Enhancing the RC4 Algorithm by Eliminating the Initiative Vector (IV) Transmission

Waleed Abdelrahman Yousif Mohammed
Faculty of Computer Studies and Information Technology, Nile University, Khartoum, Sudan
waleed_i66@hotmail.com

Salmah Fattah
Cybersecurity Research Lab, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, Kota Kinabalu, Malaysia
salmahf@ums.edu.my (corresponding author)

Khalid Mohammed Osman Saeed
Faculty of Computer Science and Information Technology, Omdurman Islamic University, Sudan
khalidmohosman@gmail.com

Ashraf Osman Ibrahim
Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia | Positive Computing Research Center, Emerging & Digital Technologies Institute, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia
ashraf@utp.edu.my

Safaa Eltahier
Software Engineering Department, College of Computer Engineering and Science, Prince Sattam Bin Abdulaziz University, KSA | Software Engineering Department, College of Computer Science and Information Technology, Sudan University of Science and Technology, Khartoum, Sudan
s.eltaher@psau.edu.sa

Received: 8 October 2024 | Revised: 19 November 2024 | Accepted: 23 November 2024

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: https://doi.org/10.48084/etasr.9208

ABSTRACT

The Rivest Cipher RC4 encryption algorithm is commonly utilized to generate keys of varying lengths. Despite its rapid processing speed, vulnerabilities within the algorithm have made it susceptible to exploitation, allowing attackers to compromise it within a matter of minutes. This paper introduces an innovative approach to address the vulnerabilities of the RC4 encryption algorithm by employing an Initiative Vector (IV). The proposed method incorporates a lengthy random text without transmitting an initialization vector. The proposed solution was rigorously validated, demonstrating performance comparable to existing solutions while simultaneously expanding the range of potential solutions and mitigating security threats. Further exploration into the use of a complex equation is recommended for calculating the swapping value *j* while maintaining the same high level of performance.

Keywords-encryption algorithm; file protection; cryptography; RC4; Initiative Vector (IV)

I. INTRODUCTION

The RC4 encryption algorithm is widely known for generating key streams of various lengths, primarily for encryption purposes. It is part of the RC cipher family, a collection of encryption algorithms developed by Ronald L. Rivest. Although most algorithms within this family rely on block encryption techniques, RC4 stands out for its use of the stream technique. The RC cipher family includes six iterations, the most prominent being the latest three versions: RC4, RC5, and RC6. Despite the speed of RC4, it suffers from multiple vulnerabilities that can lead to its compromise in minutes [1]. As reported in [2], the principles behind these attacks depend on the operations of the RC4 algorithm. Various security standards widely employ RC4, including Wired Equivalent Privacy (WEP), WPA wireless security, and Transport Layer

Security (TLS). In all these cases, it derives session keys from master keys. The primary objective of this study is to introduce a solution to address the vulnerabilities inherent in the RC4 encryption algorithm by eliminating the need to transmit the Initiative Vector (IV). This paper aims to validate the effectiveness of the proposed solution, ensuring that it does not adversely affect the algorithm's performance while doubling its security level compared to existing solutions. Furthermore, more research is needed to enhance the proposed solution by incorporating complex equations to calculate the swapping value while maintaining the current level of performance. This study contributes significantly to strengthening the security of RC4 encryption and advancing the field of cryptography. Its detailed contributions can be summarized as follows:

- Proposes a new solution to address the weaknesses of the RC4 encryption algorithm. The proposed solution involves using long random text and not sending the IV.
- Maintains the same level of performance as existing solutions while doubling the range of possible solutions and reducing the threat of attacks.

In response to the absence of RC1, the regulatory authorities directed their efforts toward managing a sequence of symmetric keys, collectively recognized as the Rivest Cipher algorithms [3]. Alongside, RC2 was introduced, a block encryption algorithm designed to accommodate variable key lengths. RC2, an enhancement over RC1, incorporated internal operations such as XORing and mathematical functions to improve its cryptographic capabilities. However, it remained susceptible to vulnerabilities, including differential and linear cryptanalysis [4]. RC3 was introduced but unfortunately faced a significant setback as it was compromised before any practical deployment [5].

The IV introduces dynamism to the key stream but has notable weaknesses. Its size is limited to 24 bits, is transmitted in plaintext with each packet, and is often reused, creating vulnerabilities. Additionally, there is no specification on how the IV should be chosen or how frequently it should change. On the other hand, RC5, a symmetric block cipher, processes fixed-length bit sequences (blocks) and is optimized for both hardware and software due to its word-oriented design. This adaptability allows RC5 to efficiently utilize processors of varying word lengths [6]. RC6 is a block cipher that uses 128-bit block size and supports key sizes of 128, 192, and 256 bits. It is an enhancement of the RC5 algorithm and is designed to meet the Advanced Encryption Standard (AES) requirements. It offers a better level of security against attacks, which may be possible in RC5, which means it is more secure. It is also protected from various other possible security attacks, uses fewer rounds, and offers higher throughput [7].

The RC4 algorithm is widely applied because of its simplicity and real-time capabilities. However, it has its weaknesses: the utilization of the IV. Previous studies have shown that the IV can be used for attacks, including the FMS attack that targets the Key Scheduling Algorithm (KSA) [8]. IV reuse and insufficient randomness undermine the robustness of the algorithm and make it possible to extract encryption keys [4]. In efforts to improve RC4, it is common to see attempts to reduce IV-based susceptibility. For instance, features such as generating dynamic keys or even complete withdrawal of IV have been proposed to address the key stream predictability of RC4 [9]. These enhancements should retain the speed and simplicity of RC4, which is a major asset while eliminating its major vulnerabilities. In the case of HIV, the complete removal of IV transmission depends on safe key management and randomization. This approach eliminates the exposure of sensitive cryptographical elements and makes the encryption process less vulnerable to known IV-related attacks.

In [10], the five best-known variants of RC4 were experimentally compared and classified into two groups according to the presence or absence of such a weakness. The study in [11] fills the gaps of the RC4 stream cipher and discusses the aspects of keying and correlation output, proposing three enhancements. There are a few variations of RC4, such as RRC4, which modifies a random initial state to solve a weak key problem, RC4-2S uses two state tables with permutation that effectively decrease output relationships while improving speed, and RC4-2S+ generates four keys per cycle to increase randomness, security, and speed. In [12], the stream cipher RC4 was modified with an optimized pseudorandom bit generator, improving the weaknesses of the RC4. This combined design can enhance inter-state, resist the main attacks, provide greater randomness, and provide higher performance compared to RC4.

In [13], it was found that the RC4 stream cipher has statistical flaws, especially in the relation between the inputs and the outputs, applying two common tests known as strict avalanche and bit-independence tests. Five widely discussed RC4 variants were experimentally tested and then divided into groups according to the existence of these shortcomings. In [14], an improved, high-speed, lowest latency RC4 encryption algorithm was developed for the 14-nm SMIC process. This design is effective since it achieves four S-box swaps per clock cycle through combinational logic, executes key scheduling algorithms in 64 cycles, and produces 4 bytes of keystream per cycle. It implements a compact countermeasure against Differential Power Analysis (DPA) attacks, operates at 1 GHz, has 32 Gbps throughput and 74 cycle latencies, and occupies 51,596 μm². In [15], a modified RC4 algorithm was presented to enhance security using four state tables of key-stream generator along with key control for the random selection. Forward and backward effects are used to enhance randomness concerning encryption and decryption. The evaluation results showed that the proposed algorithm was relatively more complex than general RC4 and passed most of the NIST randomness using different key sizes.

In [16], RC4 was used to effectively protect the authenticity of the data, preventing unauthorized changes by individuals. In [17], a new algorithm, called MRC4, used the SRFG method for operation to counteract some of the shortcomings in the normal RC4 cipher stream. MRC4 enhances security characteristics, such as nonlinearity, resistance, and confusion and diffusion balance characteristics, achieving 60% better confusion and 50% better diffusion. In [18], a hybrid encryption model was suggested, merging ECC, RC4, and SHA-256 to enhance data security in IoT-based smart irrigation systems, achieving significantly shorter encryption and decryption times compared to other methods. In [19], an overview of IoT security threats, attacks, and solutions was presented, along with prospects to connect IoT nodes and layers as per IoT connectivity, communication, and management layers. This study underlines calls to harmonize the efforts of international specialists to develop a set of rigorous security norms and, in turn, sets security requirements as a reference for evaluating IoT solutions.

The proposed solution differs from others in avoiding sending the IV and merging its field from the MAC frame. This approach enhances the security of the RC4 encryption algorithm while maintaining its performance and provides a unique solution to address its weaknesses.

II. ATTACK ON RC4

Continuous wireless network messages with a substantial volume of transmission pose a challenge. Encryption cannot be achieved effectively with a limited set of shared keys, necessitating the generation of a long key stream. To address this issue, the shared key is input into RC4, as shown in Figure 1, to produce extended key streams. Nevertheless, vulnerability arises as the same key stream is consistently generated, rendering it susceptible to exploitation. Therefore, the incorporation of an IV into RC4 provides a solution, allowing cryptography experts to infer a portion of the stream and consequently expose a fragment of the shared key.

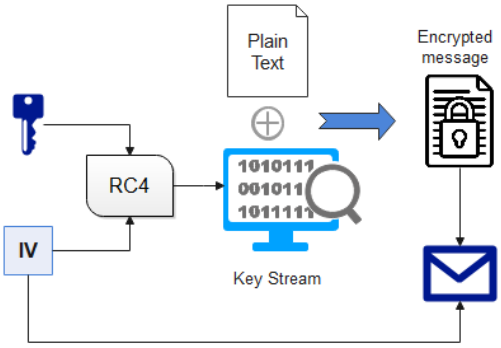

Fig. 1. Message encryption process.

As a starting point, an intriguing attack on the WEP protocol's RC4 key was proposed based on unique pattern IVs. As a result, motivated by the FMS attack, in [20], new IV patterns were identified that could be applied to WEP to recover the RC4 key. However, it was soon realized that the new pattern could also be applied to the FMS attack. As a result, combining the new pattern may increase the number of IVs that can be used in the FMS attack. The FMS attack extracts the initial byte of the key stream, which is based on RC4 operation, using a brute force attack. Success rates increase with traffic and exceed a few seconds in duration. The first-round attack is relatively straightforward but successful. Assume that you can mimic the first phase of the RC4-KSA by knowing $K[0]\colon K[L-1]$ and wanting to recover $K[L]$. The value $k = Sl[jl + Sl[l] + K[l]]$ is then changed to $Sl + 1[l]$ in the following step. Thus, just as in the FMS attack or Mantin's second-round attack, knowledge of k would reveal $K[L]$. Therefore, before retaking the value L, wait until exactly $n-2$ of the remaining RC4-KSA $(n-L-1)$ steps have been completed and the RC4-PRGA $(L-1)$ steps have been completed.

III. AMENDMENT OF RC4

The common usage of RC4 is WEP security. Figure 2 shows the WEP operations. The original RC4 operations are performed as in (1) to calculate the j value, which is used to swap with the value pointed by i. The attacker uses this equation to find the j value every time.

$$(j = (j + S[i] + T[i]) mod\ 256) \tag{1}$$

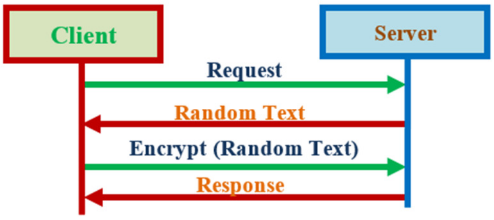

Fig. 2. WEP security operations.

The solution uses the exact steps of RC4, including the equation of the old version of RC4, to calculate the j value, but there are some differences. One of them is that the IV is not sent obviously as plain text. Second, long random text is used instead of short ones in old RC4. The IV is not generated on the requested side but is selected from the random text received from the second side (Access Point-AP) using the password as follows.

A. On the First Side (Client)

1. The client receives random text from the AP.
2. The user enters the password.
3. Convert the first two bytes from the password to an integer (INT).
4. Use the integer number (INT) in step 3 as a start point or pointer to get N bytes from the random text received in the first step, as shown in Figure 3.
5. Pass the N bytes selected from random text to the RC4 algorithm to use as IV to generate the session key.
6. End.

B. Second Side (Server)

The second side can complete the operation because it has all the factors used to do so. The elements are random text and a pre-shared password, then can get the same IV. For example, suppose the first two bytes of the master key after being converted to an integer number is 3742. In that case, this number is used as a pointer to get the IV value. Starting from this point of random text, which is received from the second side, this method can select 4 or 6 bytes.

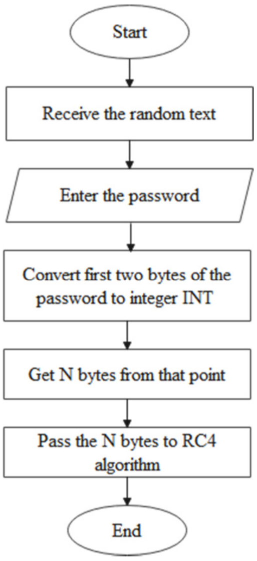
Start
Receive the random text
Enter the password
Convert first two bytes of the password to integer INT
Get N bytes from that point
Pass the N bytes to RC4 algorithm
End

Fig. 3. Proposed RC4 method.

C. *Security Testing*

The proposed modified RC4 was implemented in C++ and the resulting outputs of both the original and modified RC4 were subsequently subjected to analysis using the Abel Cain Tool to quantify the time required for their respective decryption processes. Tables I and II present the findings, showing that the proposed RC4 exhibited robustness by successfully detecting the original password without using an IV. In contrast, the original RC4's password was swiftly deciphered within a few seconds, although the precise duration was not measured (indicated by a small value). Notably, the enhanced RC4, despite consuming more than an hour, did not identify or produce any output during this timeframe.

TABLE I. RESULTS OF NEW RC4

No	Output of New Algorithm	Detection	Password
A	09B2234C16645525	Not Detected	,gd] 123#
B	42D72148E851F260	Not Detected	Uf]hgvlk
C	7E231231A2123718	Not Detected	12345678
D	9255419E32085220	Not Detected	Wal123!9
E	1B64727E08831112	Not Detected	D,stlpl]

TABLE II. RESULTS OF OLD RC4

No	Output of Old RC4	Detection	Password
A	52B2456937004393	Detected	,gd] 123#
B	21B5659DF0F11E47	Detected	Uf]hgvlk
C	F0343164F6C2E373	Detected	12345678
D	BF1448390E7D9CBF	Detected	Wal123!9
E	1F292E7FD2024C52	Detected	D,stlpl]

D. *Performance Analysis*

The evaluation of algorithm performance is a critical aspect, and in this case, it focuses on the key length, comparing both the new and old versions. The results shown in Figure 4 underscore that performance is intricately linked to key length. Despite introducing a solution that increases both the key length and file size, the impact on performance is minimal, showcasing the robustness of the encryption algorithm. Key length serves as a pivotal factor influencing the efficiency of cryptographic processes. The analysis reveals that, while modifications have been introduced in the proposed algorithm, the performance remains comparable to the old version. This consistency implies that the algorithm's efficacy is resilient even when subjected to changes in key length and file size. The results highlight the subtle connection between key characteristics and the performance of the algorithm. The ability to enhance key length and file size without significantly compromising performance indicates the algorithm's adaptability and effectiveness. This performance analysis contributes valuable insights into the algorithm's behavior under varying conditions, which is essential for ensuring the reliability and versatility of cryptographic systems.

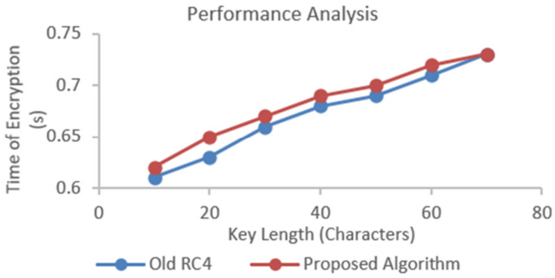

Fig. 4. Performance analysis.

E. *Entropy Analysis*

Tables I, II, and III provide the detailed results of the entropy analysis performed on the cipher texts of both the old and new RC4 versions using CrypTool. Table III explicitly outlines the entropy test results, revealing that the old RC4 produces an entropy value of 2.41 from an expected 4.7. Notably, the proposed modified RC4 also attains an entropy value of 2.41 across all tables. This uniformity in entropy values implies that both the original and proposed RC4 versions exhibit comparable levels of security and efficiency. The consistency in entropy metrics underscores the robustness of the cryptographic properties in both iterations of the RC4 algorithm.

TABLE III. ENTROPY TEST RESULTS

Cipher text	Results
Old RC4	2.41
New RC4	2.41

F. *Frequency Analysis*

Frequency analysis is a technique used in cryptography to analyze symbol distribution in a text, often represented through histograms. In the context of the RC4 encryption algorithm, Figure 5 illustrates the symbol distribution of the old RC4 version in hexadecimal format. Each bar in the histogram corresponds to a symbol, showing its frequency. This visual representation provides insights into the concentration or dispersion of symbols in the encrypted output. Figure 6 shows the histogram for the proposed RC4 algorithm. Comparing the histograms of the old and new versions allows analysts to identify variations, potential improvements, or alterations in the new version.

Comparing the original RC4 with the proposed RC4 can be well demonstrated in the distribution of the symbols. The original RC4 algorithm has not only key stream patterns or biases in terms of symbol occurrence and distribution because of flaws in the key-scheduling and pseudo-random number generation stages, but it is also easily vulnerable to a frequency analysis attack. The proposed RC4 algorithm solves these problems by providing a more random and balanced distribution improving security and unpredictability. These enhancements enhance the cryptographic immunity of the modified algorithm.

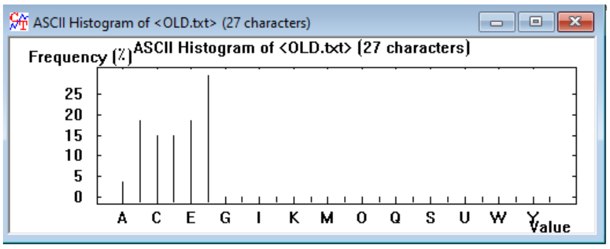
ASCII Histogram of <OLD.txt> (27 characters)
Frequency (%)
ASCII Histogram of <OLD.txt> (27 characters)
25
20
15
10
5
0
A C E G I K M O Q S U W Y
Value

Fig. 5. Frame format of old WEP.

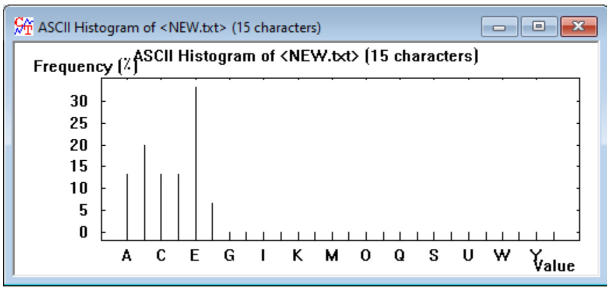
ASCII Histogram of <NEW.txt> (15 characters)
Frequency (%)
ASCII Histogram of <NEW.txt> (15 characters)
30
25
20
15
10
5
0
A C E G I K M O Q S U W Y
Value

Fig. 6. Frame format of new WEP.

Changes in the histogram may indicate modifications that affect the randomness or uniformity of the encrypted output. Well-designed encryption algorithms, such as RC4, aim to produce cipher text that appears random, resisting cryptanalysis. Histogram analysis assesses whether the algorithm achieves this by visually representing symbol frequencies. Researchers and cryptographers use frequency analysis to evaluate the security and performance of the encryption algorithm. Deviations from a uniform distribution in the histogram may indicate vulnerabilities exploitable by attackers, emphasizing the importance of thoroughly examining the histogram, as seen in Figures 7 and 8, to understand RC4's cryptographic properties. Peaks in the histogram reveal predictability, signaling vulnerabilities. Improving randomness ensures uniform distribution, enhancing encryption security.

IV. RESULTS AND DISCUSSION

By utilizing longer random text and avoiding sending the IV, the proposed solution increases the complexity of the encryption process. This adjustment effectively transforms the password into a dynamic element and mitigates vulnerabilities in the previous version of RC4. The main modification is eliminating sending the IV. This enhancement not only doubles the length but also substantially augments the derived key size from RC4 and consequently prolongs the time required to crack the encryption. The second alteration was done on the equation to calculate the j value. The equation had to be computed from 16 bits, which made it more challenging to compute. The equation was slightly modified, with the divisor value changing to 512 instead of 256 on old RC4, as:

$$j = (j + S[i] + T[i]) \; mod \; 512 \qquad (2)$$

where i and j refer to the indexing of the swapping function, the last adjustment, and more complexity. The length of the random text was duplicated. This adjustment enhances security by making it difficult for potential attackers to predict or calculate the swapping value. Consequently, the proposed solution outperforms the traditional RC4 algorithm in terms of encryption security. In addition, changes are reflected in the frame format used by WEP technology by merging the IV field to return of origin custom, as illustrated in Figures 7 and 8.

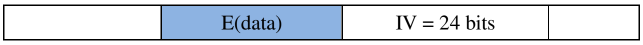

Fig. 7. Frame Format of old WEP.

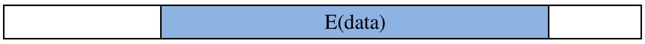

Fig. 8. Frame format of new WEP.

The results present a comprehensive analysis of the proposed solution to address the weaknesses of the widely used RC4 encryption algorithm. The focus of this study was to enhance the security of RC4 by employing an IV approach that involves using longer random text and eliminating IV sending. The study findings reveal several key points of significance that require further discussion and analysis.

In the previous version, crackers replay on IV that is sent as plain text, and the time taken to find the origin password was a few minutes. However, in the proposed solution, the time required to find the origin password increased, depending on the brute-force attack calculation when using a key length of 512 bits. From the perspective of strengthening password security, this study shows considerable improvements in the IV generation process. A comparative analysis in Table IV highlights the substantial enhancements in the proposed solution, as it significantly improves IV length, random text size, and key length. Unlike the traditional RC4, which sent the IV in plain text, the proposed solution does not send the IV at all, addressing a major security vulnerability. These enhancements collectively strengthen encryption without sacrificing performance, as indicated by the comparative data. Comparing the proposed method with the traditional RC4, the improvements are substantial.

TABLE IV. COMPARISON OF TRADITIONAL RC4 AND PROPOSED SOLUTION

Properties	Proposed algorithm	Traditional RC4
IV Length	8 Bytes (64 bits)	3 Bytes (24 bits)
Random Text	4 KB	2 KB
IV sending	Not sent	Send clear as plain text
Key length	64 - 128 - 256 – 512	64 – 128 – 256

The IV length increased to 48 bits (6 Bytes) from the previous 24 bits (3 Bytes), strengthening its security. The random text size is also doubled, adding an extra layer of complexity. Moreover, the proposed solution eliminates the sending of IV in plaintext, further enhancing security compared to the traditional method. In addition, the key length in the proposed algorithm varies from 64 to 512, providing a wider range of options for key security. These modifications collectively result in a more secure and robust encryption mechanism, as demonstrated in the updated frame format for WEP. The new frame format consolidates the IV field with the return of the origin custom, minimizing vulnerabilities present in the old format.

This study implemented two essential modifications to enhance security and prolong password longevity. The first modification involves the following:

- IV concealment: This modification reduces the information available to potential attackers, making it more difficult for them to detect the original password.
- Reduced information exposure: By not providing the attacker with crucial information, the proposed system enhances security, ultimately leading to an increase in the time required to crack passwords compared to traditional RC4.
- Increased crack time: The result of the above modifications is a notable increase in cracking time, particularly when subjected to brute-force attack techniques. The password age is extended to several years, bolstering resistance against unauthorized access.
- Improved user experience: This approach benefits users who need permission, especially when they need to be changed infrequently. Extending the password age caters to users who prefer to keep the same password for an extended period.
- Custom frame format: Another advantage of the modifications is the introduction of a custom frame format for data transmission, enhancing the overall security by returning to the origin in a controlled manner. The format adds an extra layer of protection against potential vulnerabilities.

The second modification, entailing the introduction of an equation to calculate the swapping value j, increases the computational complexity for determining j, expands the key length to 512 bits, and doubles the amount of random text. The study conducted additional measures to validate the efficiency and effectiveness of the proposed solution.

The proposed solution shows improved security of the RC4 encryption algorithm by using a long random text and not sending the IV. This solution increases the security of RC4 while maintaining the same performance and does not require significant changes to the algorithm. However, the proposed method has some advantages and disadvantages.

A. Advantages

- The proposed solution addresses weaknesses in the RC4 encryption algorithm by refraining from sending the IV and employing an extended random text.
- The security level of the RC4 algorithm is augmented twofold compared to existing solutions while maintaining consistent performance. The proposed solution achieves enhanced security without significantly altering the underlying algorithm.
- The primary objective of augmenting password dynamism and security is to resist dictionary attacks, brute-force attacks, and guessing attacks. Previous versions often exhibited vulnerabilities in their appended clear message IV transmission. To overcome this challenge, the proposed method doubles the IV length to 48 bits (6 Bytes), adding an intensified layer of difficulty to the prediction or computation process.

B. Disadvantages

- The proposed solution might require additional computation to calculate the swapping value, potentially leading to slower processing speeds on specific devices.
- Differentiating from existing systems tailored for the traditional RC4 algorithm, the distinct approach of the proposed solution can limit its adoption.

V. CONCLUSIONS

This study introduces a precise technique that incorporates improvements into the RC4 encryption algorithm in a manner that up until now has rendered it vulnerable to rapid attack vulnerability. By implementing an IV approach without IV transmission and retaining all of RC4's advantages in terms of performance, the proposed method effectively enhances the algorithm's resistance to traditional attacks such as FMS and Klein attacks. The proposed approach adds a new parameter of a random lengthy text into the key generation process, which contributes not only to the enlargement of the solution space but also to the enhancement of security as compared to the existing RC4 applications. Various validation tests showed that the proposed approach is as fast as other typical solutions. In particular, this work enhances the security model of the RC4 algorithm by proposing a further study of elaborate equations for the calculation of the swap value j to further complicate and randomize swapping without hindrance on computational speed. Future work will employ more complicated equations to improve the security of RC4 and comprehensively examine its adaptability and security through various experiments. Such techniques can be applied to cloud image security [21].

REFERENCES

[1] M. Safaei Pour, C. Nader, K. Friday, and E. Bou-Harb, "A Comprehensive Survey of Recent Internet Measurement Techniques for Cyber Security," *Computers & Security*, vol. 128, May 2023, Art. no. 103123, https://doi.org/10.1016/j.cose.2023.103123.

[2] H. N. Thakur, A. Al Hayajneh, K. Thakur, A. Kamruzzaman, and M. L. Ali, "A Comprehensive Review of Wireless Security Protocols and Encryption Applications," in *2023 IEEE World AI IoT Congress (AIIoT)*, Seattle, WA, USA, Jun. 2023, pp. 373–379, https://doi.org/10.1109/AIIoT58121.2023.10174571.

[3] C. Slamet, U. Syaripudin, F. M. Kaffah, and B. E. Tiasto, "Implementation of Rivest Cypher 4 algorithm in Security Assertion Mark-up Language protocols on Single Sign-On services," *IOP Conference Series: Materials Science and Engineering*, vol. 1098, no. 3, Mar. 2021, Art. no. 032109, https://doi.org/10.1088/1757-899X/1098/3/032109.

[4] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. New York: John Wiley & Sons Inc, 1996.

[5] M. A. Alrammahi and H. Kaur, "Development of Advanced Encryption Standard (AES) Cryptography Algorithm for Wi-Fi Security Protocol," *International Journal of Advanced Research in Computer Science*, vol. 5, no. 3, pp. 62–67, 2014.

[6] R. L. Rivest, "The RC5 encryption algorithm," in *Fast Software Encryption*, Leuven, Belgium, 1995, pp. 86–96, https://doi.org/10.1007/3-540-60590-8_7.

[7] K. Aggarwal, J. Kaur Saini, and H. K. Verma, "Performance Evaluation of RC6, Blowfish, DES, IDEA, CAST-128 Block Ciphers," *International Journal of Computer Applications*, vol. 68, no. 25, pp. 10–16, Apr. 2013, https://doi.org/10.5120/11749-7244.

[8] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," in *Selected Areas in Cryptography*, Toronto, Canada, 2001, pp. 1–24, https://doi.org/10.1007/3-540-45537-X_1.

[9] S. Paul and B. Preneel, "Analysis of Non-fortuitous Predictive States of the RC4 Keystream Generator," in *Progress in Cryptology - INDOCRYPT 2003*, New Delhi, India, 2003, pp. 52–67, https://doi.org/10.1007/978-3-540-24582-7_4.

[10] G. Paul and S. Maitra, *RC4 Stream Cipher and Its Variants*. CRC Press, 2011.

[11] M. M. Hammood, K. Yoshigoe, and A. M. Sagheer, "Enhancing security and speed of RC4," *International Journal of Computing and Network Technology*, vol. 3, no. 2, 2015.

[12] J. Xie and X. Pan, "An improved RC4 stream cipher," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, China, Oct. 2010, pp. V7-156-V7-159, https://doi.org/10.1109/ICCASM.2010.5620800.

[13] E. J. Madarro-Capó, C. M. Legón-Pérez, O. Rojas, and G. Sosa-Gómez, "Measuring Avalanche Properties on RC4 Stream Cipher Variants," *Applied Sciences*, vol. 11, no. 20, Oct. 2021, Art. no. 9646, https://doi.org/10.3390/app11209646.

[14] C. Sun *et al.*, "A high-speed and low-latency hardware implementation of RC4 cryptographic algorithm," *International Journal of Circuit Theory and Applications*, vol. 51, no. 12, pp. 5980–5996, 2023, https://doi.org/10.1002/cta.3769.

[15] S. Kareem and A. M. Rahma, "A Modification on Key Stream Generator for RC4 Algorithm," *Engineering and Technology Journal*, vol. 38, no. 2B, pp. 54–60, Jul. 2020, https://doi.org/10.30684/etj.v38i2B.404.

[16] A. F. Doni, O. A. H. Maria, and S. Hanif, "Implementation of RC4 Cryptography Algorithm for Data File Security," *Journal of Physics: Conference Series*, vol. 1569, no. 2, Jul. 2020, Art. no. 022080, https://doi.org/10.1088/1742-6596/1569/2/022080.

[17] R. Saha, G. Geetha, G. Kumar, T.-H. Kim, and W. J. Buchanan, "MRC4: A Modified RC4 Algorithm Using Symmetric Random Function Generator for Improved Cryptographic Features," *IEEE Access*, vol. 7, pp. 172045–172054, 2019, https://doi.org/10.1109/ACCESS.2019.2956160.

[18] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, "Security of Internet of Things using RC4 and ECC Algorithms (Case Study: Smart Irrigation Systems)," *Wireless Personal Communications*, vol. 116, no. 3, pp. 1713–1742, Feb. 2021, https://doi.org/10.1007/s11277-020-07758-5.

[19] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, "A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review," *Sensors*, vol. 23, no. 8, Jan. 2023, Art. no. 4117, https://doi.org/10.3390/s23084117.

[20] T. Guo, Y. Feng, and Y. Fu, "A new form of initialization vectors in the FMS attack of RC4 in WEP," *Procedia Computer Science*, vol. 183, pp. 456–461, Jan. 2021, https://doi.org/10.1016/j.procs.2021.02.084.

[21] Z. A. Mohammed, H. Q. Gheni, Z. J. Hussein, and A. K. M. Al-Qurabat, "Advancing Cloud Image Security via AES Algorithm Enhancement Techniques," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12694–12701, Feb. 2024, https://doi.org/10.48084/etasr.6601.