sensors

MDPI

Article

A Deep Cryptographic Framework for Securing the Healthcare Network from Penetration

Arjun Singh [1], Vijay Shankar Sharma [1,*], Shakila Basheer [2] and Chiranji Lal Chowdhary [3,*]

[1] Department of Computer and Communication Engineering, Manipal University Jaipur, Jaipur 303007, India; arjun.singh@jaipur.manipal.edu
[2] Department of Information Systems, College of Computer and Information Science, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia; sbbasheer@pnu.edu.sa
[3] School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore 632014, India
* Correspondence: vijayshankar.sharma@jaipur.manipal.edu (V.S.S.); chiranji.lal@vit.ac.in (C.L.C.)

Abstract: Ensuring the security of picture data on a network presents considerable difficulties because of the requirement for conventional embedding systems, which ultimately leads to subpar performance. It poses a risk of unauthorized data acquisition and misuse. Moreover, the previous image security-based techniques faced several challenges, including high execution times. As a result, a novel framework called Graph Convolutional-Based Twofish Security (GCbTS) was introduced to secure the images used in healthcare. The medical data are gathered from the Kaggle site and included in the proposed architecture. Preprocessing is performed on the data inserted to remove noise, and the hash 1 value is computed. Using the generated key, these separated images are put through the encryption process to encrypt what they contain. Additionally, to verify the user's identity, the encrypted data calculates the hash 2 values contrasted alongside the hash 1 value. Following completion of the verification procedure, the data are restored to their original condition and made accessible to authorized individuals by decrypting them with the collective key. Additionally, to determine the effectiveness, the calculated results of the suggested model are connected to the operational copy, which depends on picture privacy.

Keywords: cryptography; healthcare; twofish security; hash function

check for updates

Citation: Singh, A.; Sharma, V.S.; Basheer, S.; Chowdhary, C.L. A Deep Cryptographic Framework for Securing the Healthcare Network from Penetration. *Sensors* **2024**, *24*, 7089. https://doi.org/10.3390/s24217089

Academic Editor: Alessandra Rizzardi

Received: 14 September 2024
Revised: 27 October 2024
Accepted: 31 October 2024
Published: 4 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Intending to identify imperfections, vulnerability assessment is a proactive method of locating and taking advantage of weaknesses in information assets from an attacker's viewpoint [1]. Penetration testing is essential to ensure the reliability, accessibility, and traceability (IACR) of information security in today's digital environment [2]. This is particularly accurate given that the European General Data Protection Regulation has become legally binding for companies and associations. A wide range of vulnerability testing alternatives are available [3]. Platforms such as Kali Linux, equipped with security tools like Nmap, offer specialized tools for conducting attacks. The penetration testing process identifies and evaluates system vulnerabilities and determines suitable remedial actions [4]. They imitate several kinds of system assaults. Through these procedures, the examiner can systematically and carefully identify the vulnerabilities. These produce information on vulnerabilities in the systems that need to be fixed so that management is aware of the problems [5]. This serves as a risk assessment for verifying network security. Organizations find penetration testing crucial, yet the process and resources are expensive [6]. Thus, a specialized penetration testing technique is required to safeguard devices and systems and guarantee network and information security quickly and affordably [7]. Internet usage is becoming more and more common. For this reason, data security is crucial to thwarting

cybercriminals' attempts [8]. Specialists perform penetration tests to identify and address network vulnerabilities before criminals attempt to exploit them [9].

A network may be a LAN, WLAN, WAN, or Internet of Things network. An internet penetration examination was recommended as an ethical measure to assess the potential hazards in the event of unauthorized entry into the company's technological infrastructure or network [10]. Furthermore, this authorized cyber-attack simulation helps develop a strategy to fix security flaws in the IT infrastructure before the real assault occurs. The so-called ethical hackers, skilled security professionals, handle it [11]. Protecting data and ensuring general security are the goals of network penetration testing, particularly when it comes to handling sensitive data. Traditional viruses or malware, improperly set fire-walls, and SQL injections are a few examples [12]. To guarantee long-term security, certain policies also require ongoing maintenance and network penetration testing [13].

Network security penetration technologies research focuses on data processing, approach and method development, and modeling network attack detection [14]. In particular, data mining technology is crucial to network attack detection techniques [15]. Vulnerability detection technology finds flaws in host systems or remote hosts, creates repair plans, and guarantees the security of computer systems [16]. It helps generate security standards to prevent network harm and gives staff members in charge of network security management in-depth knowledge about system vulnerabilities [17]. Nevertheless, current vulnerability detection algorithms frequently identify group vulnerabilities without determining their connection [18]. Conventional vulnerability detection methods are limited to identifying known vulnerabilities, and simplistic vulnerability risk assessments fail to accurately depict the network's overall security status [19]. Conventional network security technology cannot effectively protect intricate network architectures [20]. The Health Insurance Portability and Accountability Act of 1996 (HIPAA) safeguards medical coverage for employees and their families who experience employment changes or loss of medical coverage. It restricts new health plans from denying coverage based on pre-existing conditions. It also mitigates healthcare fraud and abuse, reforms medical liability, and simplifies administration by mandating the establishment of national standards for internet medical services and national identities for providers, hiring managers, and health insurers. It establishes regulations for pre-tax healthcare expenses accounts. It implements modifications to health insurance legislation and adjustments for medical insurance deductions. It provides regulations for collective health insurance schemes that offer adjustments for health insurance. It regulates coverage for life insurance policies owned by the company. It establishes regulations for the treatment of individuals lacking legitimacy in the United States and rescinds financial institution regulations pertaining to interest apportionment.

A technique for assessing network security associations is suggested to tackle this issue. In previous research works, the crypto models were merged with the mathematical model to achieve the finest outcome. However, in some cases, decryption is not possible for those instances. Here, the key novelty of the work is designing the intelligent cryptosystem GCbTS to secure medical image data. Before this research study, the graph convolutional system with twofish was not implemented in any study, so it is the prime novelty of the system. This approach is more accurate in reflecting network security status and is based on conventional vulnerability detection. The following is a summary of this work's contribution:

- Initially, medical image data were evaluated and trained according to the Python system;
- Consequently, a novel GCbTS is designed with the required crypto-security constraints;
- Henceforth, the hash 1 is found, and the data are encrypted and kept in an unreadable format;
- A hash 2 calculation is done to decrypt the data, verifying whether hash 1 is equal to hash 2;
- If hash 1 and hash 2 are equal, the secret key is shared for decryption;
- Finally, metrics like encryption and decryption time, PSNR, MSE, error rate, and computation time are compared with other models.

2. Related Work

To apply reinforcement learning (RL) techniques to the field of cyber security, Ghanem et al. [21] presented an innovative strategy. Using a hierarchical representation of the complex POMDP domain is the best way to solve large Partially Observable Markov Decision Process (POMDP) problems when scaling up is a problem. By separating the system into security clusters, IAPTF gains the ability to manage every one of them independently. This comprehensive approach covers all potential basic and advanced attack techniques, guaranteeing a success rate that matches or surpasses that of a professional moral hacker. The IAPTF is a flexible and all-encompassing framework that reveals complex situations and frees human specialists from monotonous jobs. The proposed technique's inherent drawback is its limited coverage of attack vectors, which could lead to the absence of certain advanced attack vectors that a human hacker could exploit.

Preetha et al. [22] suggested a blockchain-enabled smart contract security framework for medical applications, reducing the decryption overhead and preventing unwanted access to medical records through data exchange. To improve decryption performance, the system uses a validation contract to assess participant rights and a decryption contract for partial decryption. The system performs better than conventional systems regarding file loss rate, latency, and security. The solution lowers the cost of scheduling and transmitting facsimiles, improving network performance. However, the plan still has issues with network enactment and security enactment. According to experimental data, the approach can offer safe and dependable medical record exchange with fine-grained access control. Before widespread deployment, there are still important issues to resolve, such as human factors that could limit digital platform usage.

Alabbad et al. [23] provided a brand-new penetration testing framework that draws inspiration from software engineering's parallel testing methodology. The framework takes a target network's collection of resources and policies and applies an algorithm to create a secure network. The network is evaluated based on its security procedures, segmentation strength, and compliance with the DD strategy compared to a robust network of a similar nature. The approach suits dynamic networks due to its automation capabilities. Ensuring the security of the transferred data between identical twins and target networks is of utmost importance. Two key concerns are the reliability of the permissions policies and the trustworthiness of the risk evaluation suggestions for improvement.

Specifically, Hidayanto et al. [24] introduced a computerized mapping tool that enhances the efficiency of the penetration testing process, especially for complex websites with numerous functionalities. A technique used by the OWASP Juice Shop and Startrek Payroll requires only 0.21 s to find individual test cases and 0.07 s to transfer specific endpoints. In contrast, manually converting requires 12 min and 1.79 s. Level customization improves the process time by mapping fewer test cases than the original level. The JSON-stored mapping pattern facilitates the modification of vulnerability patterns and adds test cases from other standards for penetration testers. This program also makes it easy to record test results. With 100% accuracy, automatic mapping [24] finds every potential test case for every endpoint more quickly than manual mapping.

According to a study by Pandey et al. [25], implementing blockchain technology can effectively detect and prevent the distribution of counterfeit medications within medical networks. The solution's highly decentralized network has 11 machine nodes that link together. It is extremely secure, even with its demanding computational requirements. Intelligent mobile applications can access the technology for QR code scanning, but they cannot prevent using fake or illegal medications before purchase. However, society can contribute to the system by accepting blockchain-authorized payments. Other industries, such as election management and courier cargo tracing, could benefit from blockchain's resilience and ledger-event tracing capabilities.

3. System Model and Problem Statement

Penetration targets healthcare organization networks because they manage enormous volumes of sensitive patient data. Healthcare networks are still susceptible to changing cyber threats, even with various security mechanisms in place. It takes more than just conventional embedding technology to incorporate image data security, which is complicated. When considered, poor performance and lower payload capacity were achieved.

Usually, the embedding scheme is utilized for the digital application to secure the image data. But in many cases, the embedded algorithms are not sufficient to hide the data because of poor quality images and a lack of security features in embedding algorithms. Hence, it is not suitable for all image data and does not provide the expected best outcome. So, it is not secure against the malicious event, which is defined in Figure 1.

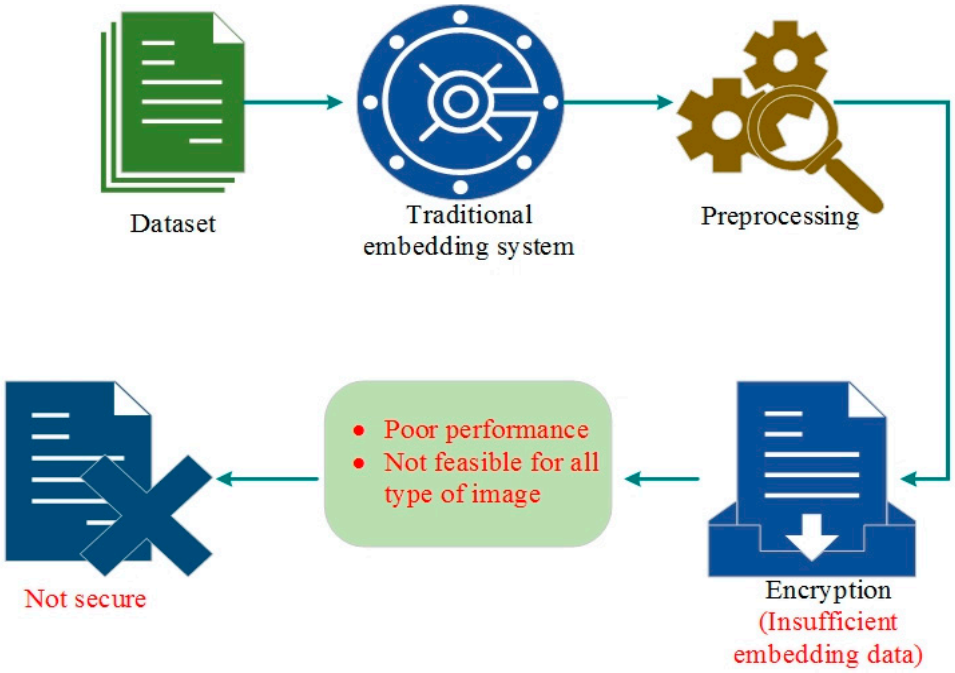


Figure 1. Difficulties with the traditional method.

4. Proposed GCbTS to Secure the Healthcare Network Data

This study introduces a new framework called the Graph Convolution-based Twofish Security (GCbTS) Framework to enhance the security of medical images in healthcare networks. The proposed framework is constructed utilizing suitable parameters and is subjected to testing using the collected data. The medical photos obtained from the Kaggle site were collected to evaluate the developed system. The structure of the suggested GCbTS approach is depicted in Figure 2 and Algorithm 1.

Algorithm 1 Graph Convolutional-Based TwoFish Security (GCbTS)

Start
{
$int\ I(D) = \{M_1, M_2, \ldots\ldots M_n\}$
//initialization of input digital video dataset
Preprocessing()
{
$int\ P,\delta,x,y$
//Preprocessing variables are initialized.
$P(D) \rightarrow \delta(x-y)$
//Noises are removed.

Algorithm 1 Cont.

Hash 1 calculation ()
{
int H_{v1},*p*
$hash_1 \rightarrow Mod(D, p)$
// The hash 1 value is estimated and stored in the framework.
$image \rightarrow frame$
//Input videos are split into the types of frames.
}
Encryption()
{
Int E_i^*,*n*,E^*
$E_i^* \rightarrow E^*(M_1, M_2, \ldots . M_n)$
//By twofish algorithm, the images are encrypted and assembled.
}
Hash 2 calculation()
{
$hash_2 \rightarrow Mod(E_n^*, p)$
// The hash 1 value is computed for encrypted data.
}
Hash validation()
{
if $(H_{v1} = H_{v2})$
{
//Verification is successful, and the system sends encrypted data along with key.
}
}
Decryption()
{
$N_d^* \rightarrow E_i^* / S_k$
//Encrypted data are retrieved into their original form.
}
}
Stop

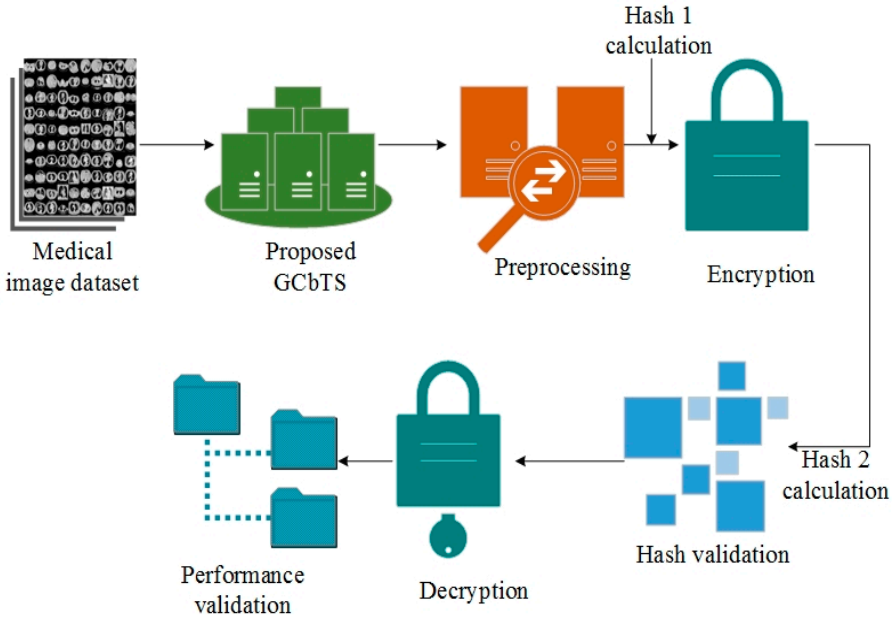


Figure 2. Proposed architecture.

The first step involves gathering the medical image dataset and calculating the hash 1s. After collecting the data, preprocess them to reduce the noise. The created key is then used to encrypt these preprocessed images throughout the encryption process. To confirm the user's identity, hash 2 is also calculated for the encrypted data and compared with hash 1. Users can access the data in their original forms after they have been verified and encrypted using 156 the shared key.

4.1. Process of GCbTS Model

The security framework comprises a graph convolutional neural network [26] and the twofish algorithm [27]. The medical image data are initially collected and included in the suggested framework for validation. The proposed approach is a graph-based convolutional neural network comprising the input layer, two concealed layers, and an output layer. The intended network's design is shown in Figure 3.

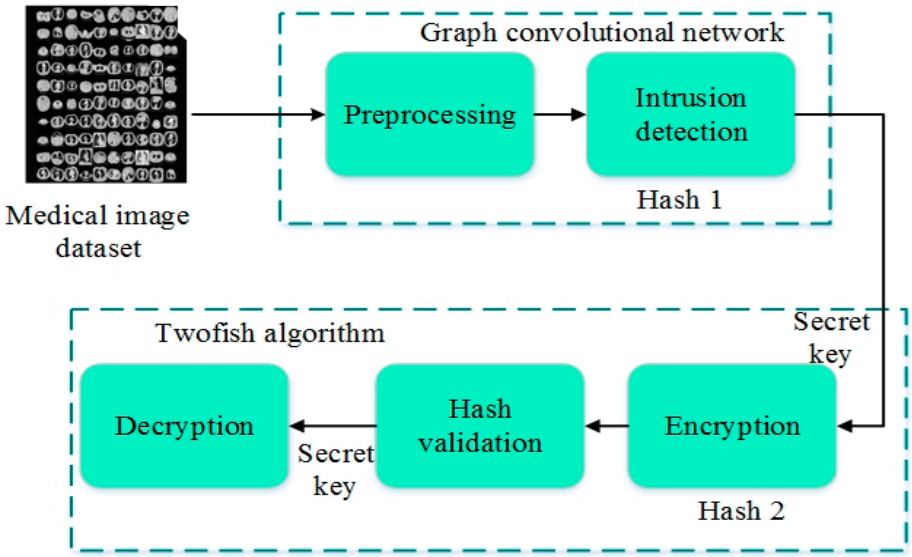


Figure 3. The process of the GCbTS.

The input layer incorporates the collected medical photos. At hidden layer 1, the hash calculation takes place after preprocessing the medical images. The neural network sequentially identifies and eliminates the assault images in the input image. Moreover, the twofish model handled the crypto mechanism. The proposed methodology is suitable for all distributed data environments by the graph convolutional neural network. To accept and process all formats of the image data type, the graph convolutional neural network was executed. It contains the three layers, which are the input layer, the dense layer, and the output layer. In the dense layer, the twofish encryption process was executed.

In addition, considering other encryption models, the twofish algorithm was designed and tested against different crypto analysis. So, it is more secure than other encryption models, and for this reason, the twofish algorithm is adopted for this research study. For this reason, the present proposed network has received a high confidential score.

4.1.1. Data Initialization and Preprocessing

The dataset has been initialized in the framework using graph convolutional neural features. Equation (1) describes the data initialization procedure:

$$I(D) = \{M_1, M_2, \ldots\ldots M_n\}, \quad (1)$$

where $I(D)$ represent the data initialization function, D represents the dataset, M represents the input medical image data, and n represents the images collected from the healthcare network. The noise in the images extracted from the dataset complicates the ongoing

process. Thus, the preprocessing stage is done to remove any noise from the dataset. Furthermore, the complexity has been lowered with the use of function preprocessing. As a result, the computing time is shortened, producing high results. Equation (2) removes the noise features from the input dataset:

$$P(D) = S + \delta(x, y) \times [x - y]. \tag{2}$$

Here, $P(D)$ represents the data preprocessing variable, δ denotes unwanted noise tracking parameters, x represents input data features, and y represents noise features present in the data. Thus, the preprocessing process removes the noise features.

Following this preprocessing function, the structure computed and stored the hash value for the input images. Equation (3) calculates the hash value. The hash 1 value is calculated by concatenating the randomly chosen prime number with the binary data on the input image [28]:

$$H_{v1} = S \ \ Mod\ p, \tag{3}$$

where H_{v1} represents the initial image hash 1 value, and p is denoted as a prime variable number. Then, the supplied datasets are entered into the encryption stage with their calculated hash values. The subsequent part provides an elaborate elucidation of the forthcoming procedures.

4.1.2. Encryption

Decoding and encryption are the two main steps in crypto analysis. In this case, the data are encrypted using the twofish algorithm. The initial generation of the key encrypts the data. In this case, the bit widths of the input data block's controlling variables are used to generate the key. The technique then uses the generated key to encrypt the images. Equation (4) provides the encryption procedure:

$$E_i^* = E^*(\sum_{i=1}^{I} Ms), \tag{4}$$

where E^* denotes the encrypted image, i represents the total images, and E^* represents the encrypting factor. Meanwhile, the encryption process changes the input medical images into a different format. Here, the image is considered as the data; primarily, the binary values of the images were extracted based on the most significant bit and least significant bit [29]. Then, for the binary data, an encryption process is performed to hide the information, as described in Equation (4). From this encrypted data, the following hash 2 was evaluated.

4.1.3. Hash Verification

Following the encryption of the data, two hash values are calculated using the encrypted data. The user is comparing the value of hash 1 with the projected hash 2 for the encrypted data. The determined hash values should correspond to the specified hash 1 value for the original material. The calculation of hash 2 is denoted by Equation (5):

$$H_{v2} = E_s^*(Mod \ \ p). \tag{5}$$

Here, H_{v2} denotes the 2nd hash function, and E^* represents the encoded dataset. To guarantee user confirmation, either hashing 2 and password 2 values are validated when the hash algorithm 2 computations are finished. Equation (6) confirms it.

$$Verify = \begin{cases} if(H_{v1} \neq H_{v2}) & Not\ verified \\ if(H_{v1} = H_{v2}) & user\ verified \end{cases}. \tag{6}$$

The system only gives access to the user when the sharing key matches the hash values. If this does not happen, the system will deny the user access. Here, hash 1 is valued by the

owner, who encrypts the message, and hash 2 is measured by the receiver, who needs to decrypt the message. Hence, if hash 1 and hash 2 are equal then the data are not injected, and in this way, they are used for verification.

4.1.4. Decryption

Decryption is the act of deciphering encrypted data. Upon completing the user validation process, the computer system will provide the user with the authentication code. Once the system confirms the user's authenticity, it decodes the encrypted videos to retrieve the original footage. This procedure is referred to as the decryption process. Equation (7) describes the decryption procedure:

$$N_d^* = E_i^* / S_k = D. \quad (7)$$

Here, N^* denotes the decrypted data, D represents the initial original data, and S_k denotes the secret key. The initial picture format was recreated by merging the decrypted images after the encrypted data were cracked utilizing an encryption key during decryption.

5. Result and Discussion

Graph Convolutional-Based Twofish Security (GCbTS) is suggested to protect pictures on a medical network. The medical images are gathered and arranged into datasets, then added to the recommended architecture. After the data are initialized, the video undergoes preprocessing to remove noisy features. The hash value of each image is subsequently calculated and stored. The twoish algorithm then combines and encrypts the photos using the provided key. The system authenticates the user by using hash 1 to calculate and validate the hash 2 values. The software delivers the encrypted video and the decryption key if the consumer's validation is successful. The execution variables are compiled in Table 1.

Table 1. Implementation parameters.

Implementation Parameters	
Parameter	**Description**
Programming platform	Python
Version	3.7.14
Operating system	Windows 10
Dataset	Medical MNIST
Data format	Image
Dataset size	6 GB
Processing power	20 mW
Cryptographic algorithm	Twofish algorithm

Furthermore, the results of the GCbTS architecture are evaluated. Moreover, an analysis of costs and benefits was done to assess the efficacy of the suggested approach.

5.1. Case Study

The unique GCbTS system is evaluated using the medical MNIST dataset downloaded from the official Kaggle website. The collection comprises a total of 58,954 pictures. This set of images, which is comparable to the MNIST dataset, is made up of 64 × 64 medical photos. The present-day arrangement of the photographs was created after they had been taken from different datasets.

This case study aimed to scrutinize the system's design process. The dataset first collects and incorporates the medical images into the recommended framework. Once the digital video datasets are gathered, each image is preprocessed to reduce noise and split into a certain number of images. The first step involves determining the hash value. The considered image data for this research investigation process and the encryption–

decryption are defined in Figure 4 and in this case study section. Here, the decryption is only eligible after the successful verification of hash 1 and hash 2. Figure 5 intimates the encryption time comparison. Thus, the message transaction was secure with high integrity.

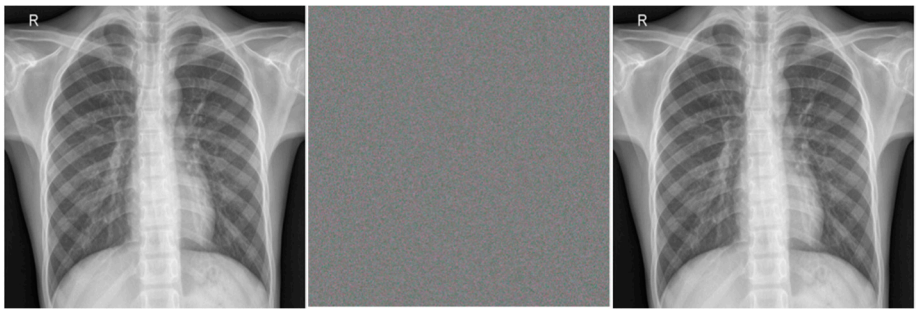


Figure 4. (**A**) input image, (**B**) encrypted image, and (**C**) decrypted image.

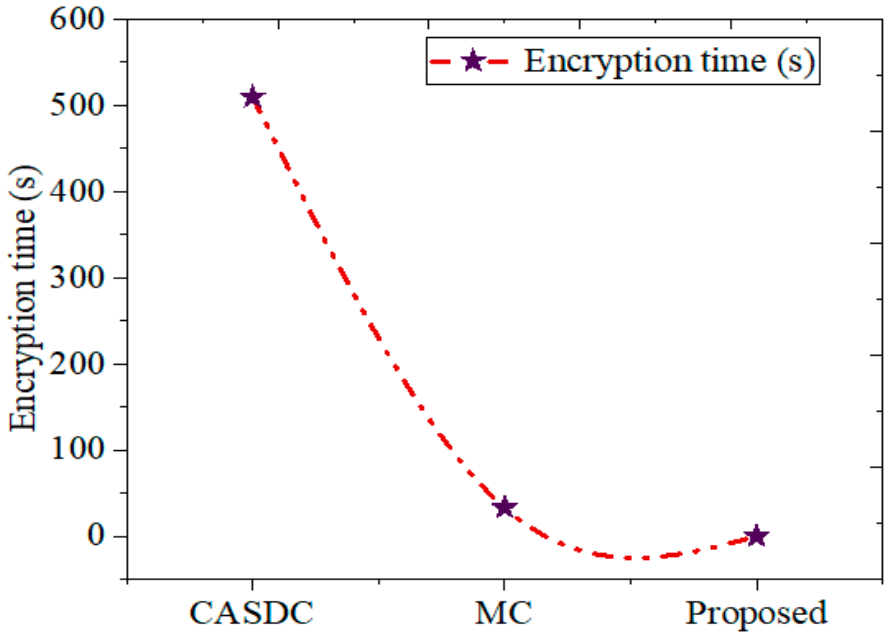


Figure 5. Encryption time comparison.

The previously produced key is then used to encrypt the preprocessed photos. To verify the user's identity, hash 2 was computed and contrasted with hash 1. After verification, the shared key decrypts the data and returns them to their original condition so users can access them. To determine the degree of improvement, we assess the outcomes generated through the GCbTS paradigm by comparing these with the results from earlier techniques. Furthermore, Figure 5 presents the data acquired from the encryption and decryption procedures.

5.2. Performance Analysis

The Python-based GCbTS framework was created on a Windows 10 system and subsequently compared to other methods to demonstrate its superior performance. These include a cryptographically secure data system (CASDC) [30], message cryptography (MC) [29], a quantum-inspired cuckoo search algorithm (QICSA) [31], a quantum-inspired tabu search algorithm (QITSA) [31], and a quantum-inspired genetic algorithm (QIGA) [31].

5.2.1. Encryption Time

"Encryption time" is the time needed to finish the encryption process for medical imaging data. It is commonly expressed in seconds. An algorithm is superior in efficiency if its encryption time is smaller.

The GCbTS framework's encrypting time has been compared to other widely used methods to verify its encryption speed. It is compared to other known approaches in Figure 5. The CASDC and MC methods achieved encryption times of 509.66 s and 33.40 s, respectively. The reduced encryption and decryption time verified the seed of the proposed system to test this security process against the malicious events.

5.2.2. Decryption Time

The decryption time refers to how long it takes to decode encrypted data back into their original image form. Cryptographic keys or algorithms reverse the data's encryption in this operation.

To validate the GCbTS framework's decryption time, the decryption time is compared to other successful approaches. Figure 6 compares it to other known techniques. The CASDC and MC methods achieved decryption times of 1802 s and 33.40 s, respectively.

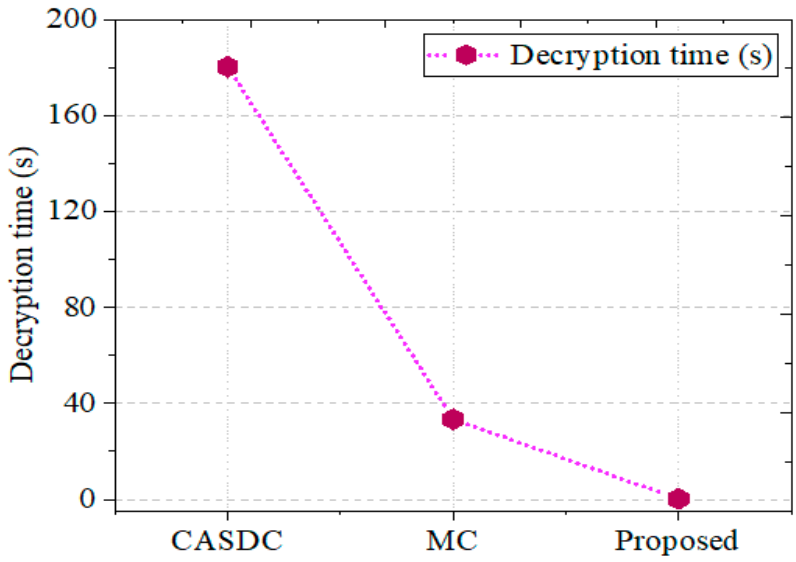


Figure 6. Decryption time comparison.

5.2.3. MSE

The mean squared error is a widely used metric for quantifying the mean square variance between the actual and anticipated values in a repression problem. It is a way to measure how accurate the model's forecasts are. Equation (8) expresses the formula for mean squared error (*MSE*):

$$MSE = \frac{1}{d}\sum_{i=1}^{d}(x_i - \hat{x}_i)^2, \quad (8)$$

where d denotes the number of taken data points, x_i denotes the actual values of points, and i denotes the compressed points. Gaining the lower *MSE* with other traditional models has revealed a better quality of image after the decryption process.

The GCbTS framework's mean squared error (*MSE*) is evaluated against several effective techniques for MSE validation. In Figure 7, it is compared with other well-established methods. High mean squared errors (MSEs) of 6.7064 and 7.6702 were obtained by the CASDC and MC techniques, respectively.

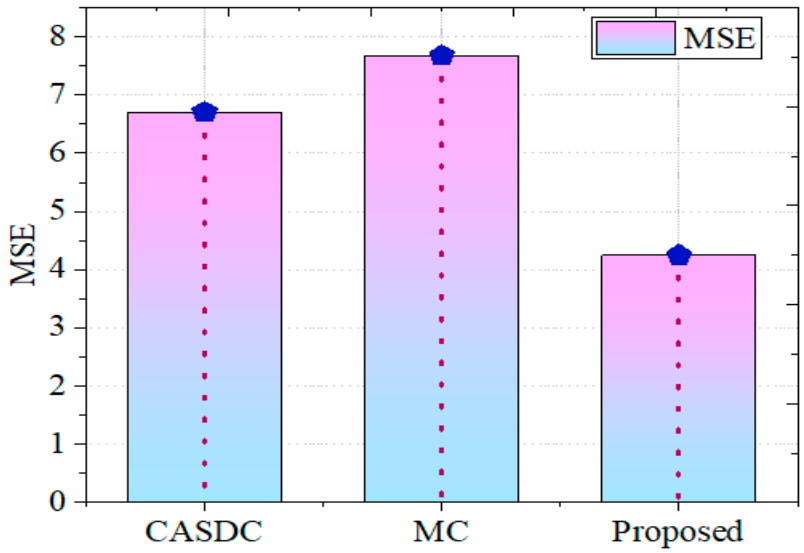


Figure 7. MSE comparison.

5.2.4. PSNR

People commonly utilize the peak signal-to-noise ratio (*PSNR*) metric when evaluating the accuracy of compressed or reconstructed images. *PSNR* is useful for assessing the caliber of cryptographic changes made to medical images when protecting healthcare networks from intrusion. Equation (9) provides the formula for calculating *PSNR*. Compared to the traditional models, the proposed model scores the highest PSNR as 24.7, which indicates that after the encryption and decryption process, the image quality was not degraded. Also, the proposed method is suitable for securing image data:

$$PSNR = 10.\log_{10}\left(\frac{255^2}{MSE}\right) \tag{9}$$

When comparing the *PSNR* of the GCbTS framework to other successful methods, the PSNR for the GCbTS framework is compared. It is contrasted with other prevailing techniques in Figure 8. The CASDC and MC methods achieved PSNR values of 22.7171 and 21.2168, respectively.

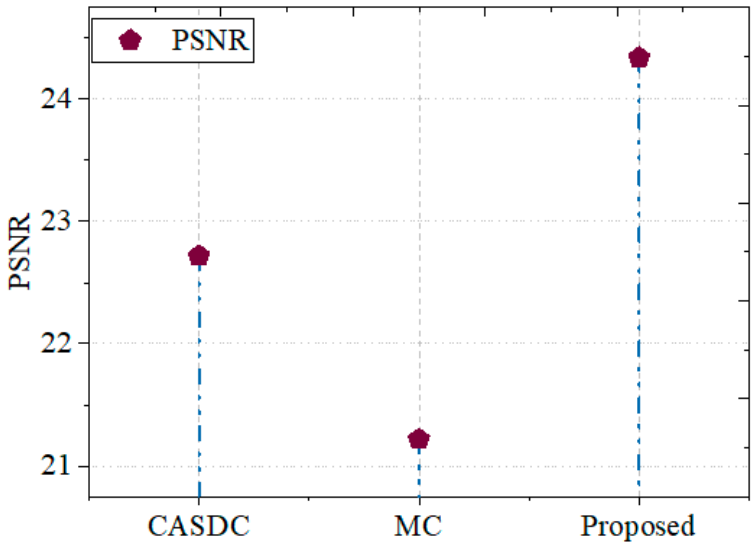


Figure 8. PSNR comparison.

5.2.5. Throughput

Throughput refers to a network's effective encryption, transmission, and decryption rate. It indicates the volume of data that a network can manage or transfer within a given time frame. The higher throughput defined that the present model is effective in the data transmission process.

The efficiency provided by the GCbTS framework is validated by contrasting it with the throughput of other effective techniques. Figure 9 compares it to other known approaches. The CASDC and MC methods achieved 32.37% and 52.79% throughput rates, respectively.

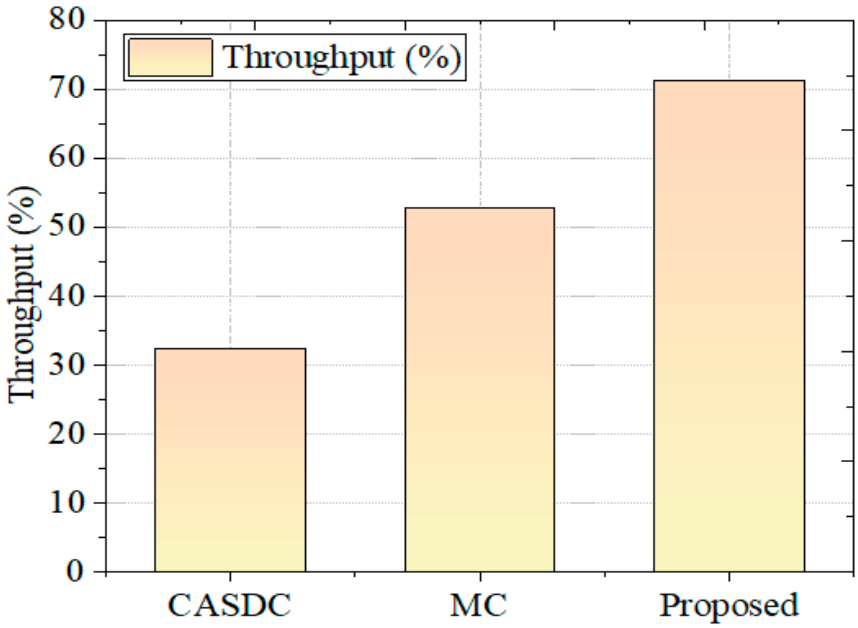


Figure 9. Throughput comparison.

The numerical results for all comparison data are given in Table 2. Here, all the traditional models were tested in the same proposed platform, which afforded an accurate outcome and made the comparative study more accurate. For the security proof, the confidentiality score is measured for all methods in Table 2.

Table 2. Overall comparison.

Method	Encryption Time	Decryption Time	Throughput	MSE	PSNR	Confidentiality Score (%)
CASDC	509.66 s	1802 s	3237.2	6.7064	22.7171	77%
MC	0.3340 s	0.3340 s	5279.7	7.6702	21.2168	72%
Proposed	0.225 s	0.231 s	71.225	4.245	24.721	98%

5.2.6. Error Rate

The error may be used to describe inconsistencies or errors that arise during cryptographic procedures. Various factors, including data corruption, incorrect encryption or decryption techniques, and deliberate attacks, may cause these problems.

The error rate of the framework is compared with other well-known techniques to validate the GCbTS framework's decrypting time. In Figure 10, it is compared with other widely used methods. Earlier techniques, such as QITSA, QICSA, and QIGA, attained error rates of 0.045%, 0.035%, and 0.024%, respectively. The overall proposed outcome is tabulated in Table 3. The proposed research is executed for the distributed environment, and it is not tested for quantum computers and quantum algorithms. To do the post-quantum cryptography in the present work, the database has to be converted to quantum

bits, and then the post-quantum crypto algorithm needs to be implemented to secure the generated quantum bits.

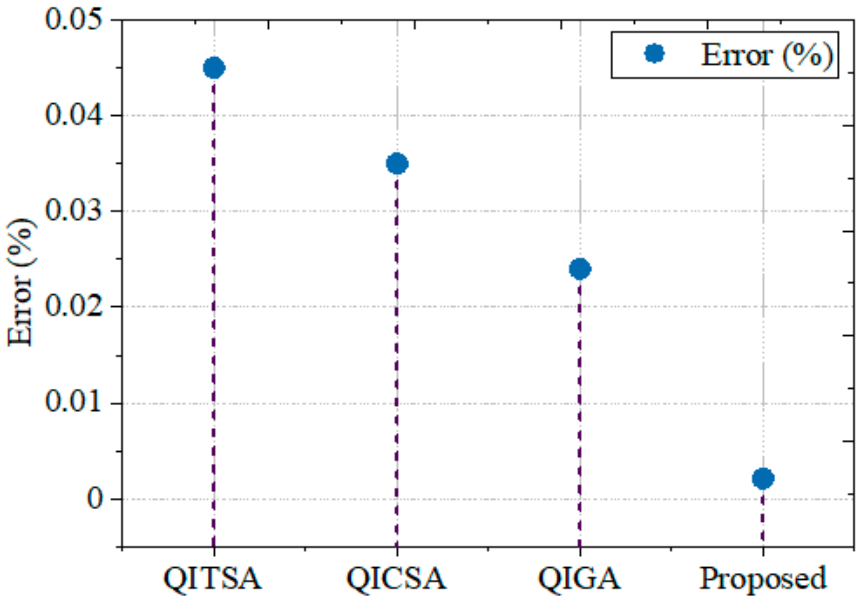


Figure 10. Error rate comparison.

Table 3. Performance statistics of GCbTS.

Metrics	Performance
Encryption time	0.225 s
Decryption time	0.231 s
MSE	4.245
PSNR	24.721
Confidentiality score	98%
Throughput	71.225%
Error	0.0021%

6. Conclusions

Securing healthcare data is more important in digital smart work; hence, the present work introduced a novel GCbTS as a security mechanism to protect medical data against third parties. Here, the owner and the third parties were identified by performing the hash 1 and 2 verification. Furthermore, the proposed model achieved 24.721 PSNR, 71.225% throughput, and a reduced average error rate of 0.002%. In addition, the average cryptographic processing time for both encryption and decryption is 0.2 s. By verifying all those outcomes, the presented GCbTS has improved the security performance by 20%. Hence, the proposed model is more apt for this specific domain to secure the digital sensible data for healthcare and other applications. However, this current research work does not address the memory space requirement or the need for energy resources.

In the future, incorporating the memory computing model and energy needs estimation along with its optimization procedure will provide the most precise reliability outcome. The key demerit of this study is that it has not been tested with quantum attacks. In the future, introducing post-quantum algorithms and testing them with quantum attacks will produce a better outcome.

Author Contributions: Conceptualization, A.S. and V.S.S.; methodology, A.S.; software, V.S.S.; validation, S.B. and V.S.S.; formal analysis, V.S.S.; investigation, C.L.C.; resources, A.S.; data curation, V.S.S.; writing—original draft preparation, A.S.; writing—review and editing, V.S.S. and S.B.; visualization, S.B.; supervision, C.L.C.; project administration, C.L.C.; funding acquisition, C.L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Thankful to VIT Vellore for providing the research environment and the resources.

Informed Consent Statement: Not applicable.

Data Availability Statement: Kaggle.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Oosthoek, K.; Doerr, C. Cyber Security Threats to Bitcoin Exchanges: Adversary Exploitation and Laundering Techniques. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1616–1628. [CrossRef]
2. Zhang, H.; Wang, J.; Wang, Y.; Li, M.; Song, J.; Liu, Z. ICVTest: A Practical Black-Box Penetration Testing Framework for Evaluating Cybersecurity of Intelligent Connected Vehicles. *Appl. Sci.* **2023**, *14*, 204. [CrossRef]
3. Tamburri, D.A. Design principles for the General Data Protection Regulation (GDPR): A formal concept analysis and its evaluation. *Inf. Syst.* **2020**, *91*, 101469. [CrossRef]
4. Tudosi, A.-D.; Graur, A.; Balan, D.G.; Potorac, A.D. Research on Security Weakness Using Penetration Testing in a Distributed Firewall. *Sensors* **2023**, *23*, 2683. [CrossRef] [PubMed]
5. Bella, G.; Biondi, P.; Bognanni, S.; Esposito, S. PETIoT: PEnetration Testing the Internet of Things. *Internet Things* **2023**, *22*, 100707. [CrossRef]
6. Altulaihan, E.A.; Alismail, A.; Frikha, M. A Survey on Web Application Penetration Testing. *Electronics* **2023**, *12*, 1229. [CrossRef]
7. Cirne, A.; Sousa, P.R.; Resende, J.S.; Antunes, L. IoT security certifications: Challenges and potential approaches. *Comput. Secur.* **2022**, *116*, 102669. [CrossRef]
8. Iqbal, F.; Debbabi, M.; Fung, B.C.M.; Iqbal, F.; Debbabi, M.; Fung, B.C.M. Cybersecurity and Cybercrime Investigation. In *Machine Learning for Authorship Attribution and Cyber Forensics*; Springer: Cham, Switzerland, 2020; pp. 1–21. [CrossRef]
9. Filiol, E.; Mercaldo, F.; Santone, A. A Method for Automatic Penetration Testing and Mitigation: A Red Hat Approach. *Procedia Comput. Sci.* **2021**, *192*, 2039–2046. [CrossRef]
10. Alhamed, M.; Rahman, M.M.H. A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. *Appl. Sci.* **2023**, *13*, 6986. [CrossRef]
11. Aslan, Ö.; Aktuğ, S.S.; Ozkan-Okay, M.; Yilmaz, A.A.; Akin, E. A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics* **2023**, *12*, 1333. [CrossRef]
12. Yaacoub, J.-P.A.; Noura, H.N.; Salman, O.; Chehab, A. Ethical hacking for IoT: Security issues, challenges, solutions and recommendations. *Internet Things Cyber-Physical Syst.* **2023**, *3*, 280–308. [CrossRef]
13. Rak, M.; Salzillo, G.; Granata, D. ESSecA: An automated expert system for threat modelling and penetration testing for IoT ecosystems. *Comput. Electr. Eng.* **2022**, *99*, 107721. [CrossRef]
14. Zhang, J.; Pan, L.; Han, Q.-L.; Chen, C.; Wen, S.; Xiang, Y. Deep Learning Based Attack Detection for Cyber-Physical System Cybersecurity: A Survey. *IEEE/CAA J. Autom. Sin.* **2021**, *9*, 377–391. [CrossRef]
15. Saheed, Y.K.; Abiodun, A.I.; Misra, S.; Holone, M.K.; Colomo-Palacios, R. A machine learning-based intrusion detection for detecting internet of things network attacks. *Alex. Eng. J.* **2022**, *61*, 9395–9409. [CrossRef]
16. Riggs, H.; Tufail, S.; Parvez, I.; Tariq, M.; Khan, M.A.; Amir, A.; Vuda, K.V.; Sarwat, A.I. Impact, Vulnerabilities, and Mitigation Strategies for Cyber-Secure Critical Infrastructure. *Sensors* **2023**, *23*, 4060. [CrossRef]
17. Kure, H.I.; Islam, S.; Mouratidis, H. An integrated cyber security risk management framework and risk predication for the critical infrastructure protection. *Neural Comput. Appl.* **2022**, *34*, 15241–15271. [CrossRef]
18. Cao, S.; Sun, X.; Bo, L.; Wei, Y.; Li, B. BGNN4VD: Constructing Bidirectional Graph Neural-Network for Vulnerability Detection. *Inf. Softw. Technol.* **2021**, *136*, 106576. [CrossRef]
19. Wartschinski, L.; Noller, Y.; Vogel, T.; Kehrer, T.; Grunske, L. VUDENC: Vulnerability Detection with Deep Learning on a Natural Codebase for Python. *Inf. Softw. Technol.* **2022**, *144*, 106809. [CrossRef]
20. Prabakaran, S.; Ramar, R.; Hussain, I.; Kavin, B.P.; Alshamrani, S.S.; AlGhamdi, A.S.; Alshehri, A. Predicting Attack Pattern via Machine Learning by Exploiting Stateful Firewall as Virtual Network Function in an SDN Network. *Sensors* **2022**, *22*, 709. [CrossRef]
21. Ghanem, M.C.; Chen, T.M.; Nepomuceno, E.G. Hierarchical reinforcement learning for efficient and effective automated penetration testing of large networks. *J. Intell. Inf. Syst.* **2022**, *60*, 281–303. [CrossRef]
22. Preetha, A.D.; Kumar, T.S.P. Securing IoT-based healthcare systems from counterfeit medicine penetration using Blockchain. *Appl. Nanosci.* **2021**, *13*, 1263–1275. [CrossRef]
23. Alabbad, M.; Mhaskar, N.; Khedri, R. Hardening of network segmentation using automated referential penetration testing. *J. Netw. Comput. Appl.* **2024**, *224*, 103851. [CrossRef]
24. Hidayanto, B.C.; Akbar, I.A.; Putra, A.Z.D. Automated Web Security Testing Guide Mapping to Accelerate Process on Penetration Testing. *Procedia Comput. Sci.* **2024**, *234*, 1412–1419. [CrossRef]
25. Manessi, F.; Rozza, A.; Manzo, M. Dynamic graph convolutional networks. *Pattern Recognit.* **2019**, *97*, 107000. [CrossRef]

26. Haq, T.U.; Shah, T.; Siddiqui, G.F.; Iqbal, M.Z.; Hameed, I.A.; Jamil, H. Improved Twofish Algorithm: A Digital Image Enciphering Application. *IEEE Access* **2021**, *9*, 76518–76530. [CrossRef]
27. Abu-Faraj, M.; Al-Hyari, A.; Al-Taharwa, I.; Al-Ahmad, B.; Alqadi, Z. CASDC: A Cryptographically Secure Data System Based on Two Private Key Images. *IEEE Access* **2022**, *10*, 126304–126314. [CrossRef]
28. Bhullar, R.K.; Pawar, L.; Kumar, V.; Anjali. A novel prime numbers based hashing technique for minimizing collisions. In Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 14–16 October 2016. [CrossRef]
29. Puteaux, P.; Puech, W. An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1670–1681. [CrossRef]
30. Abu-Faraj, M.; Al-Hyari, A.; Aldebei, K.; Alqadi, Z.A.; Al-Ahmad, B. Rotation Left Digits to Enhance the Security Level of Message Blocks Cryptography. *IEEE Access* **2022**, *10*, 69388–69397. [CrossRef]
31. Alhayani, B.A.; AlKawak, O.A.; Mahajan, H.B.; Ilhan, H.; Qasem, R.M. Design of Quantum Communication Protocols in Quantum Cryptography. *Wirel. Pers. Commun.* **2023**, 1–18. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.