Arabian Journal for Science and Engineering (2021) 46:3301–3317
https://doi.org/10.1007/s13369-020-05136-8

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Neural Synchronization-Guided Concatenation of Header and Secret Shares for Secure Transmission of Patients' Electronic Medical Record: Enhancing Telehealth Security for COVID-19

Arindam Sarkar[1] · **Moirangthem Marjit Singh**[2] · **Jyotsna Kumar Mandal**[3]

Received: 13 July 2020 / Accepted: 11 November 2020 / Published online: 2 January 2021

Abstract
This paper deals with one of the key problems of e-healthcare which is the security. Patients are worried about the confidentiality of their electronic medical record (EMR) which could be used to expose their identities. It is high time to revisit the confidentiality and security issues of the existing telehealth system. Intruders can perform sniffing, spoofing, or phishing operations effortlessly during the online exchange of the EMR using a digital platform. The EMR must be transmitted anonymously with a high degree of hardness of encryption by protecting the authentication, confidentiality, and integrity criteria of the patient. These requirements recommend the security of the current system to be improved. In this paper, a neural synchronization-guided concatenation of header and secret shares with the ability to transmit the EMR with an end-to-end security protocol has been proposed. This proposed methodology breaks down the EMR into the n number of secret shares and transmits to the n number of recipients. The original EMR can be reconstructed after the amalgamation of a minimum k (threshold) number of secret shares. The novelty of the technique is that one share should come from a specific recipient to whom a special privilege is given to recreate the EMR among such a predefined number of shares. In the absence of this privileged share, the original EMR cannot be reconstructed. This proposed technique has passed various parametric tests. The results are compared with existing benchmark techniques. The results of the proposed technique have shown robust and effective potential.

Keywords Telehealth · Security · COVID-19 · Artificial neural networks (ANNs) · Electronic medical record (EMR) · Secret share

1 Introduction

COVID-19 is a serious acute respiratory disease caused by the form of coronavirus that occurred in Wuhan, China, in December 2019, and rapidly spreading around the world, leading to the pandemic of COVID-19 [1,2]. Many countries are currently affected by the COVID-19 pandemic, which often endangers public health and also impacts other aspects of human life, particularly the global economy [3]. Presently, as the world deals with an unforeseen COVID-19 pandemic [4], the recent trend of social distancing [5] is to work from home with lockdowns and travel limitations. 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 [6]. Owing to the increase risk of disease infection by personal interaction, the regulation of the COVID-19 through minimizing direct contact is possible by the telehealth sys-

✉ Arindam Sarkar
arindam.vb@gmail.com

Moirangthem Marjit Singh
marjitm@gmail.com

Jyotsna Kumar Mandal
jkm.cse@gmail.com

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

[2] Department of Computer Science and Engineering, North Eastern Regional Institute of Science and Technology, Nirjuli, Arunachal Pradesh 791109, India

[3] Department of Computer Science and Engineering, University of Kalyani, Kalyani, West Bengal 741235, India

tem. In December 2019, the "future" of e-healthcare began [7].

Global COVID-19 is making the first telehealth trial for millions around the world. 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 [8]. E-healthcare bridges the gap between individuals, clinicians, and healthcare systems, allowing everyone, especially symptomatic patients, to remain at their home [9] and interact remotely with doctors, helping to minimize virus transmission to large populations and frontline medical workers. One of the significant applications of this system is to monitor the patients following the hospital release [10], which can also be used for COVID-19 patients. The quality of e-healthcare also depends on patients' data security. The confidentiality of the patients' data must be ensured. Telehealth practitioners have found this a major challenge [11]. Data security and confidentiality are the key issues to be addressed before communicating patients' data to the health workers.

The rest of this paper is structured accordingly. The background of the work is described in Sect. 2. Section 3 presents related works. Sections 4 and 5 discuss the problem domain and solution domain, respectively. Section 6 deals with the proposed methodology, and Sect. 7 represents the session key generation through the synchronization of the two artificial neural networks. Section 8 discusses the proposed technique using an example. Section 9 represents the time complexity analysis of the proposed technique. Section 10 deals with results and analysis. In Sect. 11, the conclusion and future scope are given and references are given at the end.

2 Background

Telehealth must be one of the alternative systems for healthcare and promoting health decision-making within the framework of improving service delivery [12]. In COVID-19, there is a need for proper medical diagnosis [13] at the right time so that the patients get exact and concrete medical advice from physicians, which improves the curable probabilities. In improving the health systems response to COVID-19, World Health Organization (WHO) cited [14] telehealth as an important service. Telehealth is perhaps the most demanding part of the e-governance and offers remote patients low-cost and efficient healthcare. E-healthcare is characterized as all clinicians utilizing Information and Communication Technology (ICT) to share relevant information for the diagnosis, treatment when the distance is a crucial factor in the delivery of healthcare services. In the hopes of improving individuals and their communities' health [15], the prevention of illness and injury, study and evaluation, and continuing outreach of healthcare providers.

The US legislation providing data protection and security measures for the protection of health information is HIPAA (Health Insurance Portability and Accountability Act, 1996). According to HIPAA, ransomware operators have carried out extensive attacks on healthcare data. Spam e-mails claiming to provide details of SARS-CoV-2 and COVID-19 cases were distributed to NetWalker during the COVID-19 pandemic [16]. Organizations with a ransomware can be tempted to pay the fee to reduce downtimes and save costs. However, Sophos' survey shows that businesses paying the bill ultimately invest much more than those who recover files from backups [17]. SARS-CoV-2 and COVID-19 research organizations are warned that hackers associated with the People's Republic of China (PRC) threaten their networks [18]. Sophisticated National State hackers have taken advantage of some vulnerabilities to target government and industry entities to obtain entry to one's systems and seize confidential information. Online meeting platform Zoom enters an agreement with NY Attorney General on security and privacy issues [19]. Corporate e-mail intrusion scammers from around the globe threaten hospitals, COVID-19 research organizations [20]. Advanced Persistent Threat (APT) groups are calling for another joint alarm from cybersecurity officials in the USA and the UK to thwart a continuous attempt to target the e-health security system. [21,22]. Cyberattacks on remote employees has risen dramatically, with application protocols being actively targeted by remote employees to communicate with business networks [23]. To strengthen security while operating remotely, the National Security Agency has provided information security guides for teleworkers. The guide was written specifically for US government agencies, and this also applicable for health professionals who offer telehealth services from their home computers or smartphones. During the COVID-19 pandemic, attacks on healthcare and vital infrastructure by ransomware were increased. [24]. Apple and Google are developing contact tracking technologies to track people reportedly contacting COVID-19, but the Electronic Frontier Foundation (EFF) has cautioned that cybercriminals are misusing this contact tracing technology [25]. WHO is one of the leading organizations fighting the COVID-19. WHO confirmed that hackers tried to access there network by spoofing [26]. There have been hundreds of attacks on health organizations [27–30]. According to a new report published by Proofpoint, more than a half million e-mails, 300,000 malicious URLs, and over 200,000 malicious e-mail attachments have been analyzed recently [31]. Attackers are performing different types of phishing, spoofing, and sniffing attacks on online meeting platforms, COVID-19 research organizations, e-healthcare systems [32–35].

3 Related Works

Nayak et al. [36] 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. [36,37] 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. [38–40] 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. [41] 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 [42] has described a chaotic encoding method of EEG signal along with a view toward logistic maps and experimental forms of decomposition. Lin et al. [43] developed a new encryption scheme. However, the security analysis was missed out. Ahmad et al. [44] 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. [45] have proposed how to improve the accuracy of a 12-lead ECG using waveform segmentation of ECG graphs. Liaqat et al. [46] 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. [47] 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. [48] 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 Problem Domain

In the e-healthcare system, the main problem is the security of patients' EMR during online transmission. The EMR of each patient is highly confidential. To maintain this confidentiality clause, there needs a highly secured encryption technique about patient's data integrity. Intruders are born to sniff the patients' confidential information/signals quietly, and they do distort or manipulate with these. The existing e-healthcare security system has several drawbacks they are as follows:

- The existing e-healthcare security system may treat the possibility of willing disclosure of patients' EMR and clinical signals for any sort of illegal works.
- Even the patients have no mechanism to resist the disclosure of her/his confidential EMR on a broader scale.
- Lack of data authentication and confidentiality checking mechanisms exist in the telehealth system. Hence, the data integrity of the patients cannot be achieved [36,37].
- Leakage of such important clinical signals may lead to misuse.
- No telehealth system exists in the literature that has a concept of priority or privileged share.
- In many of the latest e-healthcare protection strategies, complete encrypted information is stored in a single medium. Secret data can be exposed if the medium or key is lost or compromised.

5 Solution Domain

In this proposed work, the above-stated problems have been addressed. In this proposed neural synchronization-guided concatenation of header and secret share-based technique, patient has been treated as a privileged recipient. It is not possible to intentional sharing of patients' private EMR and clinical signals because the original information cannot be regenerated excluding the concern of the patient. 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 is no risk of losing or corrupting entire data due to medium or key corruption.

Hence, this proposed security model 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 EMR.

6 Proposed Methodology

In this methodology, a robust session key is generated through the synchronization of two artificial neural networks. In different sessions, different session keys are being used for secured transmission. A mask is generated to generating the n number of shares of the patient's EMR. Then each share is ciphered through the neural session key. A header is formed by collecting secret information including the session key. This header is divided into n number of shares using a header mask. Each header share is concatenated to each secret share. Finally, these concatenated shares are individually encrypted using the recipient's public key.

Proposed Algorithm

Requirement: $Patient's\ EMR$

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

Output: $n\ number\ of\ secret\ shares\ of\ EMR$

Method: $Mask\_Generation(\ )$ helps to generate a secret mask. The binary form of the EMR is then AND-ed with a mask to create confidential shares. These partially opened shares get encrypted using a neural session key. Then the header shares are concatenated with encrypted secret shares. Finally, these concatenated shares are distributed to the n numbers of intended recipients by encrypting them using the public key of the corresponding recipient.

{/* 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*/} i
$Share[n][n_{C_{k-1}}] =$
$Call\ SecretShare\_Generation$
$(Mask[n][n_{C_{k-1}}\ ],\ EMR[p][q])$;
{/*Encryption of Partially open shares*/}
$EncryptedShare[i][j] = Call\ \ ShareEncryption$
$(Share[n][\ ],\ EncryptionKey[\ ])$;
{/*Formation of Header Mask*/}
$HeaderMask[i][j] = Call$
$HeaderMask\_Generation$
$(EncryptedShare[n][\ ])$;
{/*Public Key Encryption of Encrypted Key with Privilege Recipient's Public key*/}
$PrivilegeEncryptedKey[\ ] = Call$
$PublicKeyEncryption\ ($
$EncryptionKey[\ ],$
$PrivilegeRecepient'spublicKey)$;
{/*Header Generation*/}
$Header\ [\ ] = \{n,\ k,\ PrivilegeEncryptedKey[\ ],$
$EncryptedShareSize[n]\}$;
{/*Header Share Generation */}
$HeaderShare\ [n]\ [k] = Call$
$HeaderShare\_Generation$
$(HeaderMask[n]\ [k], Header[\ ])$;
{/*Concatenation of Header Shares with Compressed Share */}
$FinalShare[i]\ [j] = Call$
$FinalShare\_Generation$
$(EncryptedShare[n]\ [\ ],\ EncryptedShareSize[n],$
$HeaderShare[n]\ [k])$;

{/*Public Key Encryption of Encrypted shares with Recipient's Public key*/}
$TransmittableFinalShare[i] =$
$Call\ PublicKeyEncryption$
$(FinalShare[n],\ Recipient'spublicKey[n])$;
Report n transmittable shares

6.1 Mask Generation

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 $Mask\left[n_{C_{k-1}}\right][n]$ 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 EMR 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 EMR. 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 EMR 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., $\frac{n!}{(k-1)!\times(n-k-1)!}$.

The total number of missing and secret data bits is $n - 1_{C_{k-2}}$ 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 is $n - 1_{C_{k-2}}$ and the number of 1's is $n - 1_{C_{n-k}}$. In the compressed share, the 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 1s and $(k - 1)$ number of 0s 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. $P = n_{C_k},\ Ceiling\_Bar,\ S,\ NO1 = 3,\ k1 = 0,\ k2 = 0,\ Mid,\ i,\ j$: Integer
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. $\quad BIN[\ n]\ \leftarrow Decimal2Binary\_Convert\ (\ i\ );$
6. $Check\ if\ One\_Count\ (BIN\ [n]\ )\ =\ NO1$
7. $Index[\ k1++]\ \leftarrow\ i\ ;$
8. $\quad end\ if$
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. $While\ (\ i <=\ Mid\ \&\&\ j <=\ P - 1\ )\ do$
14. $\quad Index2[k2++] \leftarrow\ Index[i];$
15. $\quad Index2[\ k2++] \leftarrow\ Index\ [j];$
16. $\quad increment\ i;$
17. $\quad increment\ j;$
18. $end\ while$
19. $For\ S\ =\ 0\ to\ (\ P - 1)\ do$
20. $Mask[S][\ ] \leftarrow\ Decimal2Binary\_Convert\ (Index[Index2[S]]);$
21. $end\ for$
22. $Mask\ [\ n]\ [\ P]\ \leftarrow\ [Mask[P][n]]^T$
23. $ReturnMask\ [\ n]\ [\ P]$

6.2 Secret Share Generation

A modular approach concerning shares generation is about dismantling original EMR into smaller partial ciphered secrets, with no secret contains complete information. The essential part of this kind of share generation is that each share has some sort of features of cohesion as well as coupling. If the EMR (M) may be broken into partial shares $A,\ B,\ C,\ D$, and E, then the concept of modularity is maintained. The biggest advantage of such an application of modular concept concerning the shares generations is that if any share is being corrupted during transmission, then that particular share can be regenerated and retransmitted. The loss or distortion of shares may happen in the network due to various reasons like noise, impulse, inversion of bits, etc. The demerit of applying the technique of modularity on shares generations is that the time complexities will increase both at the sender end and reassembly end. 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 secret shares

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

{/*AND-ing of EMR with Secret Mask to generate Secret Shares*/}

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

The above-stated algorithm can also be performed blockwise. 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 cannot 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 the last block, records are combined to construct a secret records.

6.3 Encryption of Partially Open Shares

The 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 intermediate encrypted shares and neural synchronized encryption key.

{/*Encryption of Partially open shares*/}

1. $for\ i\ =\ 1\ to\ n\ do$
2. $\quad for\ j\ =\ 1\ to\ n_{C_{k-1}}\ do$
3. $\quad\quad EncryptedShare[i][j] = Share[i][j]\ XOR\ EncryptedKey[j]$
4. $\quad\quad increment\ j$
5. $\quad end\ for$
6. $\quad increment\ i$
7. $EncryptedKey[\ ] = EncryptedKey[\ ] \gg j$
8. $end\ for$

6.4 Header Mask Matrix Generation

k numbers of nonzero sample bytes are now obtained from each of the n ciphered shares, and thus, header matrix of dimension $(n\ \times k)$ is formed. The header mask generation algorithm is discussed as follows.

$HeaderMask\_Generation\ (EncryptedShare[n][k])$

Input: $Share[n][k]$

Output: Header Mask Matrix.

Method: Generate a mask matrix of dimension $n_{C_{k-1}} \times n$ have $(n-k+1)$ number of 1s and $(k-1)$ number of 0s 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 $n_{C_{k-1}}$

{/*Formation of Header Mask Matrix*/}

1. $for\ i\ =\ 1\ to\ n\ do$
2. $\quad for\ j\ =\ 1\ to\ k\ do$
3. $\quad HeaderMask[i][j] = EncryptedShare[i][j]$
4. $\quad\quad increment\ j$
5. $\quad end\ for$
6. $\quad increment\ i$
7. $end\ for$

6.5 Grant Privilege to the Desire Recipient

In this proposed technique, a concept of the privileged recipient has been introduced for preserving the patients' EMR. The original EMR is reconstructed after OR-ing exactly k or more number of shares (including the share of privileged recipient). Original EMR cannot be constructed without the participation of privileged recipient's share. To implement this logic, $EncryptetionKey[\ ]$ is encrypted using the privileged recipient's public key using any public key encryption [49]. So, the privileged recipients only can decode the encryption keys of secret shares to get the original EMR.

$PublicKeyEncryption\ (EncryptetionKey[\ ],\ PrivilegeRecepient's publicKey)$

6.6 Generation of Header Shares

The n numbers of header shares are generated by multiplying the header with the $n\ \times\ k$ header mask using $\sum_{j=0\ to\ k-1}^{i=0\ to\ n-1} x[i][j] \times Header[j]$ Following is the step by step demonstration of header share generation.

$HeaderShare\_Generation(\ HeaderMask[n][k],\ Header[\ ])$

Input: $HeaderMask[n][k],\ Header[\ ]$

Output: n numbers of Header shares

Method: generate header shares by multiplying header mask with header.

{/*Header Share Generation */}

1. $for\ i\ =\ 1\ to\ n\ do$
2. $\quad for\ j\ =\ 1\ to\ k\ do$
3. $\quad\quad HeaderShare[i][j]\ =\ HeaderMask[i][j] \times Header[j];$
4. $\quad\quad\quad increment\ j$
5. $\quad end\ for$
6. $\quad increment\ i$
7. $end\ for$

6.7 Concatenation of Encrypted Shares and Header Shares

Each header share is concatenated with the corresponding encrypted share, which forms one complete share. The following algorithm generates the final share.

$FinalShare\_Generation(EncryptedShare[n][\ ],\ EncryptedShareSize[n],\ HeaderShare[n][k])$

Input: $EncryptedShare[n][\ ],\ EncryptedShareSize[n],\ HeaderShare[n][k]$

Output: Concatenated shares

Method: Perform concatenation between encrypted share and header share

{/*Concatenation of Header Shares with Compressed Share */}

1. $for\ i\ =\ 1\ to\ n\ do$
2. $\quad size =\ EncryptedShareSize[i]\ +\ k$
3. $\quad for\ j\ =\ 1\ to\ EncryptedShareSize[i]\ do$
4. $\quad\quad\quad\quad\quad FinalShare[i][j]\ =\ EncryptedShare[i][j];$
5. $\quad\quad\quad increment\ j$
6. $\quad end\ for$
7. $\quad\quad for\ p\ =\ j\ to\ size\ do$
8. $\quad\quad\quad FinalShare[i][j] = HeaderShare[i][j];$
9. $\quad\quad\quad increment\ p$
10. $\quad end\ for$
11. $\quad increment\ i$
12. $end\ for$

6.8 Public Key Encryption of Final Shares

The most important part is to take into account that intermediate encrypted shares are ultimately encrypted with the corresponding public keys of the designated recipients. Any public key algorithm can be used for this purpose. In this proposed model, assigning the n number of shares to an exact n number of recipients can be generalized using a fundamental mathematical combinational optimization technique. As discussed in this model, the general form of the problem is as follows:

There are n number of shares and $k$number of tasks in the problem. Any share can be assigned to any recipient, provided the constraint $key\ function\ (F)$ which is defined as the public key encryption applied based on corresponding recipients' public key. It has been designed in such a way that a recipient will be exactly assigned only a single share $\forall shares;\ F(n_i\ ) = Recipent_i\ (public\ key)$.

Mathematically, the formal mathematical definition of the linear assignment problem is given two sets, n and r, of equal size, combined with a $key\ function\ F : n \times r \rightarrow R$. Such kind of assigning a share to the corresponding recipient is termed as the one-to-one correspondence of shares. The following steps are represented the encryption of final share.

$PublicKeyEncryption$
$(FinalShare[n],\ Recipient'spublicKey[n])$

Input: $FinalShare[n],\ Recipient'spublicKey[n]$

Output: n number of transmittable shares.

Method: Perform public key encryption of n number of final shares with the respective recipient's public key.

{/*Public Key Encryption of final shares with Recipient's Public key*/}

1. $for\ i\ =\ 1\ to\ n\ do$
   $TransmittableFinalShare[i] =$
2. $PublicKeyEncryption$
   $(FinalShare[i],\ Recipient'spublicKey[i])$
3. $increment\ i$
4. $end\ for$

At the recipient end, the objective is to regenerate the original EMR. The threshold value of shares is bitwise OR-ed to get the original EMR as discussed earlier. Three different fields like frame header, encrypted secret data, and padding are to be extracted out. The concept of subnetting is applied to achieve this task. Suppose the recipient wants to extract the padding field of 16 bytes. It is being done by bitwise XOR operation. Binary EMR being XOR-ed with another arbitrary string that contains all 1s of 16 bytes from the least significant bit position and the rest of the bits are filled with zeros. The resultant of this operation shows a copy of the padding field only of length 16 bytes. By repeating the same operation, other three attributes can also be obtained.

7 Artificial Neural Network Synchronization

A special artificial neural network architecture called TPM (tree parity machine) is used at patient and caregiver end. These two TPMs start mutual synchronization with each other using a common input vector and random synaptic weights vector. After each step, each TPM exchanges their final output and uses suitable learning rules to update their corresponding weights. After full synchronization, both the TPMs generate identical weight vector. This identical synaptic weight vector serves as an encryption key to encrypt the partially open shares.

TPM is composed of M number of input neurons for each H number 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 $+Lrange$,$weight_{u,v} \in \{-Lrange, -Lrange + 1, \ldots, +Lrange\}$. In TPM uth hidden unit is described by the index $u = 1, \ldots, H$ and that of $v = 1, \ldots, M$ denotes input neuron corresponding to the uth 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 Eq. 1.

$$hidden_u = \frac{1}{\sqrt{M}} WEIGHT_u.XINPUT_u = \frac{1}{\sqrt{M}} \sum_{v=1}^{M} weight_{u,v}\ xinput_{u,v} \tag{1}$$

$signum(hidden_u)$ define the output $\sigma output_u$ of the uth hidden unit. (in Eq. 2),

$$\sigma output_u = signum(hidden_u) \tag{2}$$

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 Eq. 3).

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

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

$$\tau output = \prod_{u=1}^{H} \sigma output_u \tag{4}$$

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

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

$\tau output = \sigma output_1$, if only one hidden unit ($H = 1$) is there. $\tau output$ value can be the same for 2^{H-1} different ($\sigma output_1$, $\sigma output_{2,\ ...,}\ \sigma output_H$) 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 Eq. 6).

$$weight^{+}_{u,v} = fn(weight_{u,v} + xinput_{u,v}\tau output\Theta(\sigma output_u \tau output)\ \Theta(\tau output^{TPM\_A}\tau output^{TPM\_B})) \quad (6)$$

In the anti-Hebbian learning rule, both TPMs are learned with the reverse of their output (in Eq. 7).

$$weight^{+}_{u,v} = fn(weight_{u,v} - xinput_{u,v}\tau output\Theta(\sigma output_u \tau output)\ \Theta(\tau output^{TPM\_A}\tau output^{TPM\_B})) \quad (7)$$

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 Eq. 8).

$$weight^{+}_{u,v} = fn(weight_{u,v} + xinput_{u,v}\Theta(\sigma output_u \tau output)\ \Theta(\tau output^{TPM\_A}\tau output^{TPM\_B})) \quad (8)$$

If $X = Y$, then $\Theta\,(X, Y) = 1$ Otherwise, if $X = Y$ 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 Eq. 9).

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

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.

8 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 1s with three or more shares, but some positions still have 0s 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 RKMVCSAS20 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 R0M0C0A020
Share 2 : 1011110100 R0MVCS0S00
Share 3 : 1100011110 RK000SAS20
Share 4 : 0111001101 0KMV00AS00
Share 5 : 0101110011 0K0VCS0020

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 : RMCA20
Share 2 : RMVCSS
Share 3 : RKSAS2
Share 4 : KMVAS0
Share 5 : KVCS20

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 perform the encryption of partially open shares.

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

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

$$HeaderMask[i][j] = Call\ HeaderMask\_Generation\ (EncryptedShare[n][\,])$$

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.

$$PrivilegeEncryptedKey[\,] = Call\ PublicKeyEncryption\ (\ EncryptionKey[\,],\ Privilege Recepient's public Key);$$

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

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

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

$$HeaderShare\ [n]\ [k] = Call\ HeaderShare\_Generation\ (HeaderMask[n]\ [k], Header[\,])$$

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.

$$TransmittableFinalShare[i] = Call\ PublicKeyEncryption\ (FinalShare[n],\ Recipient's public Key[n])$$

9 Time Complexity Analysis

The time complexity of secret share generation technique may be calculated as i where n denotes the number of shares prepared for transmission. This 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 EMRs are 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 inter dependency 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 $\times$ number of hidden neurons) the amount of computations. For example, if the number of input neurons $(n) = 5$ and the number of hidden neurons $(k) = 6$ then total numbers of synaptic links (weights) are $(5\times6) = 30$. 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 in the form of O(initialization of input vector + initialization of weight vector + 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 weight 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(Time complexity in first iteration+(number of iteration $\times$ number of weight updation)). The $Mask\left[n_{C_{k-1}}\right][n]$ matrix is generated to generating the share of the EMR. This operation takes O($n_{C_{k-1}} \times n$) amount of time.

10 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 purpose. 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.

10.1 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 $[\ 8 + (3 \times n) + (256 \times n) + (256 \times n)]$ number of bits. So, $\frac{[\ 8+\ (3\times n)+\ (128\times n)+(128\times n)]}{8} = \left(1 + \frac{(3\times n)}{8} + 16n + 16n\right] = 32n$ to $\frac{[\ 8+\ (3\times n)+\ (256\times n)+(256\times n)]}{8} = \left(1 + \frac{(3\times n)}{8} + 32n + 32n\right] = 64n$ numbers of characters.

Consider any single encryption using the encryption key 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, USA, 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}. The total number of sec. a year has $= 365 \times 24 \times 60 \times 60 = 31, 536, 000$ 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.

10.2 Information Entropy Analysis

Entropy is a measure of a random sequence's unpredictability 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. In total, 1500 encrypted BP signals of 60 s 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%.

10.3 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 pseudovalues 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 show the histogram of plain and encrypted EMR, respectively. Since the encrypted EMR histograms are uniform, the approach proposed is robust against histogram-based statistical attacks.

10.4 Correlation Analysis

Correlation analysis has been used in the encryption algorithm to prove that the encrypted signal is uncorrelated. 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

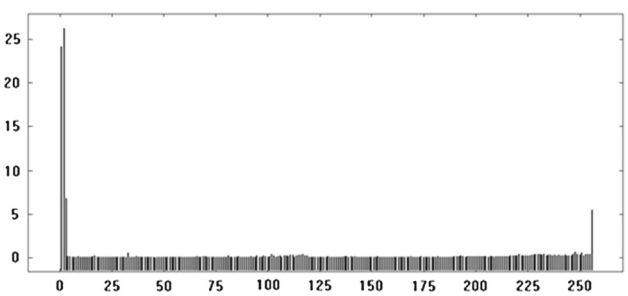

Fig. 1 Histogram of frequency distribution spectrum of input source stream characters

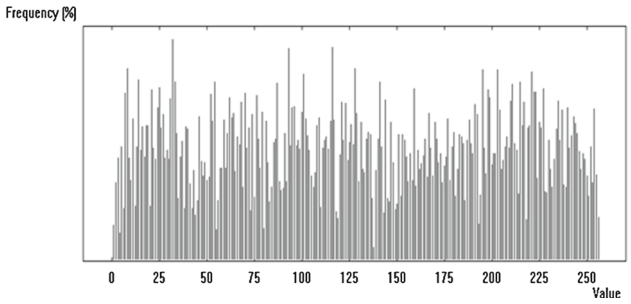

Fig. 2 Histogram of the frequency distribution continuum of encoded stream characters using the proposed technique

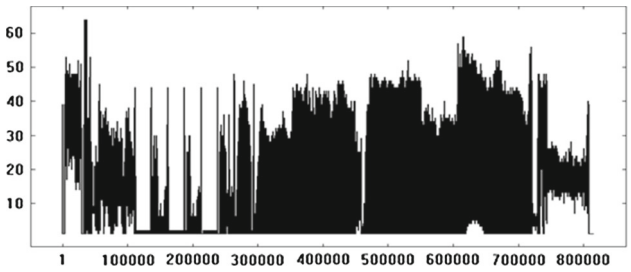

Fig. 3 Original signal's floating frequency

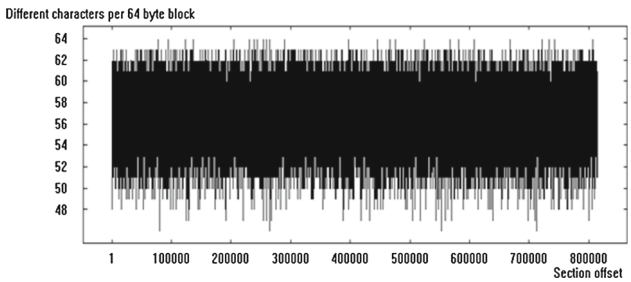

Fig. 4 Using the proposed method, the floating frequency of the encrypted signal

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.

10.5 Floating Frequency Analysis

The smoother or smaller curves in the frequency distribution spectrum indicate that it is more difficult for a cryptanalyst to detect the original text bytes, which means a better level of security. In Figs. 3 and 4, respectively, the floating frequency of plain and encrypted signals is shown. Well-distributed floating frequencies demonstrate the ruggedness of the encryption.

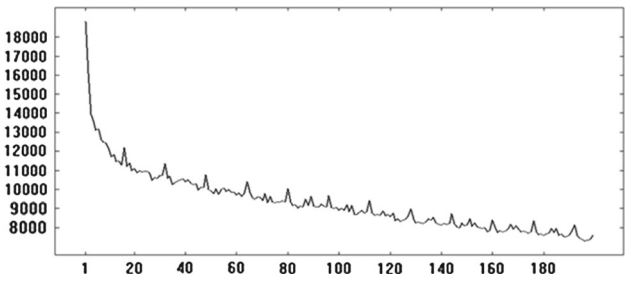

Fig. 5 Autocorrelation of the original signal

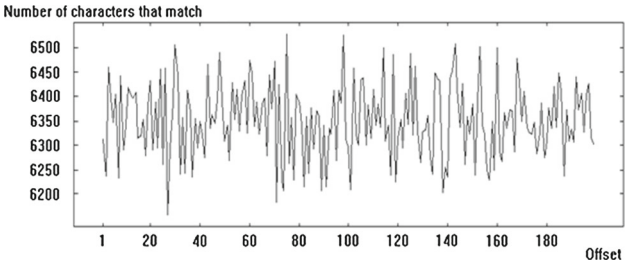

Fig. 6 Autocorrelation of the encrypted signal using proposed technique

10.6 Autocorrelation Analysis

The study of autocorrelation provides data on features such as periodicity, dependency, and repeated patterns in plain signal and encoded signal. The autocorrelation is known as the correlation between the signal and the shifted k positions themselves. For both a plain signal and an encrypted signal, the autocorrelation is measured at a bit level. Figures 5 and 6 show the autocorrelation of plain and encrypted signals, respectively. The findings show that a plain signal displays 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 uniform cryptograms. This is very important to avoid poor encryption windows that can be used in cryptanalysis in cryptosystems.

10.7 Quality Metrics Analysis

This section performed a quality review of the encoded ECG signal. It contains the mean square 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 Eq. 10.

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

The larger MSE value is often preferable by the encoding algorithm. PSNR reflects the ratio of the average pixel value

Table 1 Quality metrices estimation

Clinical signals	MSE	PSNR (dB)	SSIM
ECG [50]	10714.21	7.925	0.0429
EEG [50]	10124.16	8.213	−0.0261
BP [50]	11316.81	7.965	0.0721

to the compressed image as seen in Eq. 11.

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

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 Eq. 12.

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

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 Eq. 13, PRD is computed.

$$\text{PRD} = \sqrt{\frac{\sum_{i=1}^{L}\left(X_i - \widehat{X_i}\right)^2}{\sum_{i=1}^{L}(X_i)^2}} \times 100 \tag{13}$$

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

10.8 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, maybe the difference in fewer bits, result in two entirely different ciphertexts. However, a flip on a single bit between two encryption keys will generate two cryptograms on a completely different scenario. In total, 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 decrypt the signals.

10.9 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 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 s. 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.

10.10 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 Eq. 14.

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

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 con-

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 [50]	2.55×10^{-16}	82.16	137.89	307.34	805.72
ECG [50]	2.87×10^{-14}	19829.81	24470.56	48215.93	112137.04
EEG [50]	1.14×10^{-15}	397.09	510.73	854.25	3187.45

tains the results of NIST Statistical tests [51] on the generated random input vector.

10.11 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 decrypt the ciphertext. In the telehealth, attackers try to decrypt 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 resistant by the proposed technique against the chosen plain signal attack in the telehealth system.

10.12 Noise Attack

Changes in the shape of the transmitted ECG signals may mislead the cardiologist to diagnose the disease. Signals are distorted by the effect of the noise factor, termed as noise attack. The proposed technique is resistant to the said attack in the following way. The ECG signal has been distorted by imnoise( ) with noise value 0 as salt and 1 as pepper. Three different noise density cases are 0.017, 0.048, and 0.156 scaled at 2 %, 5 %, and 15 % of noise, respectively. In an ideal situation, mean square error, MSE should be nil, however, not realistic that one. MSE between original signal and decrypted signal is 96.43, 422.41, and 1267.54 for 2 %, 5 %, and 15 % of noise, respectively. Table 8 contains the percentage root mean square difference, PRD calculated at the different clinical signals received under noise percentage. Using the noise attack, the information transmitted from patient to doctor or vice versa may lose. By raising the sampling frequency, this stated problem of the lost signal may be addressed.

10.13 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.

10.14 Encryption/Decryption Time

All test programs for the algorithms have been designed to demonstrate the overall encryption and decryption time. 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 the time needed for the CPU clock 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 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 times of the proposed technique and current benchmark methods are shown in Table 4.

10.15 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

Table 3 NIST statistical test

NIST test	p value	Status
Frequency	0.557287	Success
Frequency within a block	0.580167	Success
Runs	0.517563	Success
Longest run of ones in a block	0.088907	Success
Binary matrix rank	0.680278	Success
Discrete Fourier transform	0.511908	Success
Non-overlapping template matching	0.459784	Success
Overlapping (periodic) template matching	0.221587	Success
Maurer's "universal statistical"	0.787421	Success
Linear complexity	0.668904	Success
Serial	0.543298	Success
Approximate entropy	0.290176	Success
Cumulative sums	0.696389	Success
Random excursions	0.350933	Success
Random excursions variants	0.234908	Success

Table 4 Comparisons of encryption/decryption time of the proposed technique with benchmark AES and TDES encryption techniques

Source file size (in bytes)	Proposed technique (in milliseconds)		AES technique [52] (in milliseconds)		TDES technique [52] (in milliseconds)	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
1,925,185	120	142	197	219	393	501
2,498,560	178	184	242	297	532	518
3,790,336	240	251	298	332	897	923
4,883,456	319	322	405	467	964	1051
5,456,704	337	350	487	517	1172	1168

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 is given in Table 5.

10.16 Avalanche and Strict Avalanche

A comparison being 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. Subtract the ratio of the measured standard deviation from the predicted value of 1.0 to achieve avalanche and a strict scale avalanche. To measure the bit independence principle, the coefficient of correlation between the jth and kth components of the output difference string is required. Table 6 demonstrates the comparison of the average values of avalanche, strict avalanche and bit independence between the proposed and current benchmark encryption techniques.

11 Conclusion and Future Scope

The security of the existing telehealth system is enhanced using the proposed neural session key generation technique. Here, a concept of the privileged recipient has also been introduced for preserving the confidentiality of the patient's EMR. The original EMR 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 EMR cannot be reconstructed without the participation of privileged share. Telehealth is already rising rapidly. The new standard will be the forced habits of today. Life in the AC (After Corona) age will never be the same again. Today's mandatory preferences will slowly shift to default mode. If face-to-face consultations is the standard and telehealth is an exception, patients cannot want to return to the BC (Before Corona) era. Extended Medicare coverage allows telehealth an efficient way to continue delivering services for patients. However, procedures will continue to give priority to patients' safety and data protection. Hackers use the COVID-19 crisis to attack with various viruses in the form of ransomware to steal data and/or bank malware such as Mustang Panda, Kimsuky, and many

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

Sl. no.	Comparative parameters	Proposed technique	Ref. [48] 2017	Ref. [41] 2016	Ref [42] 2016	Ref. [43] 2014	Ref. [44] 2011
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	Physio Bank ATM	Physio 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
20	Mean square error MSE	Yes	Yes	No	Yes	No	No
21	Pick signal-to-noise ratio PSNR	Yes	Yes	No	No	No	No
22	Signal-to-noise ratio 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	Pseudorandomness 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	0.97189467	0.9687704	0.7569857
AES [52]	0.9999469	0.9996540	0.7211989
TDES [52]	0.9999142	0.9996324	0.7147735

more. Fortunately, the world is a lot different from 1918. This proposed technique provides a better security approach to improve the e-healthcare system. The time has come for us to introduce those tools. COVID-19 provides the chance to remove all these obstacles. Years ago, telehealth was introduced. COVID-19 has speed up the process only. And the question now is how far are we prepared to move now?

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

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

Compliance with ethical standards

Conflict of interest No conflict of interest.

References

1. Zhu, N.; Zhang, D.; Wang, W.; Li, X.; Yang, B.; Song, J.; Zhao, X.; Huang, B.; Shi, W.; Lu, R.; Niu, P.; Zhan, F.; Ma, X.; Wang, D.; Xu, W.; Wu, G.; Gao, G.F.; Tan, W.: A novel coronavirus from patients with pneumonia in China, 2019. N. Engl. J. Med. **382**(8), 727–733 (2020). https://doi.org/10.1056/NEJMoa2001017
2. Huang, C.; Wang, Y.; Li, X.; Ren, L.; Zhao, J.; Hu, Y.: Clinical features of patients infected with 2019 novel coronavirus in Wuhan. China Lancet **395**, 497–506 (2020)
3. McKibbin, W.J.; Fernando, R.: The global macroeconomic impacts of COVID-19: seven scenarios. SSRN Electron. J. (2020). https://doi.org/10.2139/ssrn.3547729
4. Worldometer (2020). https://www.worldometers.info/coronavirus/
5. England, P.H. (2020). https://publichealthmatters.blog.gov.uk/2020/03/04/coronavirus-covid-19-what-is-socialdistancing/
6. Lucey, D.R.; Gostin, L.O.: The emerging zika pandemic: enhancing preparedness. JAMA **315**(9), 865–871 (2016)
7. Woodley, M.: (2020). https://www1.racgp.org.au/newsgp/clinical/chief-medical-officer-flags-coronavirus-telehealth
8. Priya, S.: (2020). https://www.opengovasia.com/singapore-government-launches-covid-19-chatbot/
9. Jakovljevic, M.; Bjedov, S.; Jaksic, N.; Jakovljevic, I.: COVID-19 pandemia and public and global mental health from the perspective of global health securit. Psychiatr. Danub. **32**(1), 6–14 (2020)
10. Williams, A.M.; Bhatti, U.F.; Alam, H.B.; Nikolian, V.C.: The role of telemedicine in postoperative care. mHealth **4**, 11–11 (2018). https://doi.org/10.21037/mhealth.2018.04.03
11. Spencer, A.; Patel, S.: Applying the data protection act 2018 and general data protection regulation principles in healthcare settings. Nurs. Manag. **26**(1), 34–40 (2019). https://doi.org/10.7748/nm.2019.e1806
12. WHO: World Health Organization. Critical preparedness, readiness and response actions for COVID-19 (2020). https://www.who.int/publications-detail/critical-preparedness-readiness-and-response-actions-for-covid-19
13. CDC: (COVID-19) situation summary (2020). https://www.cdc.gov/coronavirus/2019-ncov/index.html
14. WHO: WHO Director-General's opening remarks at the media briefing on COVID (2020). https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-
15. appropriations.senate: Senate directs aid to American people amidst coronavirus crisis, sends package to President's desk (2020). https://www.appropriations.senate.gov/news/senate-directs-aid-toamerican-people-amidst-coronavirus-crisis-sends-package-to-presidents-desk?inf_contact_key=3100f633e718effdea458ddb402d053d
16. HIPAA: NetWalker Ransomware Gang Targeting the Healthcare Industry (2020). https://www.hipaajournal.com/netwalker-ransomware-gang-targeting-the-healthcare-industry/
17. HIPAA: Study Indicated Payment of a Ransom Doubles the Recovery Cost (2020). https://www.hipaajournal.com/study-suggests-paying-a-ransom-doubles-the-cost-of-recovery-from-a-ransomware-attack/
18. HIPAA: Chinese Hacking Groups are Targeting COVID-19 Research Organizations (2020). https://www.hipaajournal.com/chinese-hacking-groups-are-targeting-covid-19-research-organizations/
19. HIPAA: FBI publish list of top 10 exploited vulnerabilities (2020). https://www.hipaajournal.com/cisa-and-fbi-publish-list-of-top-10-exploited-vulnerabilities/
20. HIPAA: Government healthcare agencies and COVID-19 research organizations targeted by Nigerian BEC scammers (2020). https://www.hipaajournal.com/government-healthcare-agencies-and-covid-19-research-organizations-targeted-by-nigerian-bec-scammers/
21. HIPAA: CISA issues fresh alert about ongoing APT group attacks on healthcare organizations (2020)
22. Physionet and Org (2016). https://physionet.org/cgi-bin/atm/ATM
23. HIPAA: Worldwide spike in brute force RDP attacks during COVID-19 pandemic (2020). https://www.hipaajournal.com/worldwide-spike-in-brute-force-rdp-attacks-during-covid-19-pandemic/
24. HIPAA: Advice for healthcare organizations on preventing and detecting human-operated ransomware attacks (2020). https://www.hipaajournal.com/advice-for-healthcare-organizations-on-preventing-and-detecting-human-operated-ransomware-attacks/
25. HIPAA: EFF warns of privacy and security risks with Google and Apple's COVID-19 contact tracing technology (2020). https://www.hipaajournal.com/eff-privacy-security-risks-covid-19-contact-tracing-technology/
26. HIPAA: WHO confirms fivefold increase in cyberattacks on its staff (2020). https://www.hipaajournal.com/who-confirms-fivefold-increase-in-cyberattacks-on-its-staff/
27. HIPAA: FBI issues flash alert about COVID-19 phishing scams targeting healthcare providers (2020). https://www.hipaajournal.com/fbi-issues-flash-alert-about-covid-19-phishing-scams-targeting-healthcare-providers/
28. HIPAA: Scammers target healthcare buyers trying to purchase PPE and medical equipment (2020). https://www.hipaajournal.com/scammers-target-healthcare-buyers-looking-to-purchase-ppe-and-medical-equipment/
29. HIPAA: INTERPOL issues warning over increase in ransomware attacks on healthcare organizations (2020). https://www.hipaajournal.com/interpol-issues-warning-over-increase-in-ransomware-attacks-on-healthcare-organizations/
30. HIPAA: FBI warns of increase in COVID-19 related business email compromise scams (2020). https://www.hipaajournal.com/fbi-warns-of-increase-in-covid-19-related-business-email-compromise-scams/
31. HIPAA: Novel coronavirus and COVID-19 themed attacks dominate threat landscape (2020). https://www.hipaajournal.com/2019-novel-coronavirus-and-covid-19-themed-attacks-dominate-threat-landscape/
32. HIPAA: Zoom security problems raise concern about suitability for medical use (2020). https://www.hipaajournal.com/zoom-security-problems/
33. HIPAA: Hackers target WHO, HHS, and COVID-19 research firm (2020). https://www.hipaajournal.com/hackers-target-who-hhs-and-covid-19-research-firm/
34. HIPAA: Cybersecurity best practices for protecting remote employees during the COVID-19 crisis (2020). https://www.hipaajournal.com/cybersecurity-best-practices-for-protecting-remote-employees/
35. HIPAA: Department of health and human services targeted in cyberattack (2020). https://www.hipaajournal.com/department-of-health-and-human-services-targeted-in-cyberattack/
36. Nayak, J.; Bhat, P.S.; Sathish Kumar, M.: 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 (2009). https://doi.org/10.1007/s10916-008-9176-2
37. Nayak, J.; Bhat, P.S.; Acharya, U.R.; Niranjan, U.C.: Simultaneous storage of medical images in the spatial and frequency domain: a comparative study. BioMed. Eng. OnLine **3**, 1–10 (2004)
38. Acharya, R.; Bhat, P.S.; Kumar, S.; Min, L.C.: Transmission and storage of medical images with patient information. Comput. Biol. Med. **33**(4), 303–310 (2003). https://doi.org/10.1016/s0010-4825(02)00083-5

39. Acharya, U.R.; Acharya, D.; Bhat, P.S.; Niranjan, U.C.: Compact storage of medical images with patient information. IEEE Trans. Inf. Technol. Biomed. **5**(4), 320–323 (2001). https://doi.org/10.1109/4233.966107
40. Niranjan, U.C.; Iyengar, S.S.; Kannathal, N.; Min, L.C.: Simultaneous storage of patient information with medical images in the frequency domain. Comput. Methods Programs Biomed. **76**(1), 13–19 (2004). https://doi.org/10.1016/j.cmpb.2004.02.009
41. Raeiatibanadkooki, M.; Quchani, S.R.; KhalilZade, M.; Bahaadinbeigy, K.: Compression and encryption of ECG signal using wavelet and chaotically Huffman code in telemedicine application. J. Med. Syst. **40**(3), 1–8 (2016). https://doi.org/10.1007/s10916-016-0433-5
42. Lin, F.: Chaotic visual cryptosystem using empirical mode decomposition algorithm for clinical EEG signals. J. Med. Syst. **40**(3), 1–10 (2016)
43. Lin, F.; Shih, S.H.; Zhu, J.D.: Chaos based encryption system for encrypting electroencephalogram signals. J. Med. Syst. **38**(5), 1–10 (2014)
44. Ahmad, M.; Farooq, O.; Datta, S.; Sohail, S.S.; Vyas, A.L.; Mulvaney, D.: Chaos-based encryption of biomedical EEG signals using random quantization technique, in: and others. In: 4th International Conference on Biomedical Engineering and Informatics (BMEI), pp. 1471–1475 (2011)
45. Mulyadi, I.H.; Nelmiawati, N.; Supriyanto, E.: Improving accuracy of derived 12-lead electrocardiography by waveform segmentation. Indones. J. Electr. Eng. Inform. **7**(1), 2089–3272 (2019). https://doi.org/10.11591/ijeei.v7i1.937
46. Liaqat, R.M.; Mehboob, B.; Saqib, N.A.; Khan, M.A.: A framework for clustering cardiac patient's records using unsupervised learning techniques. Procedia Comput. Sci. **98**, 368–373 (2016). https://doi.org/10.1016/j.procs.2016.09.056
47. Capua, C.D.; Meduri, A.; Morello, R.: A smart ECG measurement system based on web-service-oriented architecture for telemedicine applications. IEEE Trans. Instrum. Meas. **59**(10), 2530–2538 (2010). https://doi.org/10.1109/tim.2010.2057652
48. Murillo-Escobar, M.A.; Cardoza-Avendaño, L.; López-Gutiérrez, R.M.; Cruz-Hernández, C.: A double chaotic layer encryption algorithm for clinical signals in telemedicine. J. Med. Syst. **41**(4), 59–59 (2017). https://doi.org/10.1007/s10916-017-0698-3
49. Meneses, F.; Fuertes, W.; Sancho, J.: RSA encryption algorithm optimization to improve performance and security level of network messages. IJCSNS **16**(8), 55–55 (2016)
50. PhysioBank ATM. Online. https://physionet.org/
51. NIST statistical test. http://csrc.nist.gov/groups/ST/toolkit/rng/stats_tests.html
52. Lindell, Y.; Katz, J.: Introduction to Modern Cryptography. Chapman and Hall/CRC, London (2014)