncryptedKey}[i] = \text{EncryptedKey}[i] \gg j$$
8. *end for*

4.4 Grant privilege to the desire recipient

In this proposed technique a concept of the privileged recipient has been introduced for preserving the information. The original information is reconstructed after *OR*-ing exactly k or more number of shares (including privileged recipient). Original information can't be constructed without the participation of privileged recipient. To implement this logic, $EncryptionKey[]$ is encrypted using privileged recipient's public key using RSA encryption [28]. So, privileged recipient only be able to decode the encryption keys of secret shares to get the original information.

$$RSA(EncryptionKey[], PrivilegeReceipient'spublicKey)$$

5 Example

In this proposed scheme, a mask matrix is generated by calling the following mask generation function.

$$Mask[n_{C_{k-1}}][n] = Call\ Mask\_Generation(Mask[n_{C_{k-1}}][n], n, k);$$

The following is a potential set of masks for 5 shares with a threshold of 3 shares:

S_i Mask

Share 1: 1010101011
 Share 2: 1011110100
 Share 3: 1100011110
 Share 4: 0111001101
 Share 5: 0101110011

One can easily verify that *OR*-ing can generate all 1's with three or more shares, but some positions still have 0's with less than three shares, i.e. they remain missing. Now, each secret shares are generated by calling the following secret share generation function.

$$Share[n][n_{C_{k-1}}] = Call\ SecretShare\_Generation(Mask[n][n_{C_{k-1}}], EMR[p][q]);$$

Take into account that the secret message (M) is COMPUTER20 and the size of M is 10 bytes. The following shares are now created by using a logical *AND*-ing with individual masks.

S_i Mask Shared Message

Share 1: 1010101011 C0M0U0E020
 Share 2: 1011110100 C0M0U0R00
 Share 3: 1100011110 C0000TER20
 Share 4: 0111001101 00M00ER00
 Share 5: 0101110011 000PUT0020

It can be easily noted from the above shares that each share includes partial secret data. That is, a secret byte corresponding to one is retained as it is in the mask and the secret byte

corresponding to zero is retained as zero in the mask. So every share has some bytes missing and can recover these missing bytes from a collection of specific k shares. Now all zero bytes corresponding to zero bit are discarded in the mask, which introduced a special technique of compression. Therefore, the above-shared message becomes a compressed message.

S_i Compressed Message

Share 1: CMUE20

Share 2: CMPUTR

Share 3: COTER2

Share 4: OMPER0

Share 5: OPUT20

All shares now have 30 bytes of total size (< 50). All confidential information is partly open to participants here. To solve this problem, each share is encrypted by a key. The Artificial Neural Network guided session key enables each share to be encoded. The following function performs the encryption of partially open shares.

$$\text{EncryptedShare}[i][j] = \text{Call ShareEncryption} \\ (\text{Share}[n][ ], \text{EncryptionKey}[ ]);$$

Now, a header mask can be generated using the following function. This procedure is the same as the mask generation technique for shares.

$$\text{HeaderMask}[i][j] = \text{Call} \\ \text{HeaderMask\_Generation} \\ (\text{EncryptedShare}[n][ ]);$$

Any Public Key Encryption algorithm can be used to encode the encrypted Key with Privilege Recipient's Public key with the help of the following function.

$$\text{PrivilegeEncryptedKey}[ ] = \text{Call} \\ \text{PublicKeyEncryption} (\\ \text{EncryptionKey}[ ], \\ \text{PrivilegeReceipient'spublicKey});$$

A header is generated by concatenating some essential parameters like the n , k , privileged encrypted key, and the size of the encrypted shares.

$$\text{Header}[ ] = \{n, k, \text{PrivilegeEncryptedKey}[ ], \\ \text{EncryptedShareSize}[n]\};$$

The header share generation procedure is the same as the secret share generation procedure. The following function helps to generate the header share matrix.

$$\text{HeaderShare}[n][k] = \text{Call} \\ \text{HeaderShare\_Generation} \\ (\text{HeaderMask}[n][k], \text{Header}[ ]);$$

Final shares are generated after the concatenation of corresponding header shares with a compressed secret share using the following function.

```

FinalShare[i][j] = Call
FinalShare_Generation
(EncryptedShare[n][ ], EncryptedShareSize[n],
HeaderShare[n][k]);

```

Any public-key encryption can be used to generate the transmittable final share by encoding the encrypted shares with the corresponding recipient's public-key.

```

TransmittableFinalShare[i] =
Call PublicKeyEncryption
(FinalShare[n], Receipient'spublicKey[n]);

```

6 Time complexity analysis

The time complexity factor will increase with the increase in the number of shares, and vice-versa. So, here the time complexity is directly proportional to the number of shares generation. The ability of the existence of cohesive property in every share is that they are treated as an independent entity while the process of encryption is done. Only the partial ciphered information is present in the shares to make it hidden from unauthorized access like intruders, hackers, etc. The good part is that it increases the total system maintainability factor since logical changes are done inside any particular domain affect the fewer number of other modules. Another striking feature is the coupling feature. It means the degree of interdependency between the shares. During the reassembly of the threshold number of shares, the coupling parameter coexists. The coupling parameter can be computed as $O(k)$, where k means the threshold value of shares. If $(k - 1)$ of shares or less than $(k - 1)$ number of shares are combined during the reconstruction phase, then this would not be feasible. In other words, a minimum k number of shares is required to regenerate the original data. In Artificial Neural Network synchronization and session key generation technique initialization of weight vector takes (number of input neurons $\times$ the number of hidden neurons) amount of computations. For example, if number of input neurons (n) = 5 and number of hidden neurons (k) = 6 then total numbers of synaptic links (weights) are $(5 \times 6) = 30$. Computation of the hidden neuron outputs takes k computations. The generation of the number of input vector for each k number of hidden neurons takes $(n \times k)$ amount of computations. Computation of the final output value takes a unit amount of computation because it needs only a single operation to compute the value. In the best case, the sender's and receiver's arbitrarily chosen weight vectors are identical. So, networks are synchronized at the initial stage do not need to update the weight using learning rule. So, in the best case, the computation complexity can be expressed is in form of $O(\text{initialization of input vector} + \text{initialization of weight vector} + \text{computation of the hidden neuron outputs})$. If the sender's and receiver's arbitrarily chosen weights vector are not identical then in each iteration the weights vector of the hidden unit which has a value equivalent to the value of the output neuron is updated according to the learning rule. This scenario leads to average and worst-case situation where I number of iteration to be performed to generate the identical weight vectors at both ends. So, the total computation for the average and worst-case can be expressed as $O(\text{Time complexity in first iteration} + (\text{number of iteration} \times \text{number}$

of weight updation)). The $Mask[n_{C_{k-1}}][n]$ matrix is generated to generating the share of the information. This operation takes $O(n_{C_{k-1}} \times n)$ amount of time.

7 Results and analysis

For result and simulation purpose, an Intel Core i7 10th Generation, 2.6 GHz processor, 16 GB RAM is used. Python is used for simulation purposes. Comprehensive and needful security views have been focused to affect acquaintance security and robustness issues. The precision of decimal has been used in arithmetic operations according to the authenticated IEEE Standard 754. An encryption algorithm should have outstanding cryptogram pseudorandom properties. The algorithm must also avoid all known attacks on the cryptanalyst. The following subsections present some security analysis, performance analysis, and comparisons analysis of the proposed technique.

7.1 Secret key space analysis

Consider n number of cascading encryption/decryption technique is used to encrypt/decrypt the plaintext with the help of neural synchronized session key. Then a session key of length [(number of cascaded encryption technique in bits) + (three bits combinations of encryption/ decryption technique index) + (length of n number of encryption/decryption keys in bits) + (length of n number of session keys in bits)] i.e. $[8 + (3 \times n) + (128 \times n) + (128 \times n)]$ bits to $[8 + (3 \times n) + \frac{(256 \times n) + (256 \times n)}{8}]$ number of bits. So, $\frac{[8 + (3 \times n) + (128 \times n) + (128 \times n)]}{8}$ to $\frac{[8 + (3 \times n) + (256 \times n) + (256 \times n)]}{8}$ numbers of characters.

$$= \left(1 + \frac{(3 \times n)}{8} + 16n + 16n\right) = 32n \quad \text{to} \quad \left(1 + \frac{(3 \times n)}{8} + 32n + 32n\right) = 64n \quad \text{numbers of characters.}$$

Consider any single encryption using the encryptionkey of size 512 bit which is hypothetically approved and needed to be analyzed in the context of the time taken to crack a ciphertext with the help of the fastest supercomputers available at present. In this technique to crack a ciphertext, the number of permutation combinations on the encryption key is $2^{512} = 1.340780 \times 10^{154}$ trials for a size of 512 bits only. IBM Summit at Oak Ridge, U.S. invented the fastest supercomputer in the world with 148.6 PFLOPS i.e. means 148.6×10^{15} floating-point computing/second. Certainly, it can be considered that each trial may require 1,000 FLOPS to undergo its operations. Hence, the total test needed per second is 148.6×10^{12} . Total no. of sec. a year have $= 365 \times 24 \times 60 \times 60 = 31,536,000$ sec. Total number of years for Brute Force attack: $(1.340780 \times 10^{154}) / (148.6 \times 10^{12} \times 31,536,000) = 2.86109 \times 10^{132}$ years.

7.2 Information entropy analysis

Entropy is a measure of a random sequence or uncertainty. High entropy values mean strong encryption, while low values mean poor encryption. An index of the data content is the entropy of a clinical signal. It is measured in terms of bits per character in a signal. If a character has a very chance of occurrence, then its data content is very less. Any document will carry in between 0 and 255 characters. The entropy value of such a document will be between 0 and 8 bits per character. 8 bits per character denotes equally spread data values. 1500 encrypted BP signals of 60 sec are obtained using 1500 different encryption keys.

The average entropy value of these 1500 encrypted signals is 7.98. The encrypted signal unpredictability is 99.75%.

7.3 Analysis of histogram

Both the initial and encrypted clinical signals are accessed by the program and the histogram analysis has been carried out. If this method is successful, the random pseudo values should be displayed with histograms in a uniform distribution. The encryption algorithm has outstanding statistical characteristics if the histogram is uniform. Figures 1 and 2 shows the histogram of plain and encrypted information respectively. Since the encrypted information histograms are uniform, the approach proposed is robust against histogram-based statistical attacks.

7.4 Correlation analysis

The correlation coefficient is used for this analysis to decide whether the plain signal is not correlated with the encrypted signal i.e. how much its amplitude varies. The correlation coefficient range is $(-1, 1)$, where 0 does not imply a correlation. The average correlation between plain signal and EEG signal is 0.0005, BP signal is 0.0007 and ECG signal is 0.0003 in the proposed technique. The findings suggest that the proposed encryption algorithm generates a cryptogram that is strongly unrelated to the corresponding plain signal.

7.5 Floating frequency analysis

In Figs. 3 and 4, respectively, the floating frequency of plain and encrypted signals is shown. uniform distribution of frequencies demonstrate the ruggedness of the encryption.

7.6 Autocorrelation analysis

For both a plain signal and an encrypted signal, the autocorrelation is measured at a bit level. Figures 5 and 6 shows the autocorrelation of plain and encrypted signals respectively.

Fig. 1 Histogram of frequency distribution spectrum of input source stream characters

Fig. 2 Histogram of the frequency distribution continuum of encoded stream characters using the proposed technique

Fig. 3 The original signal's floating frequency

Fig. 4 Using the proposed method, the floating frequency of the encrypted signal

Fig. 5 Autocorrelation of the original signal

The findings show that a plain signal display repeated patterns with high positive autocorrelation values while the autocorrelation is near zero. As there is no regularity or repeating pattern on the encrypted signal, the proposed system of encryption is robust to generate good cryptograms.

7.7 Quality metrics analysis

This section performed a quality review of the encoded ECG signal. It contains the Mean Squared Error (MSE), the PSNR, and the Structural Similarity Index (SSIM). The variance between both the original signal X and the encoded signal E of size L can be determined using given (10).

$$MSE = \frac{\sum_{i=1}^L (X_i - E_i)^2}{L} \quad (10)$$

The larger MSE value is often preferable by the encoding algorithm. PSNR reflects the ratio of the average pixel value to the compressed image as seen in the next (11).

$$PSNR = 10 \log_{10} \left(\frac{MAXPIX^2}{MSE} \right) \quad (11)$$

Fig. 6 Autocorrelation of the encrypted signal using proposed technique

The maximum value of pixels in the original signal is MAXPIX. A method for the evaluation of the quality of the image is the Structural Similarity Index (SSIM). It includes the cross-correlation, mean, and standard deviation of the image with the compressed image, as mentioned below using (12).

$$SSIM(X, Y) = \frac{(2\mu_x \mu_y + K_1) * (2\sigma_{xy} + K_2)}{(\mu_x^2 + \mu_y^2 + K_1) * (\sigma_x^2 + \sigma_y^2 + K_2)} \quad (12)$$

Here, μ_x and σ_x represents the mean and variance of the original image respectively. μ_y and σ_y represents the mean and variance of the compressed image respectively. σ_{xy} is the cross-correlation between the original image and the compressed image. SSIM is 1 where both images are similar, while SSIM near zero implies that they differ structurally. MSE, PSNR, and SSIM these three tests use plain signals and produce 1500 encoded signals, using 1500 different encryption keys. To pursue these tests, the original signal and cryptogram are mapped from (0,1) to [0,255], as quality indicators for images are MSE, PSNR, or SSIM. Average MSE, PSNR, and SSIM are indicated in Table 1 between the plain and encrypted ECG, EEG, and BP signals. The results show high MSE, small PSNR values, and low SSIM values. The proposed encryption system, therefore, generates effective and consistent cryptographic pseudorandoms.

To test any distortion of the initial clinical signal and the clinical signal obtained, the Percentage Root Mean Square Difference (PRD) is used. Using the following (13), PRD is computed.

$$PRD = \sqrt{\frac{\sum_{i=1}^L (X_i - \widehat{X}_i)^2}{\sum_{i=1}^L (X_i)^2}} \times 100 \quad (13)$$

Following Table 2 represents the PRD in received clinical signal with noise percentage.

7.8 Secret key sensitivity

If ECG signals or other clinical signals being encrypted by two very similar keys, then also they will generate two completely different cryptograms. The encryption system should be immunized to several attacks. Two similar keys may be the difference in fewer bits, results in two entirely different. Even though, a flip on a single bit between two encryption keys will generate two cryptograms in a completely different scenario. 1500 different encryption keys with a minimum bit difference are used to generate 1500 different cryptograms. The average correlation for BP signal is 0.0069, ECG is 0.0077 and EEG is 0.0040. Thus, intruders are not able to derive any conclusion on the exact key pattern to the signals.

7.9 Plain signal sensitivity

Like the secret key sensitivity, the proposed technique should be resistant to plain signal sensitivity too. Plain signal sensitivity refers to encrypting two almost similar clinical

Table 1 Quality metrics estimation

Clinical signals	MSE	PSNR (dB)	SSIM
ECG [36]	10925.75	7.943	0.0489
EEG [36]	10286.19	8.318	−0.0314
BP [36]	11517.73	8.187	0.0759

Table 2 PRD in received clinical signal with noise %

Signal	PRD (0% Noise)	PRD (5% Noise)	PRD (10% Noise)	PRD (15% Noise)	PRD (100% Noise)
BP [36]	2.89×10^{-16}	89.75	185.05	319.72	847.16
ECG [36]	3.10×10^{-14}	19913.42	24573.95	48567.84	112186.35
EEG [36]	1.48×10^{-15}	385.31	632.08	910.75	3196.89

signals with the same key, and resultant cryptograms are far away from each other. Two healthy persons belonging from the same demographic zone and with nearly the same age will generate their BP signals of similar patterns. There are two identical BP plain signals are considered, with the value changed from 115.4 to 115.3 for just 5 seconds. Simple shifts in plain signals make the proposed scheme especially sensitive. Moreover, the histogram of two cryptograms encrypted through the same key is immensely different in its properties.

7.10 Pseudorandom analysis under NIST 800-22

For N samples of bit sequences obtained from the optimization algorithm, one p-Value and a threshold value are evaluated. The value of the threshold is defined using (14).

$$Thresholdvalue = (1 - \alpha) - 3\sqrt{\left(\frac{\alpha \times (1 - \alpha)}{N}\right)} \quad (14)$$

Here α is the value of the significance level. The value of the significance level(α) is 0.01 for all the statistical tests. The size of sample bits N is larger than the inverse of α . In the case of serial test, cumulative sums test, and Random excursions test, generating p-Values, *Thresholdvalue* should be calculated by considering $(M \times n)$ instead of M - bits. True randomness was ensured in the proposed transmission technique bypassing the fifteen tests contained in that suite. These tests are very useful for such a proposed technique with high robustness. A probability value (p-Value) determines the acceptance or rejection of the weight vector generated by the whale optimization algorithm. Table 3 contains the results of NIST Statistical tests [35] on the generated random input vector.

7.11 Chosen plain signal attack

The chosen plain text attack was among the most effective attacks, as much chaos image encryption algorithms were broken in recent years. Cryptographic algorithms are well-known to the public. A chosen plain text assault can thus be applied to find the secret key that other cryptograms may have used. The only hidden secret is the key, through which encryption had been achieved. Intruders present silently on the network, will test several sets of keys to them. In telehealth, attackers try to the ECG signal components based on some heuristic keys. The length of the key size is proportionate to the complexity to resist against the chosen plain signal attacks. For such a chosen plain signal attack, the proposed encryption algorithm is robust. Even a one bit flip in the encryption key will immensely modify the generated encoded signal. The proposed encryption algorithm is extremely reliable for a chosen plain text attack because of the high reliance on the secret shares used in the encryption process. Further, each cryptogram relies heavily on its respective plain signal as regards its secret shares. This proves the resistance of the proposed technique against the chosen plain signal attack in the system.

Table 3 NIST statistical test

NIST test	p_Value	Status
Frequency	0.568742	Success
Frequency within a Block	0.587315	Success
Runs	0.519634	Success
Longest Run of Ones in a Block	0.089347	Success
Binary Matrix Rank	0.681438	Success
Discrete Fourier Transform	0.513691	Success
Non-overlapping Template Matching	0.460869	Success
Overlapping (Periodic) Template Matching	0.222473	Success
Maurer’s “Universal Statistical”	0.789345	Success
Linear Complexity	0.669803	Success
Serial	0.545770	Success
Approximate Entropy	0.292437	Success
Cummulative Sums	0.698130	Success
Random Excursions	0.354801	Success
Random Excursions Variants	0.237895	Success

7.12 Occlusion attack

During the transmission of clinical signals under telehealth, it obvious that it shall lose data to a certain level. The tolerance level of such an ECG signal has been estimated in the proposed technique against occlusion attack. Signals lost with higher than 5% is not suited for considerations. ECG signal has 1000 signal elements between zero and one. Randomly, 20, 50, and 150 elements were selected and defined as zero. The MSE of recovered ECG signals is 0 for 0% noise, 118.75 for 2% noise, 271.91 for 5% noise, and 1024.48 for 15% noise. If there is more than 5% occlusion, it is not possible to use the signal.

7.13 Encryption/ decryption time

All test programs for the algorithms have been designed to demonstrate the overall encryption and decryption time. The period taken is the difference between the beginning and end ticks of the processor clock. In milliseconds (ms), times are measured. The lower the amount of time, the better for a typical end-user. Because of the time needed for the CPU clock

Table 4 Comparisons of encryption / decryption time of the proposed technique with existing secret sharing scheme, and benchmark TDES encryption techniques

Source file size (in bytes)	Proposed technique (in ms)		Secret sharing [8] (in ms)		TDES [25] (in ms)	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
1,925,185	123	129	197	219	393	501
2,498,560	173	178	242	297	532	518
3,790,336	221	227	298	332	897	923
4,883,456	286	292	405	467	964	1051
5,456,704	319	326	487	517	1172	1168

Table 5 Comparison of the technique proposed with recent techniques noted in the literature

Sl. No.	Comparative Parameters	Proposed technique	Ref [30] 2017	Ref [6]	Ref [32]	Ref [3]	Ref [21]
1	ECG Clinical Signal	Yes	Yes	Yes	No	No	No
2	EEG Clinical Signal	Yes	Yes	No	No	No	No
3	BP Clinical Signal	No	Yes	No	No	No	No
4	UCD Clinical Signal	Yes	No	No	No	No	No
5	Signal Database	Phy-sio Bank ATM	Phy-sio Bank ATM	MIT-BIH	UCI KDD	NTOU	Bonn University
6	Telehealth System	Yes	No	No	No	No	No
7	Live Sensing Signals	No	No	No	No	No	No
8	Data Encryption	Yes	Yes	Yes	Yes	Yes	Yes
9	Data Compression	No	No	Yes	No	No	No
10	Secret Key Space Analysis	Yes	Yes	No	No	No	No
11	Histogram	Yes	Yes	No	No	No	No
12	Correlation	No	Yes	No	Yes	No	Yes
13	Autocorrelation	Yes	Yes	No	No	No	Yes
14	Plain Signal Sensitivity	Yes	Yes	No	No	Yes	No
15	Secret Key Sensitivity	Yes	Yes	No	No	No	No
16	Entropy Analysis	Yes	Yes	No	No	No	No
17	Floating Frequency	Yes	Yes	No	No	No	No
18	Chosen plain text attack	Yes	Yes	No	No	No	No
19	Differential attacks	Yes	No	No	No	No	No

Table 5 (continued)

Sl. No.	Comparative Parameters	Pro-posed technique	Ref [30] 2017	Ref [6]	Ref [32]	Ref [3]	Ref [21]
20	Mean square error MSE	Yes	Yes	No	Yes	No	No
21	Pick signal to noise ration PSNR	Yes	Yes	No	No	No	No
22	Signal to noise ration SNR	No	No	Yes	No	No	No
23	Structural similarity index SSIM	Yes	Yes	No	No	No	No
24	Power spectral density	No	No	No	No	No	Yes
25	Encryption Time Analysis	Yes	Yes	No	No	Yes	No
26	Pseudo-randomness Analysis	Yes	Yes	No	No	No	No
27	AVAL Effect	Yes	No	No	No	No	No
28	Strict Avalanche Effect	Yes	No	No	No	No	No
29	Bit Independence Test	Yes	No	No	No	No	No
30	Comparative Study	Yes	Yes	No	No	No	No

Table 6 Average values of avalanche, strict avalanche and bit independence comparisons

Technique	Avg. Value of Avalanche	Avg. Value of Strict Avalanche	Avg. Value of Bit Independence
Proposed	0.9759632	0.9729145	0.7591037
AES [25]	0.9999469	0.9996540	0.7211989
TDES [25]	0.9999142	0.9996324	0.7147735

ticks, there might be a small difference in actual time. This variation is minimal and can be disregarded. Using the interface of telehealth, patients can provide their clinical reports and necessary data for transmission to doctors. The interaction time needed is a vital parameter for the acceptance or rejection of any system. The encryption time and decryption time should be minimized to accept or reject the proposed system efficiently. The encryption and decryption time of the proposed technique, existing secret sharing scheme, and current benchmark methods are shown in Table 4.

7.14 Comparison analysis

Chaos-based encryption techniques on clinical signals ECG / EEG / etc presented in the literature survey section have some pros and cons within themselves. Security implementations and further enhancements in the telehealth system are the most indispensable and sensitive criteria. While encrypting any patients' clinical signals, the system should be capable of resisting the different attacks on the transmitting component signals. The proposed system handles the integral features as compared with other validated techniques of others. A brief comparative study has been shown in Table 5.

7.15 Avalanche and strict avalanche

A comparison is made between the source and the encrypted signal, and a shift of bits in the encrypted signal was observed for a single bit change in the original signal for a whole or a very large number of bytes. Table 6 demonstrates the comparison of the average values of proposed and current benchmark encryption techniques.

8 Conclusion and future scope

In conclusion, recommendations on precautionary measures to prevent cybersecurity risks and policies on the revision of sustainability and development problems and responses to risks to human rights violations can be introduced at national and international levels in this COVID-19 pandemic. In this paper, the security of the existing telehealth system is enhanced using the Tree Parity Machine based key generation technique. Here, a concept of the privileged recipient has also been introduced for preserving the confidentiality of the patient's information. The original information is reconstructed after *OR*-ing exactly k or more number of shares. Among these k number of shares, one should be from privileged recipients. Original information can't be reconstructed without the participation of privileged share. As future work, a more comprehensive analysis of security is planning to carry out. Also, different nature-inspired optimization algorithms will be considered for the optimization of weights value of TPM for faster synchronization purposes.

Acknowledgments The author(s) expressed deep gratitude for the moral and congenial atmosphere support provided by Ramakrishna Mission Vidyamandira, Belur Math, India.

Author Contributions Arindam Sarakar for study design, literature search. Moumita Sarkar for analysis and interpretation and overall content and manuscript revision.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declarations

Competing interests The authors declare that they have no competing interests.

References

1. Acharya UR, Acharya D, Bhat PS, Niranjan UC (2001) Compact storage of medical images with patient information. *IEEE Trans Inf Technol Biomed* 5(4):320–323. <https://doi.org/10.1109/4233.966107>
2. Ahmad M, Farooq O, Datta S, Sohail SS, Vyas AL, Mulvaney D (2011) Chaos-based encryption of biomedical EEG signals using random quantization technique. In: et al. (ed) 4th International Conference on Biomedical Engineering and Informatics (BMEI), vol 3, pp 1471–1475. <https://doi.org/10.1109/BMEI.2011.6098594>
3. Alkhodaidi T, Gutub A (2020) Refining image steganography distribution for higher security multimedia counting-based secret-sharing. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-020-09720-w>
4. Capua CD, Meduri A, Morello R (2010) A smart ECG measurement system based on web-service-oriented architecture for telemedicine applications. *IEEE Trans Instrum Meas* 59(10):2530–2538. <https://doi.org/10.1109/tim.2010.2057652>
5. CDC (2020) (COVID-19) situation summary. <https://www.cdc.gov/coronavirus/2019-ncov/index.html>
6. Chattopadhyay AK, Nag A, Singh JP (2020) A verifiable multi-secret image sharing scheme using XOR operation and hash function. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-020-09174-0>
7. England PH (2020) <https://publichealthmatters.blog.gov.uk/2020/03/04/coronavirus-covid-19-what-is-socialdistancing/>
8. Gupta M, Gupta M, Deshmukh M (2020) Single secret image sharing scheme using neural cryptography. *Multimedia Tools and Applications* 79(17-18):12183–12204. <https://doi.org/10.1007/s11042-019-08454-8>
9. HIPAA (2020) Advice for healthcare organizations on preventing and detecting human-operated ransomware attacks. <https://www.hipaajournal.com/advice-for-healthcare-organizations-on-preventing-and-detecting-human-operated-ransomware-attacks/>
10. HIPAA (2020) CISA issues fresh alert about ongoing APT group attacks on healthcare organizations
11. HIPAA (2020) EFF warns of privacy and security risks with google and apple's COVID-19 contact tracing technology. <https://www.hipaajournal.com/eff-privacy-security-risks-covid-19-contact-tracing-technology/>
12. HIPAA (2020) FBI issues flash alert about COVID-19 phishing scams targeting healthcare providers. <https://www.hipaajournal.com/fbi-issues-flash-alert-about-covid-19-phishing-scams-targeting-healthcare-providers/>
13. HIPAA (2020) FBI warns of increase in COVID-19 related business email compromise scams. <https://www.hipaajournal.com/fbi-warns-of-increase-in-covid-19-related-business-email-compromise-scams/>
14. HIPAA (2020) Government healthcare agencies and COVID-19 research organizations targeted by nigerian BEC Scammers. <https://www.hipaajournal.com/government-healthcare-agencies-and-covid-19-research-organizations-targeted-by-nigerian-bec-scammers/>
15. HIPAA (2020) Hackers target WHO, HHS, and COVID-19 research firm. <https://www.hipaajournal.com/hackers-target-who-hhs-and-covid-19-research-firm/>

16. HIPAA (2020) INTERPOL issues warning over increase in ransomware attacks on healthcare organizations. <https://www.hipaajournal.com/interpol-issues-warning-over-increase-in-ransomware-attacks-on-healthcare-organizations/>
17. HIPAA (2020) Scammers target healthcare buyers trying to purchase PPE and medical equipment. <https://www.hipaajournal.com/scammers-target-healthcare-buyers-looking-to-purchase-ppe-and-medical-equipment/>
18. HIPAA (2020) Zoom security problems raise concern about suitability for medical use. <https://www.hipaajournal.com/zoom-security-problems/>
19. Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan. *China Lancet* 395:497–506
20. Jakovljevic M, Bjedov S, Jaksic N, Jakovljevic I (2020) COVID-19 pandemia and public and global mental health from the perspective of global health securit. *Psychiatr Danub* 32(1):6–14
21. John Blesswin A, Raj C, Sukumaran R et al (2020) Enhanced semantic visual secret sharing scheme for the secure image communication. *Multimed Tools Appl* 79:17057–17079. <https://doi.org/10.1007/s11042-019-7535-2>
22. Liaqat RM, Mehboob B, Saqib NA, Khan MA (2016) A framework for clustering cardiac patient's records using unsupervised learning techniques. *Procedia Computer Science* 98:368–373. <https://doi.org/10.1016/j.procs.2016.09.056>
23. Lin F (2016) Chaotic visual cryptosystem using empirical mode decomposition algorithm for clinical EEG signals. *J Med Syst* 40(3):1–10
24. Lin F, Shih SH, Zhu JD (2014) Chaos based encryption system for encrypting electroencephalogram signals. *J Med Syst* 38(5):1–10
25. Lindell Y, Katz J (2014) Introduction to modern cryptography. Chapman and Hall/CRC
26. Lucey DR, Gostin LO (2016) The emerging zika pandemic: enhancing preparedness. *JAMA* 315(9):865–871
27. McKibbin WJ, Fernando R (2020) The global macroeconomic impacts of COVID-19: seven scenarios. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.3547729>
28. Meneses F, Fuertes W, Sancho J (2016) RSA encryption algorithm optimization to improve performance and security level of network messages. *IJCSNS* 16(8):55–55
29. Mulyadi IH, Nelmiawati N, Supriyanto E (2019) Improving accuracy of derived 12-lead electrocardiography by waveform segmentation. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)* 7(1):2089–3272. <https://doi.org/10.11591/ijeel.v7i1.937>
30. Murillo-Escobar MA, Cardoza-Avendaño L, López-Gutiérrez RM, Cruz-Hernández C (2017) A double chaotic layer encryption algorithm for clinical signals in telemedicine. *J Med Syst* 41(4):59–59. <https://doi.org/10.1007/s10916-017-0698-3>
31. Nabyev V, Ulutaş M, Ulutaş G (2010) Secret sharing scheme to implement medical image security. In: 2010 IEEE 18th signal processing and communications applications conference, Diyarbakir, pp 820–823
32. Nag A, Singh JP, Singh AK (2020) An efficient Boolean based multi-secret image sharing scheme. *Multimed Tools Appl* 79:16219–16243
33. Nayak J, Bhat PS, Acharya UR, Niranjana UC (2004) Simultaneous storage of medical images in the spatial and frequency domain: a comparative study. *BioMed Eng OnLine* 3:1–10
34. Nayak J, Bhat PS, M Sathish Kumar RAU (2009) Efficient storage and transmission of digital fundus images with patient information using reversible watermarking technique and error control codes. *J Med Syst* 33(3):163–171. <https://doi.org/10.1007/s10916-008-9176-2>
35. NIST (2020) NIST statistical test. http://csrc.nist.gov/groups/ST/toolkit/rng/stats_tests.html
36. PhysioBank (2020) Physiobank ATM. Online. <https://physionet.org/>
37. Priya S (2020) <https://www.opengovasia.com/singapore-government-launches-covid-19-chatbot/>
38. Raeiatibanaadkooki M, Quchani SR, Khalilzade M, Bahaadinbeigy K (2016) Compression and encryption of ECG signal using wavelet and chaotically huffman code in telemedicine application. *J Med Syst* 40(3):1–8. <https://doi.org/10.1007/s10916-016-0433-5>
39. Shivani S, Rajitha B, Agarwal S (2017) XOR based continuous-tone multi secret sharing for store-and-forward telemedicine. *Multimedia Tools and Applications* 76(3):3851–3870. <https://doi.org/10.1007/s11042-016-4012-z>
40. Spencer A, Patel S (2019) Applying the data protection act 2018 and general data protection regulation principles in healthcare settings. *Nurs Manag* 26(1):34–40. <https://doi.org/10.7748/nm.2019.e1806>
41. RA U, Bhat PS, Kumar S, Min LC (2003) Transmission and storage of medical images with patient information. *Comput Biol Med* 33(4):303–310. [https://doi.org/10.1016/s0010-4825\(02\)00083-5](https://doi.org/10.1016/s0010-4825(02)00083-5)
42. UC Niranjana RAU, Iyengar SS, Kannathal N, Min LC (2004) Simultaneous storage of patient information with medical images in the frequency domain. *Comput Methods Prog Biomed* 76(1):13–19. <https://doi.org/10.1016/j.cmpb.2004.02.009>

43. WHO (2020) WHO director-general's opening remarks at the media briefing on COVID. <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19>
44. WHO (2020) World Health Organization. Critical preparedness, readiness and response actions for COVID-19. <https://www.who.int/publications-detail/critical-preparedness-readiness-and-response-actions-for-covid-19>
45. Williams AM, Bhatti UF, Alam HB, Nikolian VC (2018) The role of telemedicine in postoperative care. *mHealth* 4:11–11. <https://doi.org/10.21037/mhealth.2018.04.03>
46. Woodley M (2020) <https://www1.racgp.org.au/news/p/clinical/clinical-chief-medical-officer-flags-coronavirus-telehealth>
47. Worldometer (2020) <https://www.worldometers.info/coronavirus/>
48. Zhu N, Zhang D, Wang W, Li X, Yang B, Song J (2019) A novel coronavirus from patients with pneumonia in China. *New England J Med*

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