Advancing the RSA Cryptanalysis: An Experimental Demonstration of Plaintext Recovery Using Neural Networks

Vaideeshwaran Saravanan, Charlie Obimbo, Fatemeh Khoda Parast
University of Guelph
Canada

Abstract

Cryptography is essential to cybersecurity since it guarantees data's secrecy, integrity, and validity against threats. The fast expansion of artificial intelligence (AI) in many applications needs the equal importance of mitigating AI-induced cybersecurity vulnerabilities and protecting AI from cyberattacks. This study investigates a plaintext recovery assault on the most common public key algorithm of RSA cryptographic technique using a 66-bit public key and a compact neural network with 1,276 parameters. The testing reaches Bit Probability Accuracy, precision, F1 score, recall, and specificity of an average of 95% for each, a system of combined bits of accuracy of 85% tested with a full dataset, and illustrating how AI may enhance cryptanalysis under certain situations by identifying three out of four plain texts from the amalgam of ciphertext and the public key. These results underscore the possible hazards linked to extensive AI systems that may train models with millions to billions of parameters, emphasizing the need for more research into AI's function in cryptanalysis.

1. Introduction

The RSA cryptosystem is the most prevalent method for safeguarding digital communications, mostly due to the challenge of calculating products of big prime numbers, which is fundamental to its mathematical foundation [1]. An asymmetric encryption method permits only those with the correct decryption key to retrieve the original data. This set it as a fundamental element for secure digital communications, financial transactions, and cybersecurity [2]. Nonetheless, advancements in machine learning, particularly in neural networks, are generating new challenges. These improvements have exposed potential vulnerabilities in cryptographic systems, resulting in novel and intriguing opportunities in cryptanalysis [3].

Encryption methods, whether fabricated or altered, fundamentally rely on mathematical concepts. Neural networks have provided adversaries with opportunities to exploit encrypted data patterns. This raises concerns about the robustness of security offered by RSA encryption, particularly when key lengths are quite short. This implies that even a minute of plain text knowledge may enable an attacker to infer values.

The project aims to assess the efficacy of neural networks by conducting plaintext recovery attacks against RSA-encrypted codes. An example of a basic use of this principle is replicated against the RSA-encrypted data with a key length of sixty-six bits and evaluated with additional metrics than the previous study. Through the advancement of AI technologies, researchers are elevating the requirements to prevent the exploitation of newly concealed vulnerabilities inside outdated frameworks.

2. Background

The distinctive feature of public key cryptography is that the key used for encrypting plaintext cannot be used for decryption or retrieval. Consequently, the procedure encompasses both public and private keys [7]. Each receiver holds a pair of keys; as shown by the nomenclature, they give the public key for universal access and encryption, while keeping the private key for decryption of the encrypted content. Although the public key is generated from the private key, it should be challenging to deduce the private keys from the disclosure of public keys, since private keys are used for decryption, regardless of the mathematical distinctions across various algorithms.

RSA is among the most widely used public key encryption schemes. The RSA cryptosystem [1] is widely used and regarded as secure due to the mathematical complexity of prime factorization. In the best situation, given two prime numbers, p and q, compute the public keys as the product of p and q, then derive the private keys from the same prime numbers (see Figure 1). Given that the public key is accessible to everybody, solving its prime factors allows for identifying the matching private key within seconds to minutes. Nonetheless, the RSA depends on the difficulty of figuring out the prime factors of large integers. A practical time for factoring a big composite number composed of two prime factors does not exist [8]. Approximately $2x10^{97}$ primes make up a prime number with 100 digits, necessitating significant effort and processing resources to identify two numbers from such a big integer. For example, a 96-bit public key would need around two years to

 | DOI: 10.20533/jitst.2046.3723.2024.0102

factor, while a 104-bit key would need five centuries to decompose using existing classical binary digit systems by brute force.

Nonetheless, factorizing lengthy composite numbers is not unfeasible but computationally challenging. The only danger foreseen to RSA is the technique introduced by Peter W. Shor in 1994, which effectively decreases the time required to factor numbers into primes from exponential to polynomial time; nevertheless, it theoretically requires a quantum computer to realize its full capabilities, which is anticipated to be achieved soon. Moreover, it is essential to recognize that Shor's algorithm is not the only threat to RSA.

Cryptography mostly concentrates on two principal attacks: the plaintext recovery attack and the key recovery attack. The key recovery attack on the ASCON encryption employs cube-based partial polynomial multiplication to get a 128-bit secret key in 2^{123} time steps. The paper [12] uses a deep neural network for the plaintext recovery attack. ASCON employs symmetric cryptography, using a single key for encryption and decryption, while RSA runs on a distinct principle, using a dual-key scheme.

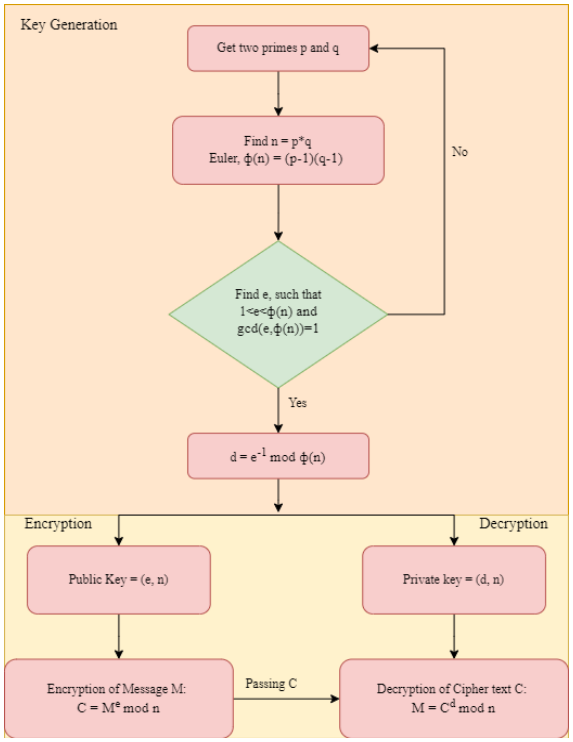

Figure 1. Flowchart of RSA

AI was nascent, making it inadequate for sophisticated cryptography and cryptanalysis for two decades until recently. In contrast to Shor's technique, which is theoretically robust for prospective quantum computing, it may nevertheless proficiently factor modest public keys using the current classical system. Artificial Intelligence (AI) spans several domains, complicating cryptanalysis. Between 1980 and 2000, academics analyzed the MD hash [13], the Data Encryption Standard (DES) [14], a linear cryptanalysis technique, and many minor methods. As technology and hardware advance, AI, particularly in cryptanalysis, intensifies. A significant research article [15], released in 2023, discusses a variant of s-DES using deep neural networks that share a quantum circuit simulator working inside a classical system as one of the layers. This strategy reached a best accuracy of 81% for the key recovery attack. The paper's execution of data preparation is considered substantial, and this study motivates a comparable method. The acquisition of contemporary advanced technologies has not yet contributed to enhancing the RSA cryptanalysis using AI or neural networks. This study is the first novel effort to approach this problem from research.

3. Methodology

A neural network model is constructed to execute a plaintext recovery attack against RSA encryption, with a custom dataset generated from a 66-bit public key and 8-bit plaintext, implemented in the Python programming language.

3.1. Dataset

The self-constructed dataset has nine attributes. It includes 50,000 instances featuring diverse combinations created and generated via the RSA algorithm, encompassing plaintext, ciphertext, decrypted text (identical to plaintext), public key, private key, and two prime numbers used for key pair generation, along with the Euler totient function of ϕ. Each prime number has 33 bits, yielding a 66-bit public key N, while the plaintext is 8 bits in size.

Allocate 70 percent of the dataset, equating to 35,000 out of 50,000 examples, for training, and divide the remaining 30 percent into two equal segments of 15 percent each for validation and testing, amounting to 7,500 instances each.

3.2. Data preprocessing

The challenge with RSA is decrypting the 8-bit plaintext without the private key, relying only on the ciphertext and the public key pair N and e. The RSA method converts N, e, and the encrypted text from the dataset into binary code, including 0s and 1s, and merges them into a singular input array. This yields 137 binary digits, including the zeroes and ones of the 66-bit public key N. The result converts the plaintext value into an 8-bit binary format, giving each 8-bit instance to separate variables within a list of eight. Consequently, each variable has a singular bit that may assume either 0 or 1 values, leading to a bitwise binary classification when the model is trained with a

 | DOI: 10.20533/jitst.2046.3723.2024.0102

designated number of cases.

3.3. Neural Architecture

The hyperparameters and nodes were changed for enhanced training to learn the global minima, using more examples. The input layer has 137 nodes, each corresponding to a binary number, based on the 137 values in the input list.

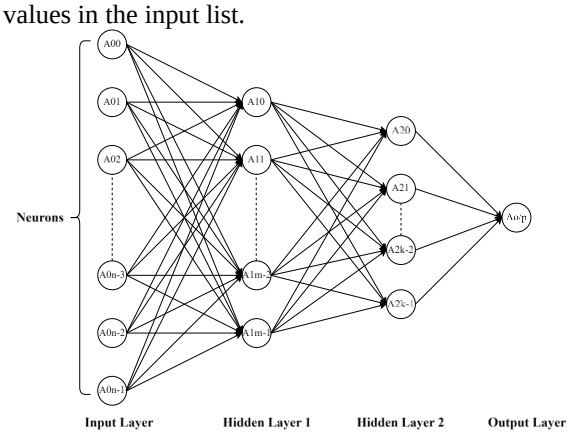

Figure 2. Diagram of a generic classical neural network

Values combine into an Ao/p after traversing the requisite hidden layer(s), yielding the essential output value for model training, as seen in Figure 2. The output layer stands for a binary classification node for a single bit, helping create 8 models for 8 bits. Each model network has 1276 parameters, including both weights and biases. Ultimately, using binary cross-entropy loss and the Adam optimizer enhances performance via modifications to the loss function, activation, and optimization algorithm.

3.4. Evaluation Metrics

Accuracy and Bit-Accurate Probability (BAP) is the number of correct outputs each model can calculate for the given dataset of respective bits. It is calculated as the ratio of the sum of true values to the total values. Bits with a BAP ranging from 50 to 60 percent are considered safe, those from 60 to 80 percent are regarded as normal, and those over 80 percent are classified as susceptible [15].

$$BAP = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is the number of correctly predicted positive values to the total positive prediction. It is calculated as the ratio of true positive to the total positive predictions.

$$Precision = \frac{TP}{TP + FP}$$

Recall is the number of correctly predicted positive values for all the positive cases. It is calculated as the ratio of true positive to the total positive values.

$$Recall = \frac{TP}{TP + FN}$$

F-score (or F1-score) is the harmonic mean measure combining precision and recall into a single metric. It is calculated as twice the ratio of precision and recall products to the sum of precision and recall.

$$Fscore = 2 * \frac{precision * recall}{precision + recall}$$

Specificity is the opposite of precision with negative predictions. It is calculated as the ratio of true negative to the total of negative predictions.

$$Specificity = \frac{TN}{TN + FN}$$

TP, TN, FP, and FN are the True positives, True Negatives, False Positives, and False Negatives, respectively. The explored models are better understood using the stated confusion matrix.

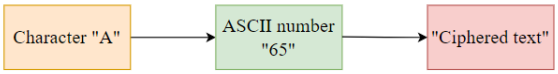

Figure 3. Converting text to ASCII numbers for easy encryption.

The decryption of a small replica of the real-time scenario using the trained models is demonstrated as the final evaluation metric. A random choice of plain text is encoded into its ASCII form, converted to numerical form, and encrypted using the RSA using 66-bit public key N and a maximum of 5-bit public key e, as shown in Figure 3. With the ciphered text and the public keys, decryption is approached and tested with the help of trained models by decoding the predicted ASCII to its character form.

4. Results

After training for 150 epochs with a batch size of 20, the model required 72.41 minutes to complete the training for all the bits. Consequently, the mean duration to complete each segment was around 9 minutes.

The graphs of training and validation accuracies from Figures 3 to 10 show that, for every model when both accuracies grow similarly, all eight models show effective training performance. Furthermore, Table 1 illustrates that the trained models consistently provide at least 92 percent BAP in the validation and testing

 | DOI: 10.20533/jitst.2046.3723.2024.0102

datasets, showing strong predictability for all bits. The overall detection rate for all bits was 95.2 percent for the validation dataset and 95.4 percent for the testing dataset, proving the networks' efficacy in compromisi -ng the RSA cryptosystem.

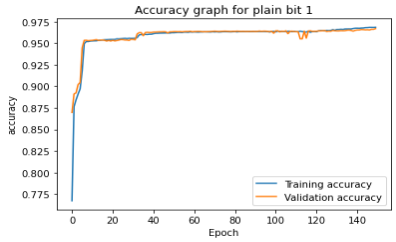

Figure 4. BAP graph of plaintext bit 1

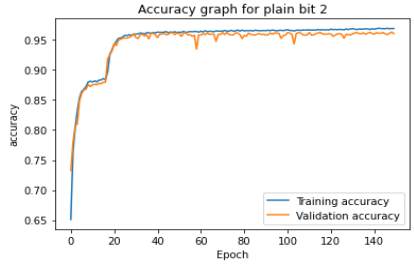

Figure 5. BAP graph of plaintext bit 2

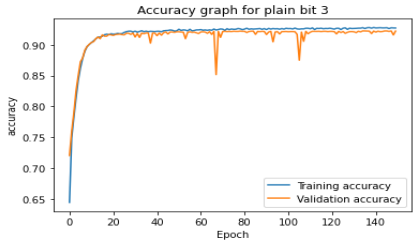

Figure 6. BAP graph of plaintext bit 3

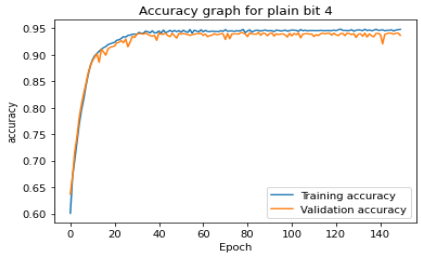

Figure 7. BAP graph of plaintext bit 4

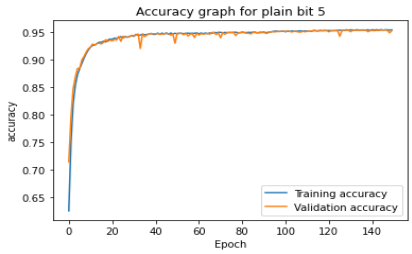

Figure 8. BAP graph of plaintext bit 5

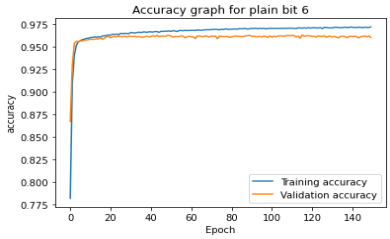

Figure 9. BAP graph of plaintext bit 6

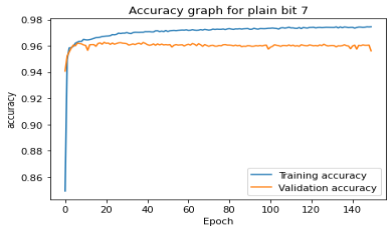

Figure 10. BAP graph of plaintext bit 7

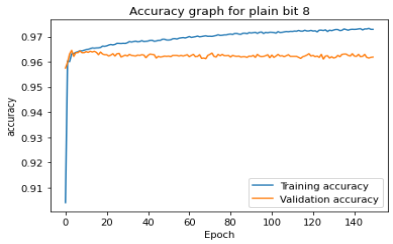

Figure 11. BAP graph of plaintext bit 8

Table 1. Final evaluation of BAP after training the models for 150 epochs each

Binary digits of plaintext	Final accuracy in the Validation Dataset	Accuracy in Testing Dataset
1	96.68	96.48
2	96.04	96.13
3	92.23	92.83
4	93.67	94.27
5	95.33	95.44
6	96.03	96.17
7	95.61	95.75
8	96.19	96.47

Similarly, Table 2, Table 3, and Table 4 are achieved by running the models through the combination of the entire dataset of 50,000 inputs and recording the predicted values.

Table 2. Precision and F1 Score for each of the 8-bit models

Binary digits of plaintext	Precision	F1 Score
1	94.14	96.90
2	96.19	96.65
3	87.42	93.21
4	90.32	94.67
5	99.89	95.25
6	99.67	96.72
7	93.92	96.66
8	94.73	97.11

 | DOI: 10.20533/jitst.2046.3723.2024.0102

Precision and F1-score are derived from 8 models as shown in Table 2 achieving an average of 94.54 percent and 95.90 percent respectively. Likewise, Table 3 shows the values of recall and specificity with a minimum of 89 percent and attaining 99 percent for most of the models.

Table 3. Recall and Specificity for each of the 8-bit models

Binary digits of plaintext	Recall	Specificity
1	99.84	93.69
2	97.13	96.17
3	99.83	95.63
4	99.48	89.40
5	91.03	99.90
6	93.94	99.70
7	99.58	93.61
8	99.63	96.19

Table 4. BAP of each bit and the final accuracy of the convergent metric model

Binary digits of plaintext	Accuracy of each bit	Full Accuracy
1	96.79	84.83
2	96.65	
3	92.73	
4	94.42	
5	95.47	
6	96.84	
7	96.58	
8	97.03	

Table 4 depicts the individual models and convergent system combining all 8 independent models. It shows the BAP of accuracy varied between 92.73 to 97.03 percent of every bit predicted for 50,000 inputs and attained the practical accuracy of 84.83 percent predicted for the combined system.

Figures 12 and 13 illustrate a practical demonstration of decryption using the combined system. We encrypt a small paragraph, chosen randomly from *The Invisible Man*, using four public keys separately, as shown in Figure 12, along with the ciphered text. Among these four key pairs, the trained combined system successfully breaks the encryption for three pairs and derives understandable solutions as shown in Figure 13.

5. Conclusion

Overall, neural network's pattern recognition can be used to study the structure of the RSA method. 137 neurons were employed for the 66-bit key, training with a small-time difference and without using any of Shor's algorithms. The convergence and noise spikes are insignificant to the performance attained because the epochs are fixed. The replica's demonstration of achieving 85 percent accuracy and breaking three out of four ciphered texts with various public keys highlights the necessity to fortify the cryptographic algorithm against actual adversaries offering valuable resources and neural network possibilities in the field of cryptanalysis. This suggests that, along with Shor's algorithm, the AI operating in classical systems poses a risk.

Plain text	The Invisible Man of the title is "Griffin", a scientist who theorizes that if a person's refractive index is changed to exactly that of air and his body does not absorb or reflect light, then he will not be visible. He successfully carries out this procedure on himself, but begins to become mentally unstable as a result...	
	Public Keys	Ciphered text
Encryption	(4667946881334533884 9, 3)	[592704, 1124864, 1030301, 32768, 389017, 1331000, 1643032, 1157625, 1520875, 157625, 941192, 1259712, 1030301,...]
	(3377777759544586290 7, 7)	[29509034655744, 131593177923584, 107213535210701, 34359738368, 11047398519097, 194871710000000, 318547390056832, 140710042265625,
	(5172982401778241006 9, 13)	[15975583671539814980, 13940488994029433292, 42386702377083866402, 36893488147419103232,
	(3846821804908074890 9, 5)	[4182119424, 12166529024, 10510100501, 33554432, 2073071593, 16105100000, 22877577568, 12762815625,...]

Figure 12. illustrates the plain text and demonstrates how it was encrypted with four public keys.

	Decrypted text
(4667946881334533884 9, 3)	ThE InvisiblE Man of thE titlE is "Griffin", a sciEntist who thEorizEs that if a pErson's rEfractivE indEx is changEd to Exactly that of air and his body doEs not absorb or rEflEct light, thEn hE will not bE visiblE. HE succEssfully carriEs out this procEdurE on himsElf, but bEgins to bEcomE mEntally unstablE as a rEsult...
(3377777759544586290 7, 7)	The Invisible Man of the title is "Griffin", a scientist who theorizes that if a person's refractive index is changed to exactly that of air and his body does not absorb or reflect light, then he will not be visible> He successfully carries out this procedure on himself, but begins to become mentally unstable as a result>>>
(5172982401778241006 9, 13)	³³²\x00³óó³ó³³ó²\x00³³ó\x00\x93³\x00ó³²\x00ó³ó ó²\x00³ó\x00³³ñÓ³³³³ó³³ÿ\x00³\x00óó³²óó³óó\x0 0³³\x93\x00ó³²\x93Ó³ó²ó\x00ó³³ó\x00³³\x00³\x0 0³²Óó\x93ó³ó\x00Ó²³Ó³óó³ó²\x00³òò²³\x00³ó\x 00ó³³ó³²ò\x00ó\x93\x00²³³óóó³\x00ó³³ó\x00\x9 3³\x00³³Ó\x00³òò\x00³³ó\x00³\x93ò³\x00ò\x93² ó\x00ó\x93ó\x00³³ó\x93Ó³\x00\x93Ó\x00Ó²³ó ²óó\x00ó³³³óÿ\x00ó³²ó\x00³²\x00³³óó\x00ó\x93 ó\x00³²\x00ó³ó³³ó²ó\x00³²\x00ó³óó²óó³³óó³\x00 ó³ÓÓ³²ó\x00\x93³ó\x00ó³³ó\x00³Ó\x93ó²ò³Ó²\ x00\x93ó\x00³³óó²ó³ÿ\x00³³ó\x00³²³³óó\x00ó\x 93\x00³²ó\x93ó²\x00ó²óó³óó³\x00³óóó³³ó²\x00³ ó\x00³\x00Ó²ó³óóóó'
(3846821804908074890 9, 5)	The Invisible Jan of the title is "Griffin", a scientist who theorizes that if a person's refractive index is chanwed to exactly that of air and his body does not absorb or reflect liwht, then he will not be visible. He successfully carries out this procedure on himself, but bewins to become mentally unstable as a result...

Figure 13. Demonstrates the decryption of the ciphered text given the public keys using the convergent system of neural networks

 | DOI: 10.20533/jitst.2046.3723.2024.0102

6. References

[1] Rivest, R. L., Shamir, A., and Adleman. L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 2 (Feb. 1978), pp. 120–126. DOI: 10.1145/359340.359342.

[2] Menezes, A. J., van Oorschot, P. C., and Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press. DOI: 10.1201/9780429466335.

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

[4] Alani, M. M. (2020). Neural Network Approach to Cryptography: A Survey. International Journal of Computers and Applications, 42(7), pp. 1-6.

[5] Abdelmoumen, M., and Kengne, J. C. (2021). Neural Networks and Cryptanalysis: Advances and Challenges. Journal of Cryptology, 34(4), pp. 1-15.

[6] Rashid, A. B., Kausik A. K. (2024). AI revolutionizing industries worldwide: a comprehensive overview of its diverse applications. Hybrid Advances, Volume 7, DOI: 10.1016/ j.hybadv.2024.100277.

[7] Diffie, W. (1988). The first ten years of public-key cryptography. in Proceedings of the IEEE, vol. 76, no. 5, pp. 560-577, May, DOI: 10.1109/5.4442.

[8] Rowland, H. (2016). The Role Of Prime Numbers in RSA Cryptosystems.

[9] Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, pp. 124-134, DOI: 10.1109/SFCS.1994.365700

[10] Kandi, A., Baksi, A., Gerlich, T., Guilley, S., Gan, P., Breier, J., Chattopadhyay, A., Shrivastwa, R. R., Martinásek Z., Bhasin S. (2023). Hardware Implementation of ASCON. In NIST LWC Workshop 2023.

[11] Rohit, R., Hu, K., Sarkar, S., Sun, S. (2021). Misuse-free key-recovery and distinguishing attacks on 7-round ascon. Cryptology Archive, Paper 2021/194. Cryptology ePrint Archive. https://eprint.iacr.org/2021/194 (Access Date: 18 December 2024).

[12] Jankovikj, D., Mihajloska Trpceska, H., and Dimitrova, V. (2022). Cryptanalysis of Round-Reduced ASCON powered by ML.

[13] Dobbertin, H. (1996). Cryptanalysis of MD4. In: Gollmann, D. (eds) Fast Software Encryption. FSE 1996. Lecture Notes in Computer Science, vol 1039. Springer, Berlin, Heidelberg. DOI: 10.1007/3-540-60865-6_43

[14] Matsui, M. (1994). The First Experimental Cryptanalysis of the Data Encryption Standard. In: Desmedt, Y.G. (eds) Advances in Cryptology — CRYPTO ’94. CRYPTO 1994. Lecture Notes in Computer Science, vol 839. Springer, Berlin, Heidelberg. DOI: 10.1007/3-540-48658-5_1.

[15] Kim, H., Lim, S., Baksi, A., Kim, D., Yoon, S., Jang, K., Seo, H. (2023). Quantum Artificial Intelligence on Cryptanalysis. Cryptology Archive, Paper 2023/004. Cryptology ePrint Archive. https://eprint.iacr.org/202 3/004 (Access Date: 21 December 2024).

