Journal of Information Systems Engineering and Management
2025, 10(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Medical Data Security using Deep Learning based Key Generation, Quantum Key Exchange and Modified AES

Konka Kishan[1], A.Obulesu[2]

[1]Research Scholar, Department of CSE, JNTUH R&D Centre, Vidya Jyothi Institute of Technology, Himayat Sagar Rd, Hyderabad, Telangana-500075

[2]Associate Professor, Department of CSE, Vidya Jyothi Institute of Technology, Himayat Sagar Rd, Hyderabad, Telangana-500075

1kishan526@gmail.com, 2avuku06@gmail.com

ARTICLE INFO

Received: 10 Oct 2024
Revised: 10 Dec 2024
Accepted: 24 Dec 2024

ABSTRACT

Medical data security refers to the protection and safeguarding of sensitive patient data in the healthcare domain. It encompasses various measures and protocols designed to ensure the confidentiality, integrity, and availability of medical information. This includes not only medical imaging data but also electronic health records (EHRs), medical test results, patient demographics, and other personally identifiable information (PII). Medical data security is of utmost importance due to several reasons like patient privacy, preventing unauthorized access or disclosure of personal health information, risk of data breaches and identity theft. This paper presents a medical data security framework using deep learning based key generation, quantum key exchange and modified Advanced Encryption Standard (AES). Deep learning-based random number generation for encryption key is an approach that utilizes deep learning models to generate secure and unpredictable sequences of numbers that can be used as encryption keys. Traditional random number generators rely on algorithms or physical processes to generate randomness. Deep learning, on the other hand, offers an alternative approach by leveraging the power of neural networks to learn patterns and generate seemingly random sequences. This generated random number is used as key for quantum key exchange. The proposed framework uses BB84 protocol the utilizes principles of quantum mechanics to achieve secure key exchange. This ensures the confidentiality and integrity of the transmitted data. The AES algorithm is a widely used symmetric encryption algorithm that is known for its high level of security and efficiency. The conventional mix column operation is replaced with low complex algorithm for better performance.

Keywords: Medical data security, electronic health records, deep learning, key generation, quantum key exchange, Advanced Encryption Standard.

INTRODUCTION

The protection of sensitive patient information that is present inside medical scans is a primary concern in the healthcare business [1]. It is necessary to protect these images from being seen, altered, or disclosed by unauthorized individuals. There are many different elements that contribute to the need of medical image security. Maintaining patient anonymity is of the highest significance when it comes to the use of medical images, as they include patient-specific information. Unauthorized access to such information may result in breaches of patient confidentiality and put the patient's confidence at risk. In addition, protecting the integrity of the data is very important since any unauthorized modification or manipulation of medical images may lead to inaccurate diagnoses, improper treatments, or compromised patient safety [2].

Another key factor driving the need for medical image security is compliance with laws [3]. In India, healthcare practitioners and institutions are required to comply with the norms and standards for data protection that are set out by Medical Council of India (MCI). It is possible for healthcare organizations to

incur financial and reputational losses, in addition to legal implications, if they fail to deploy sufficient security measures. In addition, medical images are regularly utilized in research projects and partnerships, making it imperative that their confidentiality be maintained at all times. The safe and confidential communication of medical images helps build trust, promotes the sharing of data, and makes it easier to make advances in medical knowledge and patient care [4].

Artificial Intelligence (AI) is being used to solve the problems involved in the field of medical image security. AI has a number of skills that may strengthen and improve security measures. For example, AI systems may identify irregularities and potentially suspect behaviors inside medical imaging [5]. AI is able to spot possible security breaches by analyzing patterns and features, such as efforts to gain unauthorized access or modifications made to images. The use of access control and authentication techniques that are enabled by AI, such as face recognition, fingerprint authentication, watermarking etc, helps to guarantee that only authorized staff are able to access critical medical images, which in turn helps to strengthen security [6]. Algorithms created by AI have the ability to identify possible security flaws and vulnerabilities inside medical imaging systems [7, 8]. AI may warn administrators to possible threats, such as data breaches or unauthorized data transfers, by analyzing network traffic and detecting patterns of suspicious activity. This is done by recognizing patterns of suspicious behaviors.

LITERATURE SURVEY

Zhenlong Man et al [9] introduced a dynamic adaptive diffusion-based double image encryption method with a convolutional neural network (CNN). Compared to the present double image encryption approach, this methodology is different. Based on the properties of digital pictures, they developed a dual-channel encryption method that makes use of both a digital channel and an optical channel. This method increases the security of double images while simultaneously enhancing the efficiency of encryption and lowering the likelihood of future assaults. They used a chaotic map to regulate the starting values of a 5D conservative chaotic system to increase the encryption key's security. In a convolutional neural network, a chaotic sequence was also used as the convolution kernel, producing chaotic pointers that are connected to the plaintext. Attacks using known-plaintext and chosen-plaintext are effectively defended against by this approach. As a result, this approach makes it possible to manipulate how the two pictures are scrambled.

Deqiang Ouyang et al [10] proposed a novel controller that explicitly addresses the integration of "actuator overload" in order to impulsively synchronise linked delayed neural networks. The sector nonlinearity model approach is used in the research to evaluate abrupt controls using actuator saturation and partial actuator saturation. Through this study, some relevant sufficient criteria are found. Using a numerical programme, the theoretical results are checked. The author emphasises the need of spontaneous synchronisation for picture security in his concluding paragraph.

Shaochuan Xu et al [11] demonstrated a brand-new fractional-order chaotic system built on the four-neuron Hopfield Neural Network (HNN) model. The Adomain decomposition method was used to resolve the fractional-order chaotic system that was suggested in the paper. Various dynamic attributes were shown by the system when the orders were changed. The pseudo-random numbers (PRNs) produced by the proposed system were also used to create a unique technique for building a multiple hash index chain. Additionally, a novel method of picture encryption was developed employing a chain of several hash indexes. The security test results showed that the newly created encryption method offers improved security. Using Multisim circuit modelling, the 4-neurons-based HNN fractional-order system was finally put into practice.

Farhan Musanna et al [12] utilized a cellular neural network and a fractional-order chaotic system to develop a revolutionary digital picture encryption technique. The encryption is carried through via a permutation-substitution architecture based on chaos. The Merkel-Damgard concept served as inspiration

for key-generation, which makes up the majority of the contribution. Conway's game of life and the NARX network are used to carry out the diffusion process.

Malika Izabachène et al [13] investigated whether or not it would be practical to implement Fully Homomorphic Encryption (FHE) techniques in order to solve the problems that were outlined. The researchers investigated the viability of implementing FHE on a neural network while maintaining the confidentiality of the network's topology, weights, and user data inside the encrypted domain. They were able to do this by using Hopfield neural networks, which, in comparison to their feed-forward analogues, are more amenable to FHE. In addition, they suggested thinking about alternate neural network models, ones that are still applicable and more suited for FHE. This was done with the intention of improving the effectiveness with which real-world applications might be managed.

Bernardo Pulido-Gaytan et al [14] explored the fundamental concepts of Fully Homomorphic Encryption (FHE), investigates its practical applications in the real world, explores its practical applications in the real world, investigates state-of-the-art approaches, analyses the limitations, advantages, and disadvantages, and also discusses potential future applications and development tools, with a particular focus on neural networks. Over the course of the last several years, the area of FHE has seen a number of important developments. However, those who are seeking for viable methods to implement FHE generally depend on previously conducted research that is associated with homomorphic neural networks.

Jungha Jin et al [15] suggested a technology known as the three-dimensional (3D) cube approach, which may be used to increase security resilience in an environment including quantum computing, be used. In order to use this method, you will need to generate the key that corresponds to the most recent version of the symmetric key encryption scheme. To be more specific, it provides a solution for the secure key-sharing procedure that reduces the amount of harm that is incurred as a result of the disclosure of the pre-shared key (PSK). This is made possible by the method, which eliminates the need of transferring the PSK from one machine to another by making use of deep neural network learning to produce symmetric keys. This makes it easier to generate the key and put it to use inside the symmetric key encryption system that the 3D cube method employs, which simultaneously protects the key's confidentiality while enhancing the system's efficiency.

Xue Wei et al [16] presented KNEW Key Generation using Neural Networks from Wireless Channels is to produce keys that have a high agreement rate. This is accomplished by extracting the implicit qualities of the channel in a compressed format. In this method, two neural networks are trained concurrently to transform the channel estimates of each other into a distinct domain known as the latent space. This space is kept out of the reach of adversaries during the training process. The Key Distribution Rate (KDR) is improved as a result of the model's ability to narrow the gap that exists between the latent spaces produced by two trustworthy pairs of nodes. This is how the model accomplishes this goal.

C. Ismayil Siyad et al [17] proposed an innovative method for the generation of encryption keys at the physical layer that utilises a turbulent neural network and is influenced by Chua's chaotic dynamics. This approach has the ability to produce keys with a very high level of security. In addition to this, the paper suggests a whole new dual key transmitter and receiver for the physical layer security (PLS) protocol. For the purpose of determining whether or not the suggested system is successful, secrecy measures like "secrecy rate" and "secrecy outage probability" are used.

Arindam Sarkar et al [18] demonstrated the design of an ASIC that can allow re-keying in a Triple Layer Vector-Valued Neural Network (TLVVNN) using a tiny logic area. The 65 and 130 nm CMOS architectures are used in the design. It has not received much effort to optimise neural weights for quicker neural synchronisation. In this situation, Harris' Hawks are used to accelerate coordination by optimising the neural network's weight vector. The synchronised weight is used as the session key after this procedure is

finished. This method's ability to generate the session key via mutual brain synchronisation over a public channel is one of its benefits. It enables the neural weight vector to be optimised using Hawks, which speeds up neural synchronisation across public channels. According to behavioural modelling, a 20% weight imbalance might cause the time needed for synchronisation during the re-keying phase to decrease from 1.25 ms to less than 0.7 ms. Attacks using geometry, brute force, or a majority are all forbidden.

Yi Ding et al [19] proposed a unique network called DeepKeyGen that makes use of deep learning methods to create a private key for utilising a stream cypher to encrypt and decode medical pictures. A generative adversarial network (GAN) is used by DeepKeyGen's learning network to generate the private key. In order to aid the learning network in comprehending the procedure for producing the private key, the transformation domain—which stands for the intended "style" of the private key is also used. Learning the connection between the original picture and the private key is DeepKeyGen's major goal.

Arindam Sarkar et al [20] proposed the use of a multilayer neural network with synchronised session keys-based encryption for data or information transfer via radio. Multilayer perceptron transmission systems that accept identical input vectors, produce an output bit, and train the network based on this output bit are used in this system at both the sending and receiving ends. Then, a variable-length secret key is generated by the trained network, guaranteeing its security. A random hidden layer from the multilayer neural network is chosen at random for each session, and the weights or hidden units of this layer help create a private session key. The session key is produced by a multilayer perceptron, and it is used to encrypt the original text via chaining, cascading, along with XOR processes. The last block of plain text doesn't alter if its size is less than the key size. The recipient uses the same session key produced by the multilayer perceptron for decoding.

PROPOSED MODEL

The proposed section covers image security and watermarking in two sections. The first segment introduces a novel method for establishing large invisible color watermarks. The technique uses a cutting-edge medical image-specific conditional deep autoencoder model. Medical imaging is secured and authenticated using advanced deep learning algorithms to prevent modification and distribution. The second part discusses image security and an extensive strategy to protect medical images. Deep Learning-based Key Generation improves encryption and controls access to sensitive medical data. Quantum Key Exchange technology, which uses quantum physics to provide safe cryptographic keys, is also proposed. This quantum-enhanced security method strengthens picture protection. Finally, this research work gives a comprehensive approach to medical image protection. Advanced watermarking methods in the first part and a multidimensional security architecture in the second section are used to produce a complete and resilient solution for medical picture security and authenticity.

1.1 Image Watermarking

In recent days, there has been a lot of interest in the authentication of multimedia files as a method of avoiding unauthorized use, theft, and misrepresentation of the content. Invisible watermarking is the process of hiding information inside a medium in order to provide evidence of ownership, verify the medium's integrity, or keep a message a secret. The purpose of invisible watermarking is to hide the watermark so that it will not draw attention to the fact that the cover image has been watermarked. In this research, an invisible watermarking technology is presented. Using the proposed method [21], big color watermarks can be hidden inside the cover media. The watermark is encoded into the cover picture with the help of a Conditional Variational Autoencoder (CVAE), which is the model that was presented. After

arriving to the receiver, the picture that has been watermarked is decoded in order to retrieve the watermark.

A stacked autoencoder served as the inspiration for the design of the autoencoder that was proposed. Before presenting the proposed framework, a variety of autoencoders, such as stacked autoencoders, variational autoencoders, and conditional variational autoencoders, are discussed and analyzed in detail. Autoencoders use neural networks for the goal of representation learning and falls within the category of autoencoders. To be more specific, a neural network design is developed that, by means of the imposition of a bottleneck on the network, results in the induction of a compressed knowledge representation of the initial input. If the input attributes were entirely independent of one another, then compressing them and subsequently recreating them would be an extremely difficult task. It is possible to discover a structure in the data, such as correlations between the properties of the input, which may then be used to push the input beyond the bottleneck in the network, if one exists. The autoencoder is made up of several layers, each of which has nodes and edges.

The proposed method has the capability of embedding huge, full-color images as watermarks into the cover images. The traditional methods of watermarking in general hide a simple, small, monochromatic image as the watermark. The stego images generated a high PSNR that was more than 40 for each of the various watermarks, and the images are visually indistinguishable from the original cover images. After performing watermarking, the proposed encryption is applied to these images which is discussed in the next section.

1.2 Encryption

This section introduces a medical data security framework that combines deep learning-based key generation, quantum key exchange using the BB84 protocol, and a modified Advanced Encryption Standard (AES). The framework leverages deep learning models to generate secure and unpredictable encryption keys by learning patterns and generating random sequences. These keys are then used for quantum key exchange, ensuring secure key exchange based on the principles of quantum mechanics. The AES algorithm, known for its high security and efficiency, is employed with a modified low complexity algorithm replacing the conventional mix column operation for improved performance. Overall, the framework aims to safeguard the confidentiality and integrity of medical data through a combination of deep learning-based key generation, quantum key exchange, and modified AES encryption.

When it comes to the generation of keys for cryptographic systems, having true random numbers is very necessary to assure both security and resilience to assaults. The susceptibility of the system to assaults such as brute-force attacks and key-guessing attacks is increased when keys are generated using predictable or non-random patterns. The use of true randomness ensures that the keys will be unpredictable, which makes it more difficult to guess or infer them. This protects the integrity of sensitive data as well as the secrecy of sensitive information. Without really random numbers, cryptographic systems would be vulnerable to attack, putting at risk the confidentiality of sensitive information as well as the integrity of financial transactions and sensitive conversations.

1.2.1 Deep learning based random number generator

Traditional techniques for generating random numbers often make use of deterministic algorithms, which result in sequences that have the appearance of being random but may really be predicted or manipulated. Deep learning models, on the other hand, have the ability to both learn patterns and create numbers that seem to be random by employing complicated neural network structures. In order to generate random numbers using AI deep learning, one must first train a deep neural network on a huge dataset consisting of random numbers. The network is able to learn the statistical patterns and attributes of the training data, which enables it to create new numbers that display characteristics that are comparable to those of the

original data. The network is capable of being educated to produce numbers that fall inside a certain range or that follow a given distribution. The use of an AI that is based on deep learning to generate random numbers has the benefit of being able to create numbers that display complicated patterns and statistical qualities in a manner that is comparable to that of real-world random data.

The layers used in the proposed deep learning model for random number generation are:

- **Dense Layers**: Dense layers or fully connected layers, are essential structural components of neural networks. A dense layer is one in which every neuron is coupled to each and every neuron in the layer underneath it. In order to calculate the output of a dense layer, a weighted sum of the inputs must first be computed, and then an activation function must be applied to that total. During the training process, each neuron in a dense layer is given its own unique set of weights, which are then used by that neuron. These weights establish the significance and degree of power possessed by every input link. The activation function brings non-linearity into the network, which enables the network to learn complicated patterns and to generate predictions based on such patterns. To acquire higher-level representations of the input data, dense layers are often utilised in the intermediary layers of a neural network. The dimensionality of the output may be tuned by adjusting a hyperparameter that is the number of neurons that are contained inside a dense layer.
- **Dropout Layers**: Dropout layers are a regularisation method used to avoid overfitting in neural networks. Dropout layers are utilised in neural networks. When a model performs well on the training data but is unable to generalise to data that it has not seen before, this is an example of overfitting. Dropout is a technique that helps prevent overfitting by arbitrarily setting a certain percentage of the input units (neurons) to zero during the training process. During the training process, a dropout layer will randomly mask a portion of the input units, which will have the same effect as temporarily removing them from the network. since of this, the network is required to learn redundant representations since it is unable to depend on the existence of any one neuron. Masking neurons differently for each training sample helps avoid complicated co-adaptations between neurons and is done independently for each training example. Dropout layers provide a regularization impact, which is akin to introducing noise to the network. This effect helps to lessen the dependence on particular neurons and supports learning that is more resilient and universal. During the inference phase, also known as the testing or prediction phase, on the other hand, no units are discarded, and the whole network is used for predictive purposes.

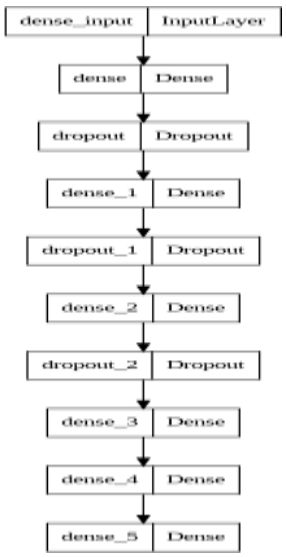


Figure 1: Proposed deep learning model

1.2.2 Quantum key generation and exchange

BB84 is a technique for quantum key distribution (QKD). BB84 was developed to facilitate the safe and confidential exchange of cryptographic keys between two parties, regardless of the security of the underlying communication channel. In order to guarantee the safety of the key exchange procedure, the protocol makes use of the concepts that may be found in quantum physics. For the purpose of establishing a shared key that is information-theoretically secure, it makes use of the basic aspects of quantum states, such as the concept of uncertainty and the superposition principle.

The following are the stages that are involved in the key exchange mechanism of BB84 algorithm. BB84 algorithm is based on quantum mechanics.

- In the first step, the sender generates a random key to form the basis. In the proposed model, the random sequence for the key is generated using a deep learning model. Each bit in the key may be encoded using either the vertical or horizontal polarization, or the diagonal (45 degrees to 135 degrees or 0 degrees to 90 degrees) polarization.
- The sender transmits the encoded bit sequence to the receiver via a quantum channel, which is a physical device that is capable of conveying quantum states, such as photons. Whenever the sender transmits a message, one of the two polarizations to assign to each bit is selected at random.
- When the receiver gets the data, a random sequence of bits is generated to act as basis. This basis is used to measure the polarization using either the vertical/horizontal or diagonal basis.
- In the last step, the receiver uses the selected basis to measure the polarization of each photon that has been received. Depending on the polarization of the photon, the measurement results are obtained that are either 0 or 1.

Table 1 outlines the protocol to be followed for the exchange of quantum keys.

Table 1: Quatum key exchange rules - Sender

Basis	Input Bit	Output
Z	0	0
Z	1	1
X	0	+
X	1	-

When bit 0 of the input data is set and Z is used as the basis, the output will be 0. When the basis is Z and the data bit in the input is 1, the result will be 1. The output is a plus sign if the data bit in the input is a zero and the basis is X. The output is a negative if the data bit in the input is 1 and the basis is X.

Table 2: Quantum key exchange rules - Receiver

Basis		Received Bit	Output
Z		0	0
Z		1	1
X		0	0/1
X		1	0/1
X		+	0
X		-	1
Z		+	0/1
Z		-	0/1

When bit 0 of the incoming data is 0, and Z is used as the basis, the output will be 0. When bit 1 of the data being received corresponds to Z as the basis, the output will be 1. When bit 0 of the incoming data is 0, and Z is used as the basis, the output will be 0. When the incoming data bit is 0 and the basis is X, the output is either 0 or 1, depending on which one of those two values is greater. When bit 1 of the received data is 1 and basis is X, the output will either be 0 or 1 depending on which. The output is 0 if the incoming data bit is a plus sign and the basis is X. The output is a one when the incoming data bit is a negative value and the basis is X. The output will either be 0 or 1 depending on whether or not the incoming data bit is a plus sign and the basis is Z. If the data bit that was received had a negative value and the basis was Z, then the output is either 0 or 1.

1.2.3 Simplified AES

The Advanced Encryption Standard (AES) is a symmetric encryption technique that is used to protect sensitive data. It is extensively utilized. The process of encrypting and decrypting data using AES is broken down into four primary steps: key expansion, byte substitution, shifting rows, and mixing columns.

1. **Key Expansion**: The first phase in the process involves expanding the original encryption key into a series of round keys. These round keys are then used for each round of encryption.
2. **Byte Substitution**: This phase involves replacing each byte of the input data with a matching byte taken from a predetermined substitution table known as the S-box. Because the S-box is intended to be resistant to a variety of cryptanalytic assaults, it will be difficult for an adversary to decrypt data that has been encrypted using it.
3. **Shift Rows**: The rows of the state array, which is a block of the input data, are shifted by a certain amount of bytes at this step. There is no shift made to the first row, a shift of one byte to the left is made to the second row, a shift of two bytes to the left is made to the third row, and a shift of three bytes to the left is made to the fourth row.
4. **Mix Columns**: This is the fourth phase, and it involves transforming each column of the state array by performing a matrix multiplication on it. This stage adds additional layer of protection to the encryption process and offers dispersion throughout the columns of the state array.

Since the MixColumns operation includes arithmetic operations over the Galois Field (GF) (2^8), it takes more calculations and memory than the other operations in AES, such as SubBytes and ShiftRows. Because of its intricacy, it is more difficult for potential attackers to crack the encryption by using brute-force assaults or any of the other cryptanalysis methods. A simpler alternative is presented as an option in the model that is being proposed in order to lessen the amount of computing effort that is required by the overall framework.

Algorithm: Low complex Galois field multiplication algorithm

Inputs: Two 8-bit numbers.
Outputs: Product of the inputs
Steps:

1. Initialize a variable **result** to zero.
2. While both **a** and **b** are non-zero:
    a. Check the least significant bit of **b**:
        - If it is set (equals 1), XOR **result** with **a**.
    b. Check the most significant bit of **a**:
        - If it is set (equals 1), left-shift **a** by 1 bit and XOR **a** with the irreducible polynomial 0x1b.
        - Otherwise, simply left-shift **a** by 1 bit.
    c. Right-shift **b** by 1 bit.
3. The final **result** is the Galois field multiplication of **a** and **b**.

This algorithm performs the Galois field multiplication by iteratively checking the bits of **b** and performing left-shift and XOR operations on **a** based on the bit values.

EXPERIMENTAL RESULTS

During the course of the experiment, the deep learning model is subjected to a total of 20 iterations of training and testing. Following the training of the model to generate keys of various sizes—16 bits, 32 bits, 64 bits, 128 bits, and 256 bits—during each iteration, the amount of time required for key creation is logged.

The purpose of the experiment is to investigate the effect that key size has on the amount of time needed for key creation. To do this, the bit sizes will be changed during each iteration. In the particular setting of the experiment, the results of this study may provide light on the trade-off that exists between the size of the key and its computing efficiency or performance.

The term "key generation time" (KGT) pertains to the duration required to create a key. A lengthier key generation time generally indicates a higher likelihood of the key being secure. This is because longer key generation times typically involve intricate and randomized procedures, making it more challenging for an adversary to guess or infer the key. Key length (KL) denotes the quantity of bits present in a key. Increased key lengths generally offer greater security compared to shorter ones as they provide a larger pool of potential bit combinations that an attacker would need to test in order to discover the key. For instance, a 128-bit key encompasses a staggering 2^128 possible combinations, an exceedingly vast number that would require an impractical amount of time for an attacker to attempt all the combinations.

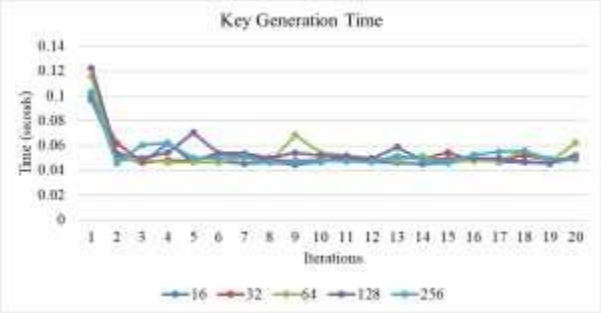


Figure 2: Key generation time

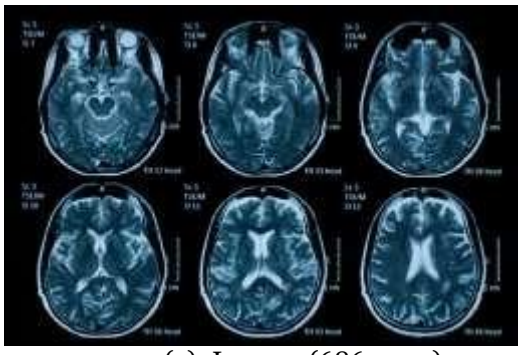
The average time of key generation is 50 milli seconds.

The conventional mix columns operation with Galois field multiplication has a fixed number of iterations (8 iterations) since it operates on 8-bit numbers in GF(2^8). It performs the multiplication using a loop that iterates 8 times, checking the bits of the multiplier **b** one by one. This fixed iteration approach ensures that the algorithm covers all possible bits of the 8-bit numbers. The simplified algorithm presented in section 3.3 reduces the complexity by removing the fixed number of iterations. Instead, it utilizes a **while** loop that continues until either **a** or **b** becomes zero. This allows the algorithm to terminate early if the remaining bits of **a** or **b** are zero, resulting in fewer iterations for certain inputs.

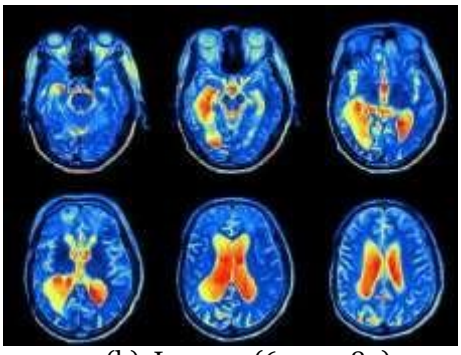
In terms of the irreducible polynomial used for reduction, the conventional algorithm uses **0x11b**, while the simplified algorithm uses **0x1b**. These polynomials have different degrees, but they both define irreducible polynomials for the Galois field GF(2^8). The difference in the irreducible polynomial affects the reduction step when the most significant bit of **a** is set. The polynomial **0x11b** requires a higher degree of reduction compared to **0x1b**, which results in additional XOR operations.

Overall, the simplified algorithm reduces the number of iterations compared to the initial algorithm by terminating the loop earlier when possible. However, the reduction step for the simplified algorithm may involve slightly more XOR operations due to the different irreducible polynomial.

In terms of complexity, the simplified algorithm generally performs better for inputs where **a** or **b** have fewer non-zero bits, resulting in fewer iterations. However, for inputs with more non-zero bits, the simplified algorithm may involve slightly more XOR operations. The overall difference in complexity is relatively minor, and both algorithms provide efficient Galois field multiplication for AES and similar applications.

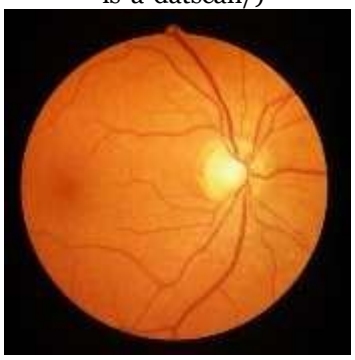


(a) Image 1 (686 x 455) (Source: https://www.universityofcalifornia.edu/news/medical-scanning-rates-continue-rise-despite-health-risks)



(b) Image 2 (640 x 480) (Source: https://www.independentimaging.com/what-is-a-datscan/)

(c) Image 3 (617 x 352) (Source: https://catcoolwall.blogspot.com/2018/02/mri-scan-vs-ct-scan.html)



(d) Image 4 (1024 x 1024) (Source: https://www.reviewob.com/retina-health-clinic-opens-inside-a-cvs/)

Figure 3: Input cover images

This table 1 shows the comparison results of existing and proposed modes results for input file size 4MB is taken as input. The performance comparison is evaluated using different parameters Encryption time (ET), Decryption time (DT) and Throughput (TP).

Table 1: Comparison results

	Conventional AES [22]	Proposed modified AES
ET (In Seconds)	1.5454	0.4997
DT (In Seconds)	0.2794	0.00261
TP (KB/Second)	2650.4464	8196.9181
Avalanche	0.5063497976532	0.500452163

CONCLUSION

This research paper proposed a medical data security framework that combines deep learning-based key generation, quantum key exchange, and a modified version of the AES. Deep learning is utilized to generate secure and unpredictable sequences of numbers for encryption keys, departing from traditional random number generators that rely on algorithms or physical processes. By leveraging neural networks, deep learning enables the generation of seemingly random sequences. These generated numbers serve as keys for the quantum key exchange process, which utilizes the BB84 protocol based on principles of quantum mechanics to achieve secure key exchange. This ensures the confidentiality and integrity of transmitted data. The AES algorithm, known for its high level of security and efficiency, is employed in the proposed framework. To enhance performance, a low-complexity algorithm replaces the conventional mix column operation. By combining these techniques, the framework aims to address the critical aspects of medical data security, protecting patient privacy, ensuring secure key exchange, and maintaining the integrity of sensitive information.

REFERENCES

[1] Albahri, A. S., Ali M. Duhaim, Mohammed A. Fadhel, Alhamzah Alnoor, Noor S. Baqer, Laith Alzubaidi, O. S. Albahri et al. "A systematic review of trustworthy and explainable artificial intelligence in healthcare: Assessment of quality, bias risk, and data fusion." Information Fusion (2023).

[2] Javaid, Mohd, Abid Haleem, Ravi Pratap Singh, and Rajiv Suman. "Towards insighting cybersecurity for healthcare domains: a comprehensive review of recent practices and trends." Cyber Security and Applications (2023): 100016.

[3] Shuaib, Mohammed, Shadab Alam, Mohammad Shabbir Alam, and Mohammad Shahnawaz Nasir. "Compliance with HIPAA and GDPR in blockchain-based electronic health record." Materials Today: Proceedings (2021).

[4] Haleem, Abid, Mohd Javaid, Ravi Pratap Singh, Rajiv Suman, and Shanay Rab. "Blockchain technology applications in healthcare: An overview." International Journal of Intelligent Networks 2 (2021): 130-139.

[5] Hadjiiski, Lubomir, Kenny Cha, Heang-Ping Chan, Karen Drukker, Lia Morra, Janne J. Näppi, Berkman Sahiner et al. "AAPM task group report 273: Recommendations on best practices for AI and machine learning for computer-aided diagnosis in medical imaging." Medical Physics 50, no. 2 (2023): e1-e24.

[6] Paul, Metty, Leandros Maglaras, Mohamed Amine Ferrag, and Iman AlMomani. "Digitization of healthcare sector: A study on privacy and security concerns." ICT Express (2023).

[7] Diaz, Oliver, Kaisar Kushibar, Richard Osuala, Akis Linardos, Lidia Garrucho, Laura Igual, Petia Radeva, Fred Prior, Polyxeni Gkontra, and Karim Lekadir. "Data preparation for artificial intelligence in medical imaging: A comprehensive guide to open-access platforms and tools." Physica medica 83 (2021): 25-37.

[8] Lee, DonHee, and Seong No Yoon. "Application of artificial intelligence-based technologies in the healthcare industry: Opportunities and challenges." International Journal of Environmental Research and Public Health 18, no. 1 (2021): 271.

[9] Man, Zhenlong, Jinqing Li, Xiaoqiang Di, Yaohui Sheng, and Zefei Liu. "Double image encryption algorithm based on neural network and chaos." *Chaos, Solitons & Fractals* 152 (2021): 111318.

[10] Ouyang, Deqiang, Jie Shao, Haijun Jiang, Sing Kiong Nguang, and Heng Tao Shen. "Impulsive synchronization of coupled delayed neural networks with actuator saturation and its application to image encryption." *Neural Networks* 128 (2020): 158-171.

[11] Xu, Shaochuan, Xingyuan Wang, and Xiaolin Ye. "A new fractional-order chaos system of Hopfield neural network and its application in image encryption." *Chaos, Solitons & Fractals* 157 (2022): 111889.

[12] Musanna, Farhan, Deepak Dangwal, and Sanjeev Kumar. "Novel image encryption algorithm using fractional chaos and cellular neural network." *Journal of Ambient Intelligence and Humanized Computing* (2022): 1-22.

[13] Izabachène, Malika, Renaud Sirdey, and Martin Zuber. "Practical fully homomorphic encryption for fully masked neural networks." In *Cryptology and Network Security: 18th International Conference, CANS 2019, Fuzhou, China, October 25–27, 2019, Proceedings*, pp. 24-36. Cham: Springer International Publishing, 2019.

[14] Pulido-Gaytan, Bernardo, Andrei Tchernykh, Jorge M. Cortés-Mendoza, Mikhail Babenko, Gleb Radchenko, Arutyun Avetisyan, and Alexander Yu Drozdov. "Privacy-preserving neural networks with homomorphic encryption: C hallenges and opportunities." *Peer-to-Peer Networking and Applications* 14, no. 3 (2021): 1666-1691.

[15] Jin, Jungha, and Keecheon Kim. "3D CUBE algorithm for the key generation method: applying deep neural network learning-based." *IEEE Access* 8 (2020): 33689-33702.

[16] Wei, Xue, and Dola Saha. "KNEW: Key Generation using NEural Networks from Wireless Channels." In *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, pp. 45-50. 2022.

[17] Ismayil Siyad, C., and S. Tamilselvan. "Chaotic deep neural network based physical layer key generation for massive MIMO." *International Journal of Information Technology* 13 (2021): 1901-1912.

[18] Sarkar, Arindam. "A symmetric neural cryptographic key generation scheme for Iot security." *Applied Intelligence* (2022): 1-24.

[19] Ding, Yi, Fuyuan Tan, Zhen Qin, Mingsheng Cao, Kim-Kwang Raymond Choo, and Zhiguang Qin. "DeepKeyGen: a deep learning-based stream cipher generator for medical image encryption and decryption." *IEEE Transactions on Neural Networks and Learning Systems* 33, no. 9 (2021): 4915-4929.

[20] Sarkar, Arindam. "Multilayer neural network synchronized secured session key based encryption in wireless communication." *IAES International Journal of Artificial Intelligence* 8, no. 1 (2019): 44.

[21] Kishan, Konka, and B. Vijay Kumar. "Efficient large invisible color watermark embedding using conditional deep autoencoder model for medical applications." *Measurement: Sensors* (2023): 100850.

[22] Zhang, Yong. "Test and verification of AES used for image encryption." 3D Research 9 (2018): 1-27.