Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
DOI : 10.47974/JIOS-1837

Enhancing information security and efficiency through integration of neural networks with RSA Cryptography

Abhishek *
Ruchi Sehrawat †
University School of Information, Communication & Technology
Guru Gobind Singh Indraprastha University
Dwarka 110078
New Delhi
India

Abstract

The cryptographic system aims to facilitate the secure exchange of information among the intended users and also avoid unauthorized access. It is critical task to generate cryptographic keys to ensure the security of encrypted information. The widely adopted RSA algorithm relies on prime numbers for public key cryptography. Aim of this paper is to augment the existing RSA key generation process by incorporating neural network model. In recent years use of neural networks has shown great potential to improve the effectiveness and security of cryptographic systems. This paper is based on deep learning based neural networks model, which are trained on large amount of data to learn patterns to optimize and improve the level of unpredictability in the key generation process to enhance security of the RSA. This work also conduct a comprehensive analysis of the key space, entropy and performance evaluation to validate the effectiveness in generating secure keys.

Subject Classification: *68M25, 68P25, 68P27, 94A60.*

Keywords: *Cryptography, Deep neural networks (DNN), Key generation, RSA.*

1. Introduction

In the immersive growth of digital communication, cryptography holds a substantial role in the area of electronic communication through networks to guarantee the confidentiality and trustworthiness of sensitive data, new techniques have been developed to increase the reliability of

* *E-mail:* **abhisheksaini348@gmail.com (Corresponding Author)**
† *E-mail:* **ruchi.sehrawat@ipu.ac.in**

encryption tasks and reduce the dependency of in-person interaction to exchange or update the encryption and decryption keys [31,32]. Encryption is used as a vital tool to protect confidential information from unauthorized entity, having the incorporation of cryptographic protocols. These protocols are sets of rules and procedures which are designed to ensure secure communication and data exchange. Some cryptographic protocols are SSL/TLS, PGP, Kerberos, etc. with the objective of key exchange, entity authentication and more [19, 24, 27].

Two primary types of cryptography exist: symmetric key cryptography and asymmetric key cryptography. In symmetric key cryptography single key is utilized for both the encryption and decryption of message. AES and DES algorithms are examples of symmetric key algorithms [9, 24]. In the realm asymmetric key cryptography, distinct key pairs are serves the function of encryption and decryption. The process of encryption relies on a public key, while decryption require a private key. Asymmetric key algorithm includes RSA and ECC algorithms [9, 21, 24].

RSA cryptography is the most popular asymmetric key cryptography algorithm that provides secure communication over networks. The security of the RSA cryptography algorithm relies on the strength of the key, if the key is not strong enough then it is susceptible to attack. To tackle this issue, many strategies have been suggested [3] to optimize key generation in cryptography.

Prime numbers are an essential component of RSA cryptography, which plays important role in the security mechanism of the RSA algorithm. Currently, there are no efficient methods for producing extremely large prime numbers. Generally, an odd number is chosen and the primality is tested. If the number is not prime another odd random number is chosen until a prime number has been identified [7, 24]. Numerous primality testing algorithms have been developed [2, 5]. The Miller-Rabin algorithm [22] is a well-known algorithm for primality testing of large prime numbers. Testing for primality is almost probabilistic, it only shows that a given number is probably prime or not [24, 29]. This paper is integrated with machine learning algorithm to predict the prime numbers using deep neural networks to boost the security of RSA Cryptography while generating keys. DNN model has been trained to learn the characteristics of prime numbers. This model is proficient of learning the intricate patterns and connections present in the data. The deep neural networks can optimize the key generation and improve the randomness of key generation, to make it more challenging for attackers to break the key.

Though cryptography is not a definite way to secure sensitive information, still many vulnerabilities can be exploited by cybercriminals. Therefore, it is important to improve and update the cryptographic system to stay ahead of the threats.

2. Background

2.1 *RSA*

In 1977 Ronald Rivest, Adi Shamir and Leonard Adleman designed the public key cryptography algorithm at MIT known as RSA (Rivest Shamir Adleman) [26]. It is a very popular classical algorithm, which plays significant role in transmission of information in almost every field: digital signature, online shopping, banking, e-commerce, etc.

Security of the RSA algorithm is dependent on the factorization of product of two large prime numbers, there product can be 2048 bit to 4096 bit long, increase in bit length will increase the level of the security required. Larger key size can slow the computational performance but increases the security. Thus, the strength of the RSA algorithm rely on the difficulty presented by factoring the modules of the public key. The RSA algorithm uses a key pair, private key and the public key. On the Receiver side receivers generate the key pair of the RSA algorithm and release their public key on the network, on sender side, sender can encrypt the message or information using receiver public key and send it to the receivers and the receiver decrypt the message using own private key.

In RSA algorithm two big prime numbers, *pr1* and *pr2* are chosen and both are multiplied with each other to produce a product module, $n = pr1 \times pr2$, and generate the Euler's totient function $\varnothing(n) = (pr1-1) \times (pr2-1)$. This module is utilized for generate the public and private keys. Next, a number '*e*' is selected that is relatively prime to $\varnothing(n)$ [24], selected value of '*e*' should satisfy the condition, $1 < e < \varnothing(n) \in gcd(\varnothing(n), e) = 1$, and thereafter, compute the inverse module '*d*', $d = e^{-1} \bmod \varnothing(n)$. Finally, the RSA keys are, *(n, e)* for public key and *(n, d)* for private key.

To encrypt the message, public key module is used, encrypted message is known as cipher text that can be calculated as, $c = m^e \bmod n$, and the resulting cipher text can only be decrypted using the private key i.e. *original message* = c^d *mod n.*

Therefore, RSA algorithm is generally considered a reliable and secure algorithm for encryption [21].

2.2 *Neural Networks*

Neural networks are a design system that imitate how the brain executes a specific task of interest. Artificial neural networks, often referred as neural networks, are constructed with interconnected artificial neurons analogous to biological neural networks [18].

Neural networks are structured into layers, the initial layer is the input layer, and final layer is an output layer and one or more intermediary layers known as hidden layers between input and output layers. Each layer consists of multiple neurons, each neuron takes input signals, applies an activation function to its weighted inputs, performs computation on them, and produces an output signal.

Neural networks have proven their effectiveness across wide range of fields, such as image and speech recognition, medical diagnosis, cryptography and more [25]. The network adapts knowledge from its surroundings via the learning process. The realm of neural network is a continuously landscape of research and development, contributing to advancements in artificial intelligence. With the ability to learn from data and provides predictions, neural networks have become powerful tool for solving complex problems of various fields.

2.3 *Deep Neural Networks*

A type of artificial neural network referred as deep neural network considered as deep learning model that consists of multiple hidden layers between the input and output layer [13, 19] can be denoted by eq. (1).

$$f_x = f_{x0} \ldots\ldots\ldots\ldots\ldots\ldots f_{x-1} \tag{1}$$

Deep learning models have the properties of representation and extraction of complex patterns and features from input data to the depth of the networks [15, 25].

As the information flow deeper into the network, these hidden layers allow the network to acquire a hierarchical depiction of the input data, where each layer captures increasingly abstract features, generating an efficacious model for complex pattern problems/applications viz. natural language processing, security etc. DNNs are highly versatile to design generalized models capable of handling large and complex dataset [25].

To train deep neural networks, backpropagation technique is applied on labelled input data. The network adjust the weights and bias of the neurons based on the error between projected output and preferred

output. The optimization algorithms allow the network to optimize its internal representation and improve its performance continuously.

2.4 *Neural Cryptography*

Neural networks have the ability to learn new skills, it can be trained to safeguard the confidentiality of the information. Neural networks are proficient of learning to maintain the information privacy, determining the segment of data to encrypt and also discovering new techniques for cryptosystem and increase the data security.

There are several kinds of cryptographic algorithms with more sophisticated methods and processing power. These algorithms are vulnerable to numerous attacks such as brute force and man-in-middle attacks which aims to find the key that can be used to get the original message. By observing the behavior of cipher operation with multiple different keys attacker can know the mathematical connection between keys. RSA algorithm with 129 and 193 digits is also susceptible to attacks [24]. Thus, neural networks are applied in cryptography to address these issues, neural networks and cryptography improve the network security when both are combined. Neural networks techniques used to create encryption and decryption keys, key is represented by the neural network's weights that are more complex to break [10, 17].

3. Literature Survey

Jagadeesan and Duraiswamy [14] introduced a combined approach that integrated iris and fingerprint biometrics to generate a secure encryption key. The approach consists of three primary stages: extracting features, creating multimodal biometric template and generating cryptographic key. In the feature extraction stage, fabric characteristics such as points and features were extracted from fingerprints and iris images. Extracted features were combined at the feature level to develop a multi-biometric prototype. At last 256-bit encryption key was derived from the biometric data.

Gaddam and Lal [8] presented an innovative approach by generating cryptographic keys and secured features matrix to store fingerprint templates. The author overcome the drawback of traditional methods that depend on fixed biometric features by introducing cancellable keys derived from fingerprints, the inclusion of biometric characteristics

improved the reliability of encryption which showed the valuable solution of this issue.

Jogdand and Bisalapur [16] applied neural networks approach to create secret keys over the network channel for both encryption and decryption. Input vectors are provided to both networks, resulting in the produce an output bit and training is conducted based on this bit. Through the utilization of vector weights and the synchronization of both networks, a unique phenomenon emerged, in the establishment of a key that is employed for encryption and decryption purpose over the public channel.

A method was suggested by Dongjiang, Yandan, and Hong [6] in their work to minimize the time and space complexity while generating the RSA key pair. The authors produced the random prime number from the generator algorithm that take random 'n' values which further increased by 2, if the number is found prime, then that number passed to prime test, authors used the Miller-Rabin algorithm for the prime test. Later to generate the RSA key pair, Stain algorithm was used.

Nagar and Alshamma [21] introduced an offline approach to generate the RSA algorithm's keys using an offline database, and proposed a level-based security system for the encryption and decryption process, which have four levels such as low, medium, medium-high and high each level has own database with different key size that is 512-bits, 1024-bits, 2048-bits and 4096-bits. The database consists of numerous numbers of prime modules and corresponding key modules, each module has its index id such as 'Eid, Did and Nid'. These index id shared for both encryption and decryption instead of original modules of keys. For encryption and decryption on both networks, RSA Handshake Database Protocol was created that used an offline database for RSA keys generation, by selecting the security level. The proposed protocol also insured the security at each level and update key length and database.

Segar and Vijayaragavan [1] created a modified version of the RSA algorithm that introduced different ways of key generation compared to the RSA algorithm. In their research authors used various techniques, including the existing RSA algorithm, N prime RSA algorithm and Dual RSA algorithm to develope Pell's RSA keys. The authors also proposed Pell's RSA application in blind signatures.

Hussain and Al-Bahadili [12] proposed an approach based on DNA for derived secret key for symmetric cryptographic-based applications. A Key-Based Random algorithm used a private key as input to generate 4 vectors (A, C, G and T) to create a permutation of n size. These DNA

vectors were processed to develop cryptographic keys. This procedure repeated to generate multiple keys as needed.

Gayathri et al. [11] suggested a method to generate the private key using AES and ECC algorithms. The authors enhance the security level to increase the key length from 128 bits to 196 bits with the corresponding increase of rounds from 10 rounds to 12 rounds.

Shukur [23] presented an image-based key generation algorithm in his paper, where frequencies of red, blue and green colours were compute and the maximum frequency multiplied with each colour frequency. To generate the key, all multiplied frequency value were added. Once the key was generated, it converted into the binary. Finally, this secret key was used to cover the text using the XOR operation.

Jin and Kim, [15] proposed a three-dimensional cube algorithm where both networks generate the same secret key using a neural networks model to shuffle and solve the 3D cube without leakage or sharing a pre-shared key.

4. Proposed Work

This research paper is focused on exploring the latent applications of deep neural networks in cryptography, with an emphasis on key generation, and also to classify, how neural network helps to optimize cryptographic systems.

In the proposed work Deep Neural Networks are used to optimize the key generation process of the RSA cryptography algorithm and to construct the RSA keys that are more secure and less susceptible to attacks. The DNN model is trained on a substantial volume of data to acquire the capability to generate robust keys. DNN model is used to predict that a given prime number is truly a prime, which can help to speed up the process of generating the private and public keys of the RSA algorithm.

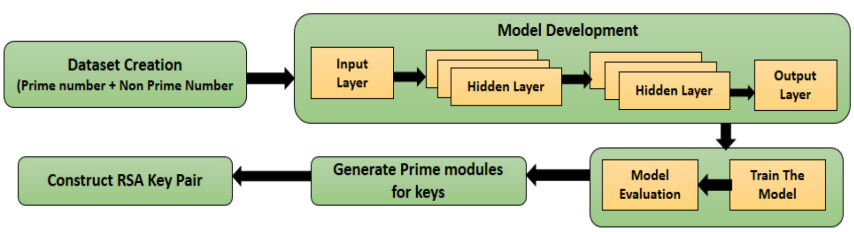

Figure 1

Sequence of proposed work

As shown in Figure 1, the proposed methodology for key generation using a deep neural network contains the following steps:

4.1 *Construct Dataset*

A huge dataset of natural numbers is generated using two function F_p and F_{np} that generates prime and non-prime numbers within a specified range *[2, n]* using eq. (2) & (3). Function F_p *(n)* generates a list *P* of prime number, and function F_{np} *(n)* generates a list *NP* of non-prime numbers. These list used construct dataset *D,* by using eq. (4)

$$P = \{p \mid p \in \{2,3,\ldots,n\}, \forall i \in \{2,3,\ldots..p-1\}, p \bmod i \neq 0\} \tag{2}$$

$$NP = \{np \mid np \in \{2,3,\ldots,n\}, \exists i \in \{2,3,\ldots..np-1\}, np \bmod i = 0\} \tag{3}$$

$$D = P \cup NP \tag{4}$$

then,

$$D = \{2,3,\ldots.,n\} \tag{5}$$

The dataset is split into training and testing sets of 80:20 ratio.

4.2 *Model Development*

Using the TensorFlow library a DNN model is developed, DNN model includes multiple dense layers with varying numbers of nodes and dropout layers to prevent overfitting [30]. The ReLU activation eq. (6) is used in hidden layers and sigmoid activation eq. (7) function is used in the output layer to predict that a given number is a prime number.

ReLU

$$R(z) = max(0, z_i) \tag{6}$$

Sigmoid

$$S(z) = \frac{1}{1+e_i^{-z}} \tag{7}$$

To compile the model Adam optimizer with a binary cross-entropy loss function and accuracy as the evaluation metric is used.

4.3 *Model Training*

Developed DNN model is trained on the training dataset, the training process involves 5 epochs with a distinct batch size. In each epoch weight

matrix w_i is updated by Xavier uniform distribution function of weight initialization i.e. *[-r, r]* where,

$$r = \sqrt{\frac{6}{f_{in} + f_{out}}} \tag{8}$$

Where f_{in} is number of input coming to node and f_{out} is number of output produced from node.

And output signal y_i of hidden and output layer obtained by using eq. (9) & (10).

$$z_i = \sum_{i=1}^{m} w_i x_i \tag{9}$$

and

$$y_i = \varphi(z_i + b_i) \tag{10}$$

let

$$v_i = z_i + b_i \tag{11}$$

Then y_i can be formulated as,

$$y_i = \varphi(v_i) \tag{12}$$

where w_i is weigh, $x_i$is input, b_i is bias, m is the number of neurons in respective layer, and φ is activation function.

4.4 *Model Evaluation*

Trained DNN model is evaluated using the testing dataset. In the evaluation process, the loss function and the accuracy of the model on the test dataset are evaluated.

4.5 *Generate Key Module*

After training and evaluating the DNN model, it is used for prime number generation for keys. A random prime number is generated within a specified range like 32-bit or more as user will define, and the model is used to predict whether the number is prime or non-prime. If the predicted number is prime, it is used to create keys of the RSA algorithm. This part involves selecting two large prime numbers ‘p’ and ‘q’.

4.6 *RSA Key pair Creation*

Using the generated prime numbers ‘p’ and ‘q’, the value of ‘n’ and ‘Phi’ are calculated. Using these modules, the public key exponent ‘e’ and

the private key exponent 'd' are calculated. And finally using the module 'n', 'e' and 'd', RSA key pair is constructed.

Let's explain with an example, assume the input number 61. Model after training predict the probability of it being a prime number as 0.92. It suggests a high likelihood of number 61 being prime. The loss function associated with this prediction is 0.08, i.e. relatively very small error.

This proposed approach comprises the construction of a dataset, involves deep learning based neural network model to optimize the process of generating prime numbers to build RSA key pair. This extensive method combined machine learning with cryptography to address prime numbers related challenges and this proposed work has the potential to speed up the key generation process while maintaining a high level of security.

5. Results and Discussions

This section presents an experimental analysis of the proposed work through various tests.

5.1 *Performance Analysis*

In this paper deep neural network is used to optimize the process of generating the key pair of the RSA cryptographic algorithm by predicting that a given number is prime. First, a list (dataset) of prime and non-prime numbers is generated, which have 2, 00,000 of natural numbers and trains a DNN model on the 80% of the dataset to predict the prime number, and assigned a label of 1 to prime numbers and 0 to the non-prime numbers, with the Adam optimizer and a binary cross-entropy loss function for model compilation, and evaluated on a 20% of the dataset. The DNN model is created with an initial layer i.e. input layer, followed by three concealed layers and concluding with an output layer. As shown in Figure 2, after undergoing 5 epochs of training, the model attained a training accuracy of 91.01% and a test accuracy of 91%. The high test accuracy indicates that the model exhibits strong generalization ability to new data and no overfitting. From Figure 3, the test loss is 0.30, that indicates that the model show relatively low errors when predicting the prime numbers from the test dataset. It is essential to understand that the loss metric is sensitive to the dataset's size and complexity of the problem, and thus should be interpreted carefully [28].

Finally, the trained model generates components of RSA key pair, this process involves generation of random numbers and predict prime number using the model. If the number is identified as prime, it will be returned as a key module. The proposed approach is capable of generating huge prime numbers such as 199999, but smaller prime integer are used for key generation to comply with the system's computing power limits. Table 1, 18-bit prime numbers are used to create keys, which produced 36-bit long private key module i.e. 43420861313.This proposed approach determines how a deep neural network can be used to improve the optimization of key generation. It can help to generate keys that are more secure and less susceptible to attacks. Utilizing deep neural networks can aid to speed up the key generation process, which is important in secure communication via the network.

5.2 *Keyspace Analysis*

Keyspace analysis is a technique used in cryptography to measure the strength and security of a cryptography algorithm based on the size of its key. The key space analysis represents the total enumeration of possible keys that can be generated for a cryptographic algorithm. When designing a cryptographic system, it's important to choose an appropriate key size, a larger key space provides great security by increasing the number of possible keys, an attacker would need to try before finding the correct key. For example, if a cryptographic algorithm uses a 32-bit key, then there are 2^{32} combinations of possible keys, which is an enormous number. The offered model employed with RSA algorithm, where the public key used two modules 'n' and 'e'. Module 'n' is an essential module of the algorithm that used up to 36-bit long digit number in the implementation of the model. The size of the keyspace would be 2^{36} which is larger i.e. 68.72 billion possible outcomes and can be increased when the size of the constructed dataset will increase. As a result, produced key module has a comparatively wide key space, providing a good level of security against brute-force attacks.

5.3 *Entropy Analysis*

The entropy test measures the amount of randomness in the generated keys. A higher value of entropy indicates higher randomness and better security of the key. A high entropy key has a vast number of possible combinations, making it almost difficult to discover through brute-force attacks. The following equation is used to analyze entropy

$$H(x) = -\sum_{i=1}^{N} p(x_i) log_2 p(x_i) \quad (13)$$

Where x_i is a random value with N outcome, p is the probability mass function. In our case the entropy value of the 48-bit public key is 4.32 and for the 64-bit private key is 4.35, which is showing that the suggested key generation model is capable of providing confrontation to brute force or entropy attacks.

Security of RSA cryptography depends on a wide range of factors [4, 20]. Other causes, such as key management, storage and distribution, must also be carefully considered to ensure that the keys are secure. Furthermore, the proposed approach shows promising results.

Table 1 shows the randomly generated prime module produced by the model and the respective value of key module n and d for the value of $e = 65537$.

Table 1

Key Modules

Bit length (n)	p	q	n	d
8	13	11	143	113
16	137	251	34387	13473
32	48593	58661	2850513973	430016193
36	184949	241469	44659450081	43420861313

The study shows that employing neural networks for key generation is highly effective. Developed DNN model confirmed noticeable accuracy on the test dataset, which is capable of generating prime numbers for key pair of the RSA cryptography algorithm. Using the model, the key generation process is substantially faster, and achieved a greater degree of randomness, thereby enhancing its security against potential breaches.

The proposed method of generating RSA key pair using deep neural networks has several advantages over the traditional method:

- **Faster Key Generation**

For large sizes of keys, the traditional method of RSA key generation is computationally intensive and time-consuming. In comparison, the proposed method using deep neural networks is significantly faster, and the model can generate keys with much greater speed.

- **Improved Key Randomness**

Randomness is essential for the security of the RSA algorithm's key pair. Traditional methods rely on pseudo-random number generators, which can have weaknesses that make them vulnerable to attacks [18, 20]. The proposed neural network-based model on a large dataset of prime and non-prime numbers makes the RSA algorithm's keys more random and secure.

- **Scalability**

Proposed method is based on deep neural networks which is trained on a larger dataset to generate keys of varying sizes, so the proposed approach is more scalable. And the traditional method has limitations in generating larger key sizes due to computational impaction. We have proposed 2^{36} which can be scaled to 2^n where $n > 48$.

Overall, the proposed approach has the potential to improve the security and efficiency of the key generation of the RSA algorithm.

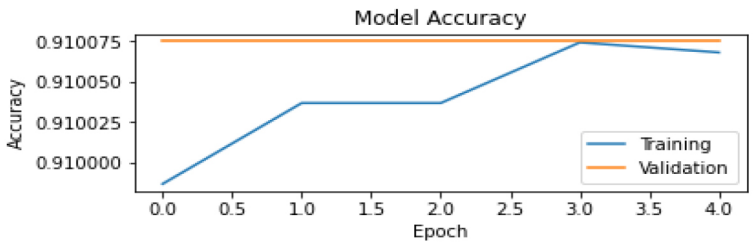

Figure 2

Model Accuracy

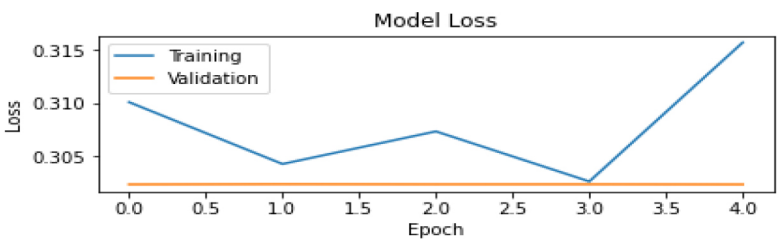

Figure 3

Model Loss

6. Conclusion

In this paper, a study of the neural network method has endeavoured to amplify the key generation process of the RSA cryptographic algorithm.

The neural networks model can be trained to learn to safeguard communications. This research paper shows the application of deep neural networks in cryptography to predict prime numbers for key generation for the RSA cryptographic algorithm with high accuracy. The evaluation of entropy and key space showcased the unpredictability and randomness of the model's outputs, reaffirming the cryptographic integrity of the generated keys. This approach ensures a secure way of transmitting sensitive information over the network by providing an additional layer of security through the integration of machine learning techniques. Although the traditional probabilistic algorithms have proven effective, this research indicate that deep neural networks can offer a competitive alternative, presenting a path to enhance the security landscape of digital communication. Deep neural networks have the capability to considerably enhance the efficiency of cryptographic systems and are a valuable tool in the field of cryptography. Neural networks can be great at cryptanalysis and inspire further research and development.

Acknowledgment

Authors express their thankfulness for the IPRF award received from 'Guru Gobind Singh Indraprastha University, New Delhi, India.'

References

[1] T. Chandra Segar and R. Vijayaragavan, "Pell's RSA key generation and its security analysis," in *Proc. 2013 Fourth Int. Conf. Comput., Commun., Netw. Technol. (ICCCNT)*, pp. 1–5, Jul. (2013).

[2] M. Agrawal, N. Kayal, and N. Saxena, "PRIMES is in P," *Ann. Math.*, pp. 781–793 (2004).

[3] Z. K. Abdalrdha, I. H. Al-Qinani, and F. N. Abbas, "Subject review: Key generation in different cryptography algorithms," *Int. J. Sci. Res. Sci., Eng., Technol.*, vol. 6, no. 5, pp. 230–240 (2019).

[4] B. R. Ambedkar and S. S. Bedi, "A new factorization method to factorize RSA public key encryption," *Int. J. Comput. Sci. Issues (IJCSI)*, vol. 8, no. 6, p. 242 (2011).

[5] D. E. Knuth, *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional (2014).

[6] L. Dongjiang, W. Yandan, and C. Hong, "The research on key generation in RSA public-key cryptosystem," in *Proc. 2012 Fourth Int. Conf. Comput. Inf. Sci.*, pp. 578–580, Aug. (2012).

[7] L. Egri and T. R. Shultz, "A compositional neural-network solution to prime-number testing," in *Proc. Annu. Meeting Cogn. Sci. Soc.*, vol. 28, pp. 28–34 (2006).

[8] S. V. K. Gaddam and M. Lal, "Efficient cancelable biometric key generation scheme for cryptography," *Int. J. Netw. Secur.*, vol. 11, no. 2, pp. 61–69 (2010).

[9] G. Ganbaatar, D. Nyamdorj, G. Cichon and T.-O. Ishdorj, "Implementation of RSA cryptographic algorithm using SN P systems based on HP/LP neurons," *J. Membr. Comput.*, vol. 3, pp. 22–34 (2021).

[10] P. P. Hadke and S. G. Kale, "Use of neural networks in cryptography: A review," in *Proc. 2016 World Conf. Futuristic Trends Res. Innov. Social Welfare (Startup Conclave)*, pp. 1–4 (2016).

[11] P. Gayathri, S. Umar, G. Sridevi, N. Bashwanth and R. Srikanth, "Hybrid cryptography for random-key generation based on ECC algorithm," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 3, p. 1293 (2017).

[12] S. M. Hussain and H. Al-Bahadili, "A DNA-based cryptographic key generation algorithm," in *Proc. Int. Conf. Secur. Manage. (SAM)*, pp. 338–340 (2016).

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press (2016).

[14] A. Jagadeesan and K. Duraiswamy, "Secured cryptographic key generation from multimodal biometrics: Feature level fusion of fingerprint and iris," *arXiv preprint*, arXiv:1003.1458 (2010).

[15] J. Jin and K. Kim, "3D CUBE algorithm for the key generation method: Applying deep neural network learning-based," *IEEE Access*, vol. 8, pp. 33689–33702 (2020).

[16] R. M. Jogdand and S. S. Bisalapur, "Design of an efficient neural key generation," *Int. J. Artif. Intell. Appl. (IJAIA)*, vol. 2, no. 1, pp. 60–69 (2011).

[17] W. Kinzel and I. Kanter, "Interacting neural networks and cryptography," in *Advances in Solid State Physics*, Springer, pp. 383–391 (2002).

[18] G. Meletiou, D. K. Tasoulis, M. N. Vrahatis, et al., "A first study of the neural network approach to the RSA cryptosystem," in *Proc. IASTED 2002 Conf. Artif. Intell.*, Calgary, pp. 483–488 (2002).

[19] I. Meraouche, S. Dutta, H. Tan and K. Sakurai, "Neural networks-based cryptography: A survey," *IEEE Access*, vol. 9, pp. 124727–124740 (2021).

[20] M. Mumtaz and L. Ping, "Forty years of attacks on the RSA cryptosystem: A brief survey," *J. Discrete Math. Sci. Cryptogr.*, vol. 22, no. 1, pp. 9–29 (2019).

[21] S. A. Nagar and S. Alshamma, "High speed implementation of RSA algorithm with modified keys exchange," in *Proc. 2012 6th Int. Conf. Sci. Electron., Technol. Inf. Telecommun. (SETIT)*, pp. 639–642 (2012).

[22] G. L. Miller, "Riemann's hypothesis and tests for primality," *J. Comput. Syst. Sci.*, vol. 13, pp. 300–317 (1976).

[23] W. A. Shukur, "A proposed method for generating a private key using digital color image," *Int. J. Appl. Eng. Res.*, vol. 12, no. 16, pp. 6235–6240 (2017).

[24] W. Stallings, *Cryptography and Network Security: Principles and Practice*. Pearson Education, United Kingdom (2020).

[25] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders and K.-R. Muller, "Explaining deep neural networks and beyond: A review of methods and applications," *Proc. IEEE*, vol. 109, no. 3, pp. 247–278 (2021).

[26] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126 (1978).

[27] L. Dong and K. Chen "Introduction of cryptographic protocols," in *Cryptographic Protocol: Security Analysis Based on Trusted Freshness*, Springer, pp. 1–12 (2012).

[28] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. CRC Press, 2016.

[29] B. A. Forouzan and D. Mukhopadhyay, *Cryptography and Network Security*, vol. 12. McGraw-Hill Education (2015).

[30] H. Lee and J. Song, "Introduction to convolutional neural network using Keras: An understanding from a statistician," *Commun. Stat. Appl. Methods*, vol. 26, no. 6, pp. 591–610 (2019).

[31] P. Manuja and R. S. Shekhawat, "Developing information security metrics and measures for risk assessment of an organization," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 4, pp. 1195–1202, May (2022), doi: https://doi.org/10.1080/09720529.2022.2075092.

[32] K. K. Jabbar, H. A. Hailal, and A. T. Naser, "Information security based on sub-system keys generator by utilizing polynomials method and logic gate," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 26, no. 4, pp. 1135–1143, Jan. (2023), doi: https://doi.org/10.47974/jdmsc-1549.

Received April, 2024