International Journal of Innovative Research and Scientific Studies, 8(5) 2025, pages: 1866-1878

ISSN: 2617-6548

URL: www.ijirss.com

Analysis of RSA algorithm cryptographic resilience using artificial intelligence methods

Zhanerke Temirbekova[1], Sakhybay Tynymbayev[2], Tolganay Chinibayeva[2*], Diana Asetova[1], Azamat Imanbayev[3]

[1]*Al-Farabi Kazakh National University, Almaty, Kazakhstan.*
[2]*Internationat IT University (IITU), Almaty, Kazakhstan.*
[3]*Kazakh-British Technical University, Almaty, Kazakhstan.*

Corresponding author: Tolganay Chinibayeva (*Email: t.chinibayeva@iitu.edu.kz*)

Abstract

This study investigates the cryptographic robustness of the RSA algorithm by applying machine learning techniques to assess its vulnerability, particularly in the context of modulus factorization, where n=p×q. The primary goal was to enhance the generation of random numbers within RSA systems to reduce the feasibility of factorization-based attacks. Four machine learning models were examined: Random Forest Classifier, Decision Tree, XGBoost, and a Sequential Model (neural network). These models were trained and evaluated using data relevant to RSA key generation and threat detection scenarios. A comparative performance analysis was conducted based on key classification metrics, including accuracy, precision, recall, F1-score, and ROC AUC. The Random Forest Classifier demonstrated superior overall performance, offering balanced detection across classes and high generalization capability. In contrast, the Sequential Model, despite high accuracy on paper, failed to identify minority class instances, limiting its reliability. The results suggest that integrating artificial intelligence, particularly ensemble learning methods, into cryptographic systems can improve the security of RSA against classical threats such as factorization. These findings highlight the potential of machine learning to support future developments in adaptive and intelligent cryptographic defense mechanisms.

Keywords: Artificial method, Cryptographic strength of the algorithm, Decision Tree, Random Forest Classifier, RSA cryptosystem, Sequential Model, XGBoost.

DOI: 10.53894/ijirss.v8i5.9313
Funding: This study received no specific financial support.
History: Received: 26 June 2025 / **Revised:** 28 July 2025 / **Accepted:** 30 July 2025 / **Published:** 15 August 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

The continuous evolution of cryptographic attacks necessitates the exploration of new approaches to evaluating the security of cryptographic algorithms. Traditional methods for assessing the security of RSA are primarily based on mathematical assumptions regarding its robustness, such as the difficulty of integer factorization and modular exponentiation. However, recent advancements in artificial intelligence (AI) offer new opportunities to analyze cryptographic resilience using data-driven methods.

Machine learning models, particularly Random Forest Classifier, XGBoost, Decision Tree, and Sequential Model, offer promising approaches for assessing RSA security. The Random Forest classifier is employed to categorize RSA key vulnerabilities based on various parameters, such as key length, prime number properties, and side-channel leakage patterns.

This model's ability to process structured tabular data makes it well-suited for identifying patterns in cryptographic weaknesses.

XGBoost, an advanced implementation of gradient boosting, demonstrates high accuracy in analyzing RSA cryptographic parameters due to its staged training of weak learners and efficient handling of previous iteration errors.

Its ability to manage missing values, regulate overfitting, and account for feature importance makes XGBoost particularly effective for detecting vulnerabilities in large and diverse datasets. However, the model’s high complexity can make the interpretation of results more challenging compared to simpler algorithms.

The Decision Tree model analyzes RSA cryptographic parameters by constructing an interpretable decision tree, where each branch represents a step in the decision-making process. Due to its transparency, the model enables the identification of key features that influence the algorithm’s cryptographic strength.

The Sequential Model, based on a neural network architecture, analyzes complex nonlinear relationships between cryptographic parameters and potential indicators of attack success. By learning from large datasets containing RSA key structures and attack outcomes, it can uncover hidden vulnerabilities that may not be apparent through traditional cryptanalysis methods.

The Random Forest Classifier is an ensemble machine learning method that builds multiple decision trees and aggregates their predictions to improve model accuracy and robustness. In the context of analyzing the cryptographic strength of the RSA algorithm, Random Forest effectively identifies statistical patterns between key parameters and attack outcomes. By leveraging a large number of training samples, the method can determine the most influential features affecting the likelihood of cryptosystem compromise, thereby enhancing the capabilities of predictive cryptanalysis compared to traditional approaches.

Assessing the cryptographic strength of RSA using artificial intelligence methods offers a modern perspective on security evaluation. This study explores the applicability of machine learning techniques for predicting RSA vulnerabilities, compares their effectiveness with traditional cryptanalysis, and identifies key factors influencing RSA resilience. The integration of artificial intelligence into cryptanalysis has the potential to enhance proactive security measures, ensuring the robustness of cryptographic systems against emerging threats.

The aim of this study is to assess the cryptographic resilience of the RSA algorithm using machine learning methods. In this context, resilience refers to the algorithm’s ability to maintain robustness across various parameter configurations that may be potentially vulnerable to attacks. The analysis is based on the predictive capabilities of machine learning models in estimating the likelihood of a successful attack depending on the characteristics of cryptographic keys. The study examines and compares four popular models: Random Forest Classifier, XGBoost, Decision Tree, and Sequential Model.

The use of the Random Forest Classifier for analyzing and identifying hidden patterns in RSA key structures is expected to achieve high accuracy in classifying potentially vulnerable keys. Owing to its ability to efficiently handle heterogeneous data and account for complex feature interactions, the algorithm demonstrates stable and balanced performance even in the presence of class imbalance within the dataset.

1.1. Problem Statement

The rapid advancement of artificial intelligence and its application across various domains, including cybersecurity, has opened new avenues for analyzing the security of cryptographic algorithms. However, despite the growing use of AI in cryptographic research, there remains a lack of studies specifically focused on evaluating the cryptographic resilience of the RSA algorithm using AI-driven methods. Traditional approaches to RSA security analysis are primarily based on mathematical cryptanalysis techniques, such as integer factorization and lattice-based attacks. While these methods have proven effective, they may fall short in addressing emerging threats posed by advances in artificial intelligence and quantum computing.

This study aims to bridge the gap by developing and implementing an artificial intelligence–based methodology for evaluating the security of the RSA cryptosystem. The research focuses on identifying potential vulnerabilities using machine learning and neural networks, and comparing the effectiveness of AI-driven analysis with that of traditional cryptographic methods. By integrating artificial intelligence into cryptanalysis, the authors seek to determine whether AI-based techniques can offer a new perspective on RSA security and contribute to the development of more robust cryptographic protection.

The RSA algorithm is one of the most popular asymmetric encryption methods and is widely used to ensure security in modern information systems. It is based on the difficulty of factoring large numbers, which makes it cryptographically secure when using the appropriate keys. However, with the development of computing technologies, in particular AI, new approaches to analyzing the cryptographic strength of RSA appear. The RSA algorithm was proposed in Rivest et al. [1]. It

uses a pair of keys: a public one for encrypting data and a private one for decrypting it. The key idea is that the algorithm is based on the difficulty of factoring the product of two large prime numbers. Let $n = p \times n$, where p and q are two large prime numbers, and n is the modulus for encryption and decryption operations [2].

The article Zhang et al. [3] shows that with the development of computing technology and mathematical methods, as well as quantum computing and repetition, you may face new challenges. This increases the need to study its cryptographic strength and develop new approaches to prevent possible attacks. In Biswas and Das [4], modern algorithms, as they are a method of estimating the world field (NFS), have significantly improved the efficiency of corporatization. This makes existing keys vulnerable to potential quantum threats. In some cases, a weak implementation of RSA, including the use of identical modules for different users or low-security parameters, can lead to successful attacks. In the research Wang et al. [5] devoted to cryptographic algorithms, two popular algorithms are considered in detail: symmetric encryption, asymmetric encryption, and the AES and RSA algorithms. The author of the study analyzes the design of these algorithms, their vulnerabilities, and production characteristics. As part of the work Disanayaka et al. [6] tests are conducted to evaluate the encryption of time, processor, and memory usage under various conditions, which may be useful for a deeper analysis of the cryptographic stability of the ASR algorithm using the artificial encryption method. The work Barker [7] indicates that Russia remains one of the few countries using mathematical modeling methods that are universal for business, indicative of cryptographic strength and the ability to adapt to new challenges. Its application covers many areas, from financial data protection and authentication to the implementation of low-level systems such as Internet of Things devices.

In the article Wang and Zhang [8] It is possible to hack the RSA server using business trips, demonstrating that such shift commands of the system can ensure the security of South Africa. The researchers Rivest et al. [1] present improved algorithms for factoring large numbers, which can reduce the security of RSA with shorter key lengths.

In Okumuş and Celık [9] the implementation of the algorithm in low-level DSA systems is considered, taking into account resource constraints and ensuring the necessary level of information protection.

The paper Kiratsata and Panchal [10] provides a method for protecting the data of the Regional State Administration from attacks using additional channels. Improvements are proposed for implementing masked ASR, which make it difficult to extract the private key by analyzing the physical characteristics of the algorithm execution, such as execution time or power consumption.

The article Jain and Gupta [11] shows how the use of homomorphic encryption with the algorithms of OGA and Paillier can increase the cryptographic strength of algorithms used to protect data in the learning process of machine learning models. This helps to improve the security of the RSA algorithm while minimizing the risks of information leaks and attacks on the confidentiality of data processed in cloud systems.

The paper Rivest et al. [1] shows that CSAs can be susceptible to various types of attacks, such as side-channel attacks, selective vulnerabilities, and, in the future, attacks using quantum algorithms, which leads to the need to constantly update algorithm security recommendations. In traditional OGA security threats, an important aspect is the use of keys that are too small or weak. The article shows Zhang and Lee [12] that to ensure the reliability of the algorithm, the minimum key length should be 2048 bits. This requirement is confirmed by numerous studies that show that factorization of numbers with shorter-length keys can be performed using the computing power currently available. Increasing the key length to 2048 bits makes factorization much more difficult and practically impossible using classical computational methods [13].

Despite the growing interest in the use of artificial intelligence to solve cryptographic problems, direct research specifically aimed at assessing the cryptographic strength of RSA using artificial methods is insufficient. Existing research mainly focuses on cryptanalysis using artificial intelligence in a general sense but does not provide a systematic methodology for evaluating RSA security using machine learning or neural networks.

Traditional cryptanalytic approaches, such as factorization-based attacks and mathematical reliability assumptions, have been thoroughly studied, but may prove ineffective in adapting to evolving computational paradigms, such as quantum computing and artificial intelligence attacks. The lack of comparative studies between artificial intelligence-based methods and classical methods leaves a gap in understanding whether artificial intelligence can provide a meaningful advantage in evaluating and improving RSA security.

2. Materials and Methods

More recently, AI methods have begun to develop for analyzing the cryptographic strength of algorithms, including RSA. Research in the field of machine learning shows that AI is being used to automatically identify vulnerabilities in cryptographic algorithms. Using deep learning approaches, it is possible to predict possible attacks and develop new methods of protection that will be much more effective than traditional methods. These methods are useful for predicting possible attacks and assessing the vulnerability of RSA in new conditions. The RSA algorithm remains the main element of the cryptographic infrastructure, but its cryptographic strength is increasingly threatened by the development of technologies such as quantum computing. Increasing the key length, using hybrid systems, and protecting against side-channel attacks are the main measures to maintain security. At the same time, new research, including in the field of artificial intelligence, opens up new opportunities for analyzing and improving the security of encryption algorithms.

The article discusses the cryptographic strength of RSA, which is determined by the time and computing resources required to successfully attack the cryptosystem. An example of calculating cryptographic stability is an estimate of the time required to factorize the modulus n=p*q, where p and q are large prime numbers.

Selection of criteria for evaluating cryptographic strength:

The article discusses several key criteria for evaluating the cryptographic strength of the RSA algorithm:

The complexity of factorization, which is a traditional cryptanalysis method that determines the strength of the RSA

algorithm against attacks based on modulus decomposition into prime factors.

Machine learning methods are a promising direction in cryptanalysis, which makes it possible to identify hidden patterns in the structure of keys and predict possible vulnerabilities.

A comparison of cryptographic strength assessment methods is presented in Table 1, which discusses its main characteristics, advantages, and limitations in the context of RSA security analysis.

Table 1.
Comparison of methods for evaluating the cryptographic strength of the RSA algorithm.

The method of analysis	Main advantages	The main disadvantages
Factorization analysis	A proven classical approach	Depends on the power of the computers
Machine learning	Able to identify new vulnerabilities	Requires a lot of data for training

The selected criteria provide a comprehensive analysis of RSA security, as they:

- Allow a comparison between classical and artificial cryptanalysis methods.
- Provide a quantitative and qualitative assessment of the level of protection of the cryptosystem.

There are other approaches to cryptographic strength analysis, for example:

- Statistical analysis of randomness - used to assess the quality of key generation, but does not provide a complete picture of their vulnerability;
- Entropy measurements are useful for analyzing resistance to attacks on key predictability but are less effective for factorization attacks.
- Security analysis of protocols using RSA - considers the strength of the algorithm in real-life scenarios, but does not always assess fundamental cryptographic strength.

These methods were not included in the main study, as they are either redundant for the context under consideration or less relevant for comparing traditional and artificial cryptanalysis of RSA.

Thus, the choice of these criteria allows for an objective comparison of RSA cryptographic strength analysis methods, identification of key vulnerabilities, and determination of how effective artificial cryptanalysis methods are in comparison with classical approaches.

To assess the cryptographic strength of the RSA algorithm, let us consider factorization analysis. Factorization analysis on RSA is related to the fundamental vulnerability of the RSA algorithm, which is the complexity of factoring the product of two large prime numbers (modulus n).

An RSA factorization attack is an attempt by an attacker to find the prime numbers p and q that make up the modulus n=p*q by applying factorization algorithms. If an attacker successfully performs factorization of n, he will be able to calculate the RSA private key and decrypt all messages encrypted using the public key.

```
Algorithm 1: Factorization analysis (Multiplier attacks)
  Input n = 2773, e = 17, P = 89
Function factorize_n(n):
   For i = 2 to sqrt(n) + 1:
      If n % i == 0:
         Return i, n // i # Found factors
   End If
   Return None, None        # Return None if no factors found
Function euler_phi(p, q):
   Return (p - 1) * (q - 1) # Euler's Totient function
Function extended_gcd(a, b):
   old_r, r = a, b
   old_s, s = 1, 0
   old_t, t = 0, 1
   While r != 0:
      quotient = old_r // r
      old_r, r = r, old_r - quotient * r
      old_s, s = s, old_s - quotient * s
      old_t, t = t, old_t - quotient * t
   End While
   Return old_s, old_t, old_r  # Return gcd and coefficients
Function mod_inverse(e, phi_n):
   s, t, gcd = extended_gcd(e, phi_n)
   If gcd != 1:
      Output: "Inverse does not exist"
      End
   End If
   Return s % phi_n  # Modular inverse
Function rsa_attack(n, e, P):
   p, q = factorize_n(n)
```

```
    If p is None or q is None:
        Output: "Error: Failed to factor n"
        End
    End If
    Output: "Factorization: p = {p}, q = {q}"
    phi_n = euler_phi(p, q)
    Output: "Euler's Totient: φ(n) = {phi_n}"
    d = mod_inverse(e, phi_n)
    Output: "Secret exponent d = {d}"
    C = pow(P, e, n)
    Output: "Encrypted message C = {C}"
  decrypted_message = pow(C, d, n)
    Output: "Decrypted message P = {decrypted_message}"
End
```

Suppose we have RSA with a modulus n of length 2048 bits (256 bytes), and we want to estimate the time required to factor it using a modern factoring algorithm, such as the number field sieve (NFS).

1. NFS Complexity: The NFS algorithm has subexponential complexity:

$$Ln[\frac{1}{3}\left(\frac{64}{9}\right)\frac{1}{3} = \exp\left(\left(c + o(1)\right) * \frac{(lnn)1}{3} * (lnlnn)\frac{2}{3}\right), \qquad c \approx 1.923$$

2. Estimated Computational Load: For 2048-bit n:

$lnn \approx 2048 \cdot ln2 \approx 1420.65$

$$lnlnn \approx ln1420.65 \approx 7.26.$$

Substitute into the formula: $Time \approx exp(1.923 * \frac{(1420.65)1}{3} * (7.26)2/3)$

3. Result: The calculation provides an approximate estimate of Ln≈2112 operations. This is equivalent to: for a supercomputer with a performance of 1018 operations per second, it will take about 2112-60≈1015 seconds (more than 30 million years).

4. Transition to practice: In practice, RSA-2048 factorization is not achievable. The largest decomposed modulus is RSA-250 (829 bits), which was accomplished in 2020 using thousands of cores over several months. For RSA with a key length of 2048 bits, the time required for a successful attack exceeds the capabilities of modern computers. However, with increasing computing power or the advent of quantum computers, the resistance may be reduced.

Using Artificial Intelligence in RSA Cryptanalysis

The AI-based cryptanalysis of the RSA algorithm involves training a neural network using known public keys and their corresponding private keys or encrypted data. Once trained, the neural network can predict private keys or decode encrypted messages if the public key is known.

In this paper, we predict a private key based on a public key. As discussed earlier, a neural network is used to train and predict a private key (d) given public parameters (modulus n and public exponent e). This requires a large dataset that includes public and private keys so that the model can uncover hidden patterns. The input data for the network is (n, e) the public parameters. The target data for the network is d, the private key.

Also, an attack on weak keys is considered. Weak RSA keys can arise due to the predictability of some parameters or insufficient key length. AI systems are used to analyze keys and identify weak or vulnerable keys that are susceptible to attack. This will allow algorithms to be optimized for generating secure keys or attacking weak keys using neural networks.

Algorithm structure prediction was developed. Neural networks were trained to identify weaknesses in the RSA algorithm itself. To find ways to reduce the complexity of finding a private key, algorithms such as "genetic cryptanalysis" or other evolutionary methods adapted to the features of the RSA cryptosystem can be employed.

In practice, the factorization of RSA-2048 remains infeasible. The largest successfully factored RSA modulus to date is RSA-250 (829 bits), which was achieved in 2020 using thousands of CPU cores over several months. For RSA with a 2048-bit key length, the time required for a successful attack exceeds the capabilities of current classical computing systems. However, with the advancement of computational power or the emergence of quantum computers, this level of resilience may be significantly reduced.

AI-based cryptanalysis of the RSA algorithm involves training a neural network using known public keys and their corresponding private keys or encrypted data. Once trained, the neural network may be able to predict secret keys or even decrypt ciphertexts when provided with the public key.

In this study, we aim to predict the private key based on the public key. As previously discussed, a neural network is employed to train and predict the private key (d) given the public parameters the modulus (n) and the public exponent (e). This approach requires a large dataset containing both public and corresponding private keys, allowing the model to learn hidden patterns. The input to the network consists of (n, e) the public parameters while the target output is d, the private key.

The study also considers attacks on weak keys.

Weak RSA keys may arise due to the predictability of certain parameters or insufficient key length. AI systems are used to analyze keys and identify weak or vulnerable ones that are susceptible to attacks. This can help optimize algorithms

for generating secure keys or facilitate targeted attacks on weak keys using neural networks.

A structural prediction algorithm was developed, and neural networks were trained to identify weak points within the RSA algorithm itself. The goal was to explore ways to reduce the complexity of private key discovery by applying techniques such as genetic cryptanalysis or other evolutionary methods adapted to the specific characteristics of the RSA cryptosystem.

During the study, a synthetic dataset was generated containing 500 samples of paired RSA cryptographic keys. Each sample included features such as the lengths of the prime numbers p and q their difference, logarithms of the modulus n, and Euler's totient function, $\varphi(n)$, as well as the logarithm of the public exponent e. A class label was assigned based on conditions potentially indicating key vulnerability (e.g., a small e value or p and q being too close in value).

Feature standardization was applied for data normalization. The dataset was then randomly split into training and testing subsets in an 80/20 ratio, with stratified sampling ensured to maintain class distribution.

Four machine learning algorithms were used for classification:

- Random Forest Classifier;
- XGBoost;
- Decision Tree;
- Neural Network (Sequential Model).

The training and testing architecture was identical for all models; the only variation lay in the choice of classifier. Tree depth, the number of trees, and other hyperparameters were manually tuned to improve accuracy while avoiding overfitting.

3. Results and Discussion

The analysis of RSA cryptographic resilience revealed that one of the key factors affecting the algorithm's security is the use of overly small or weak keys. To ensure reliability and resistance against conventional security threats, the minimum key length should be 2048 bits. This requirement is supported by research findings demonstrating that the factorization of shorter keys can be effectively performed using current computational capabilities (Figure 1). However, given the anticipated advancement of quantum technologies and their associated threats, it is also recommended to consider transitioning to cryptographic systems that are resistant to quantum attacks.

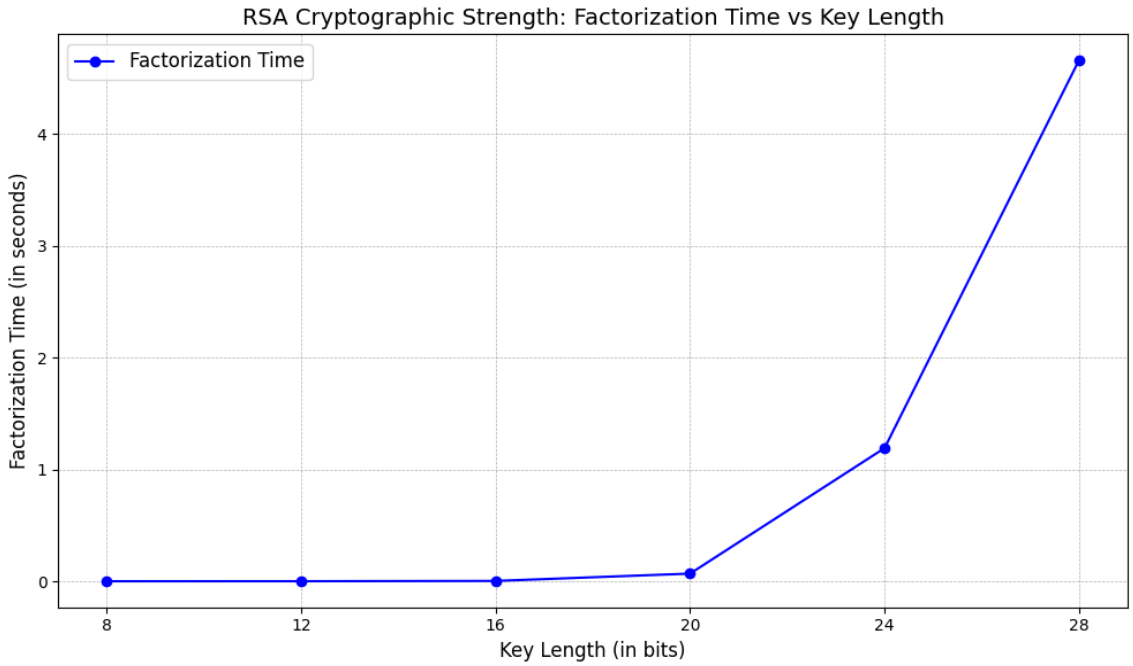

Figure 1.
Dependence of factorization time on the size of the modulus n.

The graph illustrates the relationship between the factorization time of the RSA modulus n and key length (in bits). The data were obtained through brute-force methods. It is evident that the factorization time grows exponentially with increasing key length. For keys up to 16 bits in length, factorization is completed within fractions of a second. However, for keys of 20 bits and above, the required time increases significantly, highlighting the complexity of the factorization problem. This supports the high cryptographic resilience of RSA when using standard-length keys, such as 2048 bits.

The use of long keys makes factorization virtually infeasible with current computational capabilities.

This study conducted a comparative analysis of four popular models, Random Forest Classifier, XGBoost, Decision Tree, and Sequential Model, to improve the quality of random number generation or cybersecurity analysis in the RSA cryptosystem.

The evaluation of models was based on the following criteria:

1. Accuracy in classifying vulnerabilities.

This metric reflects the model's ability to correctly distinguish cryptographic keys with varying levels of strength based on their parameters, directly impacting the reliability of predictions regarding potential vulnerabilities.

2. Interpretability of results.

An important aspect that facilitates understanding of the internal patterns identified by the model. High interpretability supports the analysis of factors influencing key security and enables the formulation of well-grounded cryptanalytic conclusions.

3. Handling of complex dependencies.

This criterion assesses the model's capability to uncover hidden nonlinear relationships, correlations, and anomalies in cryptographic data, which is particularly important when analyzing parameters that affect the likelihood of a successful attack.

4. Resistance to overfitting.

This evaluates the model's ability to maintain high accuracy when generalizing to new, previously unseen data — a critical factor for deploying the model in real-world cryptographic scenarios.

This study conducted a comparative analysis of four popular machine learning models for tasks related to random number generation and cybersecurity analysis within the RSA cryptosystem: Random Forest Classifier, Decision Tree, XGBoost, and Sequential Model. To address the binary classification task in this study, the Random Forest Classifier was employed an ensemble machine learning method based on constructing multiple decision trees followed by majority voting. This algorithm is known for its high resistance to overfitting and its ability to effectively detect complex dependencies within the data.

The performance of the model was evaluated using standard classification metrics, including accuracy, precision, recall, and the F1-score. The model achieved an accuracy of 1.000 on the training set and 0.950 on the test set, indicating a high level of generalization capability. The confusion matrix presented in Figure 2 shows that out of 100 test samples, 95 were correctly classified: 50 instances of class 0 and 45 instances of class 1. Only 5 instances were misclassified, which is reflected in the high F1-score values of 0.95 for both classes.

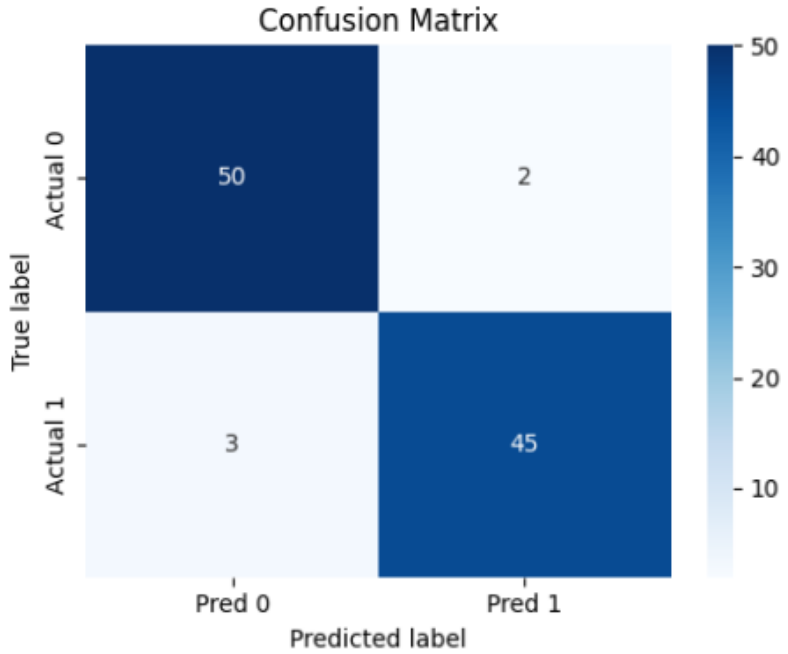

Figure 2.
The output of the Random Forest Classifier, including the confusion matrix during the training process.

Additionally, an ROC curve was constructed (Figure 3) to demonstrate the model's ability to distinguish between classes at various classification thresholds.

The area under the ROC curve (AUC) was 0.96, confirming the model's high sensitivity and specificity. An AUC value close to 1.0 indicates excellent diagnostic performance of the classifier.

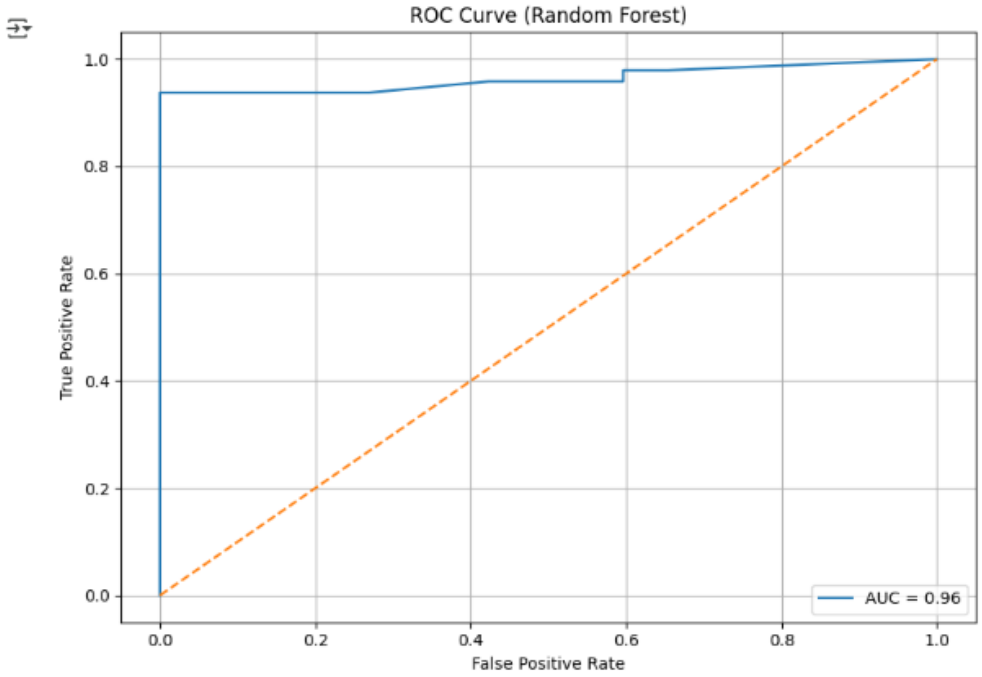

Figure 3.
ROC curve obtained for the Random Forest Classifier model.

Thus, the Random Forest model demonstrated high effectiveness and reliability in solving the classification task, providing balanced accuracy metrics and strong resistance to overfitting.

The study also evaluated the performance of the Decision Tree Classifier, which constructs a decision tree for object classification. The main advantages of this model are its simplicity and high interpretability. The model achieved an accuracy of 88% on the test set and 89% on the training set, indicating stable performance.

Figure 4 shows the confusion matrix. The model correctly identified 78 instances of the "Vulnerable" class and 10 instances of the "Secure" class. However, 11 instances of the "Secure" class were misclassified, resulting in a low recall of 0.48 for the "Secure" class, while the recall for the "Vulnerable" class reached 0.99.

Key performance metrics of the model:

Accuracy — 0.88

Precision — 0.88

Recall — 0.88

F1-score — 0.86 (macro average across classes)

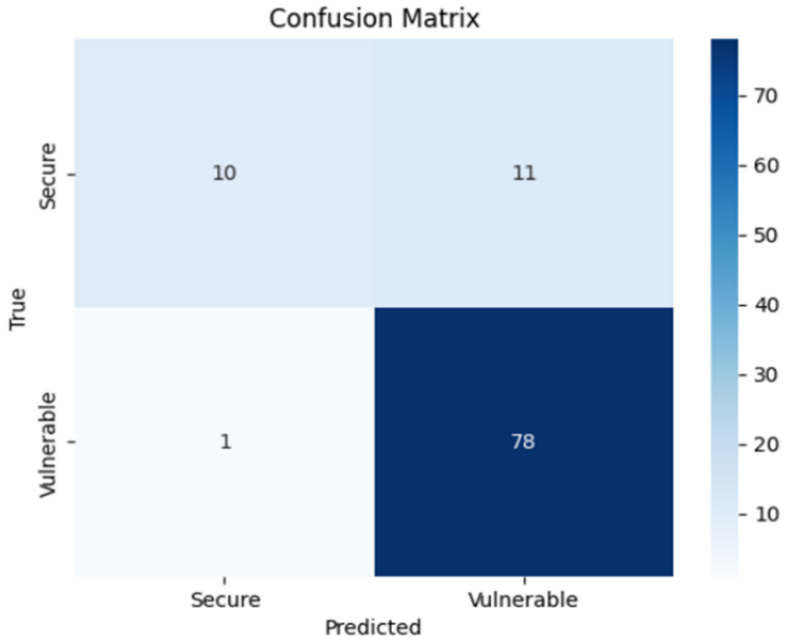

Figure 4.
The output of the Decision Tree Classifier including the confusion matrix during the training process.

Figure 5 presents the ROC curve for the Decision Tree Classifier. The area under the curve (AUC) was 0.80, indicating a moderate ability of the model to distinguish between classes compared to the Random Forest Classifier.

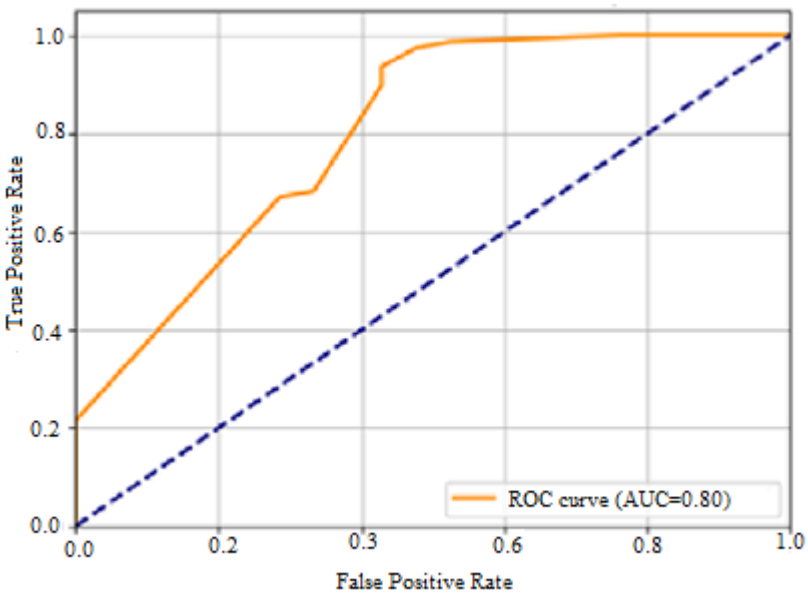

Figure 5.
ROC curve obtained for the Decision Tree Classifier model.

Thus, the decision tree showed good overall performance but struggled to accurately classify one of the classes, particularly the "Secure" class.

This may be attributed to the fact that decision trees are sensitive to class imbalance in the data.

The performance of the XGBoost Classifier, one of the most advanced ensemble gradient boosting methods, was also evaluated. This model is known for its high performance and is widely used in classification tasks due to its ability to handle imbalanced and complex data. However, its accuracy on the test set was 64%, which is significantly lower compared to the other models examined in this study.

Figure 6 presents the confusion matrix. The model correctly classified 63 instances of the "Secure" class and only 1 instance of the "Vulnerable" class. However, it misclassified 28 instances of the "Secure" class and 8 instances of the "Vulnerable" class.

Key classification metrics:

Precision — 0.81.

Recall — 0.64.

F1-score — 0.71.

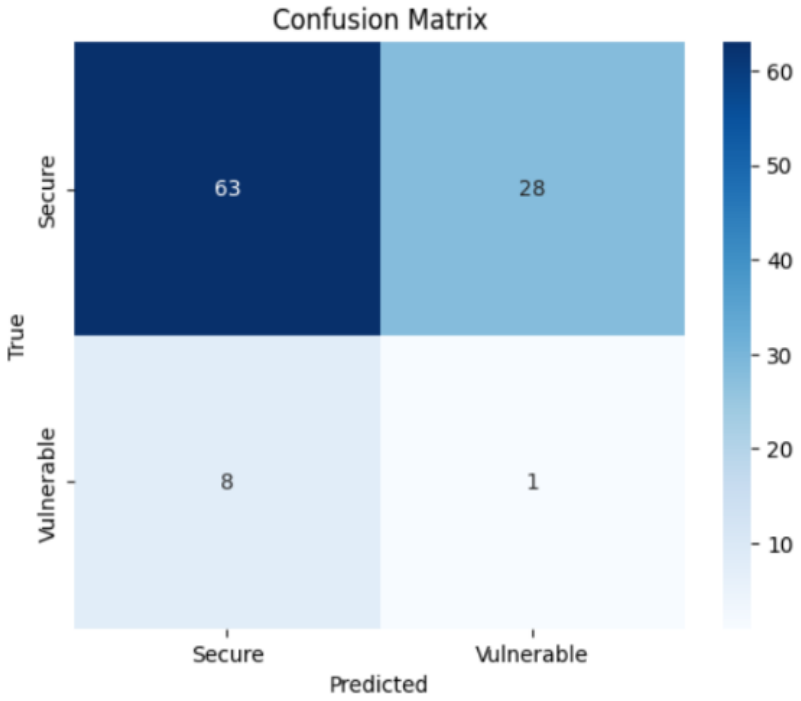

Figure 6.
The output of the XGBoost Classifier, including the confusion matrix, is shown during the training process.

To evaluate the model's ability to distinguish between classes, an ROC curve was constructed and is shown in Figure 6. The area under the curve (AUC) was 0.44, which is below the 0.5 threshold that indicates random guessing. This result reflects the low diagnostic capability of the model on the current dataset.

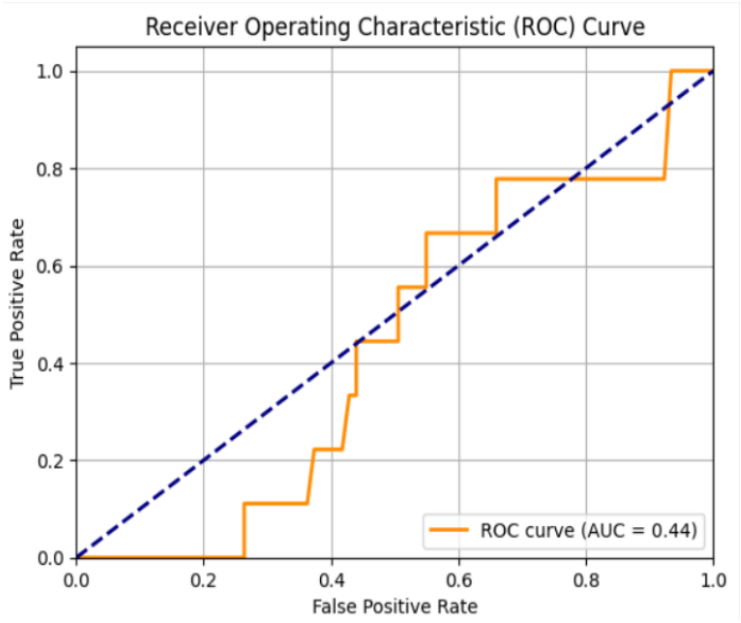

Figure 7.
ROC curve obtained for the XGBoost Classifier model.

Thus, despite the widespread use and high effectiveness of XGBoost in other tasks, the model demonstrated suboptimal performance in this study, which is likely due to the characteristics of the dataset or the chosen training parameters.

As part of the study, a Sequential Model based on a neural network was trained and tested.
The training was conducted over 50 epochs, with the dynamics of the loss function and accuracy presented in Figures 8 and 9.

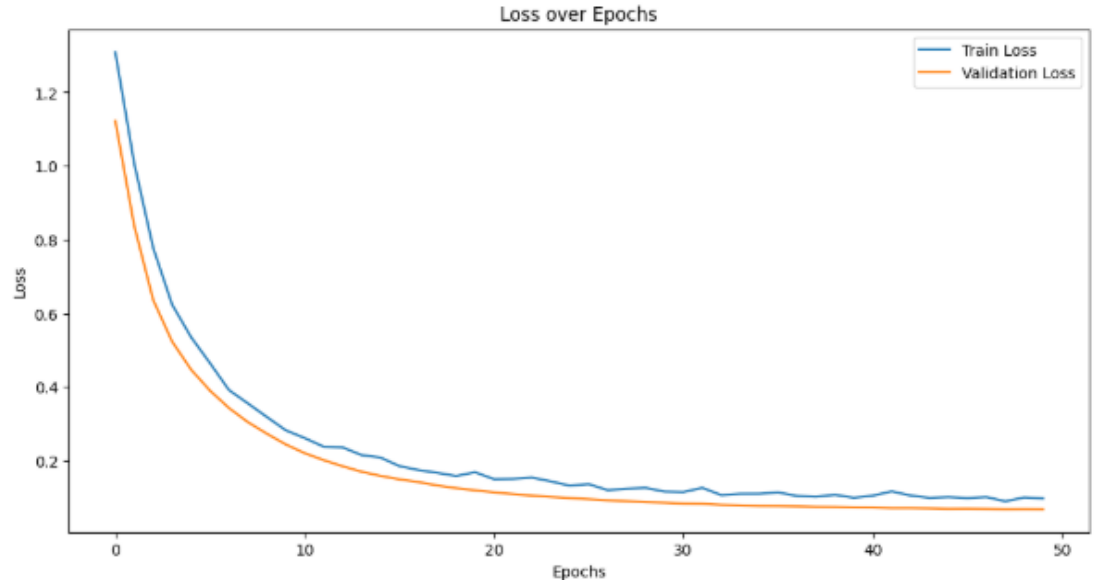

Figure 8.
Loss over training epochs for the Sequential Model, showing both training and validation loss.

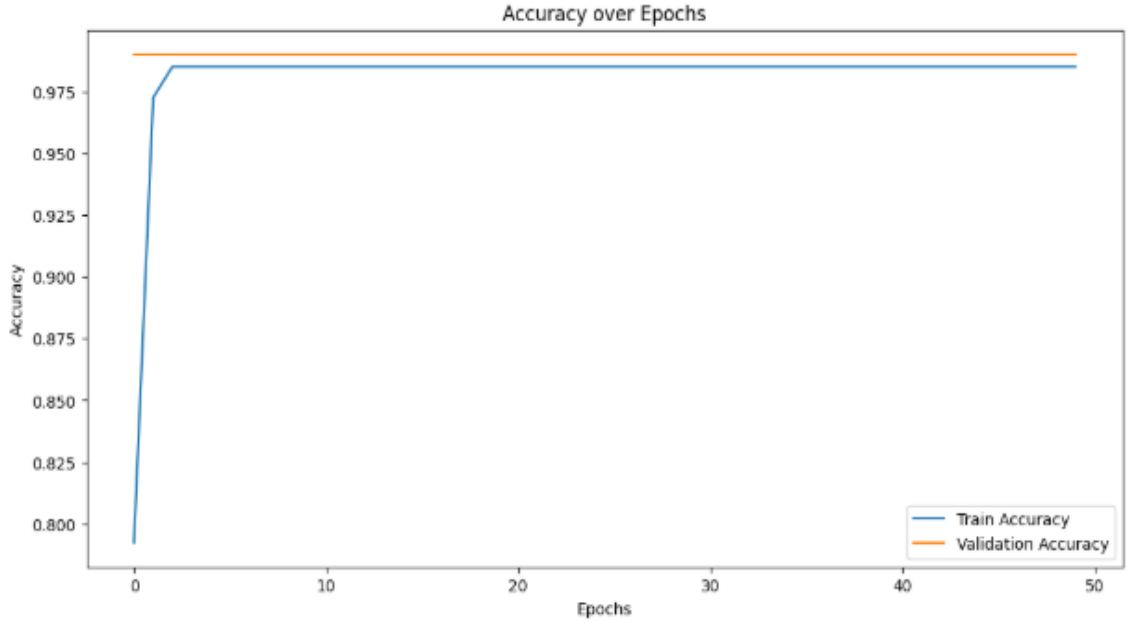

Figure 9.
Accuracy over training epochs for the Sequential Model, showing both training and validation accuracy.

During training, a stable decrease in the loss function was observed for both the training and validation sets. The model achieved high accuracy during both training and validation, 0.98, indicating good convergence. However, despite the high accuracy on the training data, testing results revealed significant limitations of the model. Figure 10 presents the confusion matrix, which shows that the model correctly classified 99 instances of the "Secure" class but failed to correctly classify any instance of the "Vulnerable" class. The only instance from this class was misclassified.

Key performance metrics of the model:

Accuracy — 0.99

Precision (weighted average) — 0.98

Recall (weighted average) — 0.99

F1-score (weighted average) — 0.99

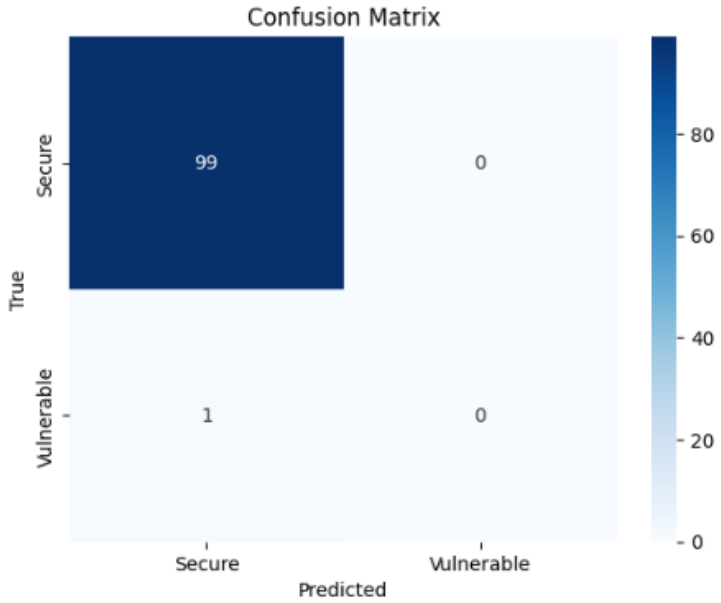

Figure 10.
The output of the Sequential Model, including the confusion matrix, during the training process.

For the "Vulnerable" class, all metric values (precision, recall, and F1-score) were 0.00, indicating the model's inability to detect this class. This points to a data imbalance problem or a bias toward one of the classes. The absence of positive predictions makes it impossible to reliably evaluate the area under the ROC curve (AUC). Thus, despite the seemingly high overall accuracy, the model failed to effectively classify instances of the "Vulnerable" class. This raises concerns about its practical applicability in real-world cryptographic tasks, where accurately identifying weak keys is critical.

Table 2.
Model comparison.

Model	Accuracy	Precision	Recall	F1
Random Forest Classifier	0.95%	0.95	0.95	0.95
Decision Tree Classifier	0.88%	0.88	0.88	0.86
XGBoost Classifier	0.64%	0.81	0.64	0.71
Sequential Model	0.99%	0.98	0.98	0.99

Table 2 presents a comparative summary of the four models used in this study: Random Forest Classifier, Decision Tree Classifier, XGBoost Classifier, and Sequential Model. Standard classification performance metrics were used to evaluate model effectiveness, including accuracy, precision, recall, and F1-score.

The Random Forest Classifier demonstrated the best performance across all major metrics, including a well-balanced precision and recall (both with weighted values of 0.95) and the highest area under the ROC curve (AUC = 0.96).

These results indicate the model's strong ability to distinguish between classes and its robustness against overfitting. The Decision Tree Classifier showed acceptable results with an accuracy of 0.88 and an AUC of 0.80; however, it falls short of the Random Forest in terms of both accuracy and robustness to class imbalance. Although the XGBoost Classifier achieved a high precision of 0.81, it showed low recall and AUC values (0.64 and 0.44, respectively), indicating a limited ability to recognize one of the classes. The Sequential Model formally achieved high metric values (accuracy = 0.99, F1-score = 0.99); however, it failed to recognize any instances of the "Vulnerable" class, which significantly reduces its practical utility. For this reason, the AUC was not computed. Thus, the most effective and reliable model in this study is the Random Forest Classifier, demonstrating a strong balance between accuracy, generalization ability, and the capacity to classify both classes effectively.

4. Conclusion

This study presents a comprehensive assessment of the cryptographic resilience of the RSA algorithm using machine learning methods aimed at optimizing random number generation and analyzing threats related to the factorization of the modulus $n = p \times q$. The comparative analysis of the Random Forest Classifier, XGBoost, Decision Tree, and Sequential Model showed that the use of the Random Forest classifier significantly enhances the effectiveness of RSA cryptosystem protection. This improvement is achieved through more reliable random number generation, which reduces the likelihood of successful factorization-based attacks.

The results confirm the potential of integrating artificial intelligence technologies into cryptanalysis processes to strengthen the cryptographic resilience of the RSA algorithm and highlight promising directions for further research in cryptographic security. Future work will explore possible combinations of various machine learning models to develop

more adaptive and effective protection mechanisms capable of countering modern cyber threats.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1977.

[2] T. S. Obaid, "Study a public key in RSA algorithm," *European Journal of Engineering and Technology Research*, vol. 5, no. 4, pp. 395-398, 2020.

[3] R. Zhang, J. Bi, L. Li, and H. Peng, "An optimal bound for factoring unbalanced RSA moduli by solving generalized implicit factorization problem," *The Journal of Supercomputing*, vol. 81, no. 1, p. 102, 2024. https://doi.org/10.1007/s11227-024-06478-y

[4] S. Biswas and P. Das, *Analysis of quantum cryptology and the RSA algorithms defense against attacks using Shor's algorithm in a post-quantum environment. In Computational Intelligence in Communications and Business Analytics*. Cham, Switzerland: Springer, 2023.

[5] Y. Wang, H. Zhang, and H. Wang, "Quantum polynomial-time fixed-point attack for RSA," in *Proceedings of the International Conference on Communications (CC)*, 2018.

[6] N. Disanayaka, D. Nanayakkara, R. Harshamal, and K. Wijesinghe, *Analysis and implementation of AES and RSA*. Malabe, Sri Lanka: Sri Lanka Institute of Information Technology, 2025.

[7] E. Barker, *Recommendation for key management*. Gaithersburg, MD, United States: NIST Special Publication, 2020.

[8] Y. Wang and H. Zhang, "Quantum algorithm for attacking RSA based on Fourier transform and fixed-point," *Wuhan University Journal of Natural Sciences*, vol. 26, no. 6, pp. 489–494, 2021. https://doi.org/10.1051/wujns/2021266489

[9] İ. Okumuş and E. Celık, "A modified key generation algorithm to rebalanced-RSA and RPower-RSA," *MANAS Journal of Engineering*, vol. 12, no. 2, pp. 192-197, 2024. https://doi.org/10.51354/mjen.1524490

[10] H. J. Kiratsata and M. H. Panchal, "A comparative analysis of machine learning models developed from homomorphic encryption based RSA and Paillier algorithm," in *Proceedings of the Fifth International Conference on Intelligent Computing and Control Systems (ICICCS 2021), 9432348*, 2021.

[11] S. Jain and S. Gupta, "Deep learning approaches for predicting weaknesses in RSA encryption in cryptocurrency transactions," *International Journal of Cryptography Research*, vol. 12, no. 1, pp. 15–28, 2022.

[12] X. Zhang and M. Lee, "Optimization of RSA key generation using genetic algorithms in cryptocurrency," *Journal of Cryptographic Engineering*, vol. 11, no. 2, pp. 101–115, 2021.

[13] F. Pistono and R. V. Yampolskiy, "Unethical research: How to create a malevolent artificial intelligence," presented at the 25th International Joint Conference on Artificial Intelligence (IJCAI-16). Ethics for Artificial Intelligence Workshop (AI-Ethics-2016), New York, 2016.