Discovering Unknown Non-Consecutive Double Byte Biases in RC4 Stream Cipher Algorithm

Sura Mahroos[*1], Rihab Hazim [2], AbdulRahman Kareem Oliwe [3], Nadia Mohammed[4], Yaqeen Saad [5], Ali Makki[6], Ibrahiem El Emary[7]

[1,2,5,6]University Of Anbar, College Of Computer Sciences and Information Technology,
Anbar, Ramadi, 31001, IRAQ
[3] University Of Anbar, Center for Continuing Education,
Anbar, Ramadi, 31001, IRAQ
[4] University Of Anbar, College Of Islamic Sciences,
Anbar, Ramadi, 31001, IRAQ
[7] King Abdulaziz University, KAU, Jeddah, 22233, SAUDI ARABIA
Emails: surasms917@uoanbar.edu.iq ; rehz1991@uoanbar.edu.iq ; tim25112019@gmail.com; nadia.fahad@uoanbar.edu.iq; yaqeen.cs91@uoanbar.edu.iq; ali_makki@uoanbar.edu.iq; omary57@hotmail.com

Abstract

RC4 is one of the most widely used stream cipher algorithms. It is fast, easy and suitable for hardware and software. It is used in various applications, but it has a weakness in the distribution of generated key bytes. The first few bytes of Pseudo-Random Generation Algorithm (PRGA) key stream are biased or attached to some private key bytes and thus the analysis of key stream bytes makes it potential to attack RC4, and there is connection between the key stream bytes that make it weak and breakable by single- and double-byte biases attack. This work shows the analysis of RC4 key stream based on its non-consecutive double byte biases by using newly designed algorithm that calculates the bias in a standard time (seconds). The results are shown that the bias of RC4 keystream is proved and got the same results that were shown in the literature with less time and discover a set of new non-consecutive double byte biases in the positions (i) and (i+n). The analysis of 256 positions is required additional requirements such as supercomputer and the message passing interface environment that are not available in Iraq, therefore; the analysis is done for 32 positions.

Keywords: RC4; Key Scheduling Algorithm (KSA); PRGA; Double Byte Bias; Non-Consecutive Double Byte Bias.

1. Introduction

Information security is an approach that an organization secures and conserve its systems [1]. Information can be conserved by encrypting it by using one of encryption algorithms. Many factors are needed to take into accounts such as security, time complexity, space complexity, and the characteristics of the algorithm. The main aim of the cryptography is not only used to provide secrecy, but also to provide solutions to other problems such as authentication, integrity, availability, and non-repudiation [2]. Various encryption algorithms are vastly used in wired networks. In symmetric encryption, when the key size is small, it must be very efficient and encryption time can be faster. Various encryption ways that used in wireless devices are dependent on symmetric encryption, such as RC4 cipher [3]. RC4 is an effective stream cipher, and it is more prevalent. It is one of significant encryption algorithms. Ron Rivest proposes it in 1987 and it is called "Ron's Code 4". It has depended on the use of random permutation [4] and was kept private in a trade until 1994. It is used in commercial software packages such as Oracle Secure SQL, MS Office. And used in network protocols such as IP Sec, Wired Equivalent Privacy (WEP) Protocol [5] and used to conserve wireless networks as a portion of

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024

Wi-Fi Protected Access (WPA) protocols and to conserve the internet traffic as a portion of Transport Layer Security (TLS) protocol and Secure Socket Layer (SSL) protocol [6]. Various people analysed RC4, and different weaknesses were discovered [7]. Mantin and Shamir [8] and Fluhrer [9] described the attack on this algorithm.

Technically, this algorithm consists of two algorithms: (KSA) and a pseudo-random generation algorithm (PRGA). The first algorithm (KSA) takes as input a key K, typically a byte array of length between 5 and 32 bytes (40 to 256 bits), and generates the initial state (S) where S is the canonical representation of a permutation on the set [10].

The main contributions of this work are:

- Designing a new, efficient, and fast algorithm to analyse the RC4 algorithm based on non-consecutive double byte bias in a few seconds of time.
- Discovering of new non-consecutive (set of positive and negative) double byte biases in the key stream.

2. Related Work

Many researchers in information security were analyzed RC4 Cipher based on its weakness and proposed different solutions, but the bias calculations ways were slow, not efficient, and used a huge amount of data. This section illustrates the previous studies that are related to this work, Fluhrer S.R. & McGrew D.A. (2001) were the first persons that determined a new method to distinguish 8-bit of RC4 from random bits and detected a new type of bias was described as double byte bias in a consecutive byte pair. They were showing long-term biases for RC4, ten conditions are showing the positive biases that mean their probability are higher than the intended value and two conditions showed negative biases that mean their probability is lower than the meant value [9]. Al-Fardan N. J. et al. (2013) were measured the RC4 security in TLS and WPA and analyzed RC4 based on its single- and double-byte biases and attacking it based on its bias by using a plaintext retrieval attack. Their results were present that there are biases in the first 256 bytes of the RC4 keystream that can quarried by passive attacks to retrieve the plaintext by using 244 random keys [10]. Hammood M. M. and Yoshigoe K. (2016) were illustrated different biases in the RC4 keystream and analyzed the developed algorithms that determined in [6] by using a message passing interface environment and C programming, and the experiments are implemented by using a high execution system with 256 processor units [11]. Searan S. M. et al. (2016) were designed new algorithms for measuring single- and double-byte biases in RC4 key stream bytes, also they were attacked RC4 algorithm based on its single byte bias by using a newly designed algorithm and retrieved the first 32 bytes of RC4 plain text [12]. This work implemented the proposed algorithm on a personal computer and got the same biases that shown previously with new non-consecutive double byte biases and in less time (seconds only).

3. RC4 Concept

Many of stream ciphers are depended on the use of Linear Feedback Shift Registers (LFSRs) especially in the hardware, but the design of RC4 algorithm evades the use of LFSR [13]. This algorithm consists of two main parts to generate the key, the first is Key Scheduling Algorithm (KSA) and the second is Pseudo-Random Generation Algorithm (PRGA) that implemented sequentially [14]. KSA is more problematic, it was prepared to be easy. At the beginning, few bytes of the PRGA output are biased or attached to some bytes of the private key; therefore, analyzing these bytes makes them probable for attacking RC4 [7]. The length of the private key is typical between 5 to 32 bytes and is recurrent to form the final key stream. In KSA, can produce initial permutation of RC4 by scramble the corresponding permutation using the key. This permutation (State) in KSA is used as an input to the second phase (PRGA) that generates the final key stream [14]. RC4 begins with the permutation and use a secret key to get a random permutation with KSA. Based on a secret key, the next step is PRGA that generates keystream bytes, which are XOR-ed with the main bytes to get the ciphertext [15]. The concept of RC4 is to make a permutation of the elements by swapping them to accomplish the higher randomness. The rc4 algorithm has a variable length of the key, which between (0-255) bytes to initialize the 256 bytes in the initial state array (State [0] to State [255]) [16]. The algorithms 1 and 2 show KSA and PRGA steps of the RC4 algorithm.

4. RC4 Weaknesses

There are many weaknesses detected in the RC4 algorithm. Some of these are easy and can be resolved, but the others are dangers that can be quarried by the attackers. RC4 failed in providing a high-security level because of the biases in the bytes of the keystream [18]. Roos [19] determines RC4 weaknesses that a high connection between the first state table bytes and generated bytes of the keystream. The main cause is the state table that

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024

began in series (0, 1, 2… 255) and at least one out of every 256 potential keys, the first generated key byte is highly attached with a few bytes of the key. Therefore, the keys allow precursor of the first bytes from the PRGA output. To eliminate this problem, it was suggested to ignore the first bytes of the PRGA output [19]. Mantin and Shamir [8] were illustrated the main RC4 weakness in the second round. The likelihood of zero output bytes. Fluher [9] found a large weakness, if anyone knows the private key portion then potential to attack fully over the RC4 [9]. Paul and Maitra [17] were found a private key by using the elementary state table, got equations on the initial state bases, and selected some of the secret key bytes based on assumption and keep private key discovery by using them equation. So the safeness of RC4 is based on a private key security and the internal states. Many attacks are focus on getting the secret key of the internal states [20]. The attack aims to recover the main key, the internal state [21], or the final key stream to access the main messages [22].

Algorithm 1 is KSA, which initializes the internal state.

ALGORITHM 1: KSA OF RC4 ALGORITHM
INPUT: Key.
1. Initialization: 1.1. For (a = 0 to 255) 1.2. State[a] = a 1.3. T[a] = Key [a mod key-length] 2. Set b = 0 3. Loop: 3.1. for (a = 0 to 255) 3.2. b = (b + State[a] + T[a]) mod 256 3.3. Swap (State[a], State[b]) 4. Output: State[a].

Algorithm 2 is PRNG. It produces the output keystream.

ALGORITHM 2: PRGA OF RC4 ALGORITHM
INPUT: State[a], Plaintext $_n$. **OUTPUT:** Key sequence.
1. Initialization: 1.1. a = 0 1.2. b = 0 2. While (True) 2.1. a = (a + 1) mod 256 2.2. b = (b + State[a]) mod 256 2.3. Swap (State[a], State[b]) 2.4. Key sequence = State [(State[a] + State[b]) mod 256] 3. Output: Key sequence.
Cipher-text $_n$ = Key sequence $_n$ $\oplus$ Plaintext $_n$ [17].

5. Analysing RC4 Algorithm Based on its Non-Consecutive Double Byte Bias

The researchers have studied and discovered biases beyond initial bytes and different multi-byte biases have been detected in the key stream of RC4. Fluhrer and McGrew were the first researchers that selected the biases in a consecutive pair of bytes (Zr, Zr+1) and detected double byte biases of RC4. They detected ten positive biases that mean their probability are upper than the meant value and detected two negative biases that mean their probability are lower than the intended value. Hammood et al. estimated the probability of the cipher for generating each pair of byte values, though each 256-byte cycles and got a complete view of the distributions of every pair of byte values at the positions (r, r+1). They replicated biases of Fluhrer and McGrew and Al-Fardan et al endorsed their work. They found two new positive biases not mentioned by Fluhrer and McGrew. Searan et al. evaluated the likelihood of producing each pair of byte values and got a full view of the distributions of each pair of byte values at the positions (z, z+1). They endorsed the Fluhrer and McGrew bias and Hammood bias. This work shows the discovery of new non-consecutive (set of positive and negative) double byte biases in the key stream of RC4 algorithm by using the newly designed algorithm.

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024

Table 1: New Non-Consecutive Double Byte Biases in RC4 Key Stream (Positive)

No.	Byte pair (Zi, Zi+n)	Internal variable of PRGA (Condition i)	Value of n	Probability
1	$(1, 2^a-3)$	i = 0	2	$2^{-2a}(1+2^{-a})$
2	$(2^{a-1}+3, 2^a-1)$	$i = 2^{a-1}+5$	3	$2^{-2a}(1+2^{-a})$
3	$(2^{a-1}-5, 2^{a-1}-1)$	$i = 2^{a-1}-4$	8	$2^{-2a}(1+2^{-a})$
4	$(2^{a-1}+5, 2^a-4)$	i = 6	10	$2^{-2a}(1+2^{-a})$
5	$(2^{a-1}+4, 2^{a-1}+4)$	$i = 2^{a-1}-5$	11	$2^{-2a}(1+2^{-a})$
6	$(2^{a-1}+6, 1)$	i = 2	12	$2^{-2a}(1+2^{-a})$
7	$(2^a-5, 8)$	$i = 2^{a-1}-3$	13	$2^{-2a}(1+2^{-a})$
8	$(4, 2^a-2)$	i = 0	16	$2^{-2a}(1+2^{-a})$
9	$(2^{a-1}+2, 2^{a-1}+3)$	$i = 2^{a-1}-1$	19	$2^{-2a}(1+2^{-a})$

Table 2: New Non-Consecutive Double Byte Biases in RC4 Key Stream (Negative)

No.	Byte pair (Zi, Zi+n)	Internal variable of PRGA (Condition i)	Value of n	Probability
1	$(2^a-8, 1)$	i = 6	2	$2^{-2a}(1-2^{-a})$
2	$(2^{a-1}+4, 7)$	$i = 2^{a-1}+5$	3	$2^{-2a}(1-2^{-a})$
3	$(2, 2^a-3)$	$i = 2^a-6$	6	$2^{-2a}(1-2^{-a})$
4	$(2^{a-1}-2, 2^a-6)$	$i = 2^{a-1}-4$	8	$2^{-2a}(1-2^{-a})$
5	$(2^{a-1}-1, 2^{a-1}-3)$	i = 8	10	$2^{-2a}(1-2^{-a})$
6	$(2^a-3, 2^{a-1}+3)$	i = 9	11	$2^{-2a}(1-2^{-a})$
7	$(2^{a-1}, 2)$	i = 10	12	$2^{-2a}(1-2^{-a})$
8	$(2^{a-1}+5, 5)$	i = 5	13	$2^{-2a}(1-2^{-a})$
9	$(0, 2^a-9)$	$i = 2^{a-1}-3$	14	$2^{-2a}(1-2^{-a})$
10	$(2^a-9, 2^a-9)$	$i = 2^a-2$	16	$2^{-2a}(1-2^{-a})$
11	$(2^{a-1}+2, 2^{a-1}+5)$	i = 1	19	$2^{-2a}(1-2^{-a})$

The below algorithm determines the Measurement of Non-Consecutive Double Byte Bias (Zr, Zr+n)

ALGORITHM 3: MEASURING DISTRIBUTIONS OF KEYSTREAM BYTES (ZR, ZR+N) (NON-CONSECUTIVE DOUBLE BYTE BIAS)

INPUT: K [k_1, k_2 … k_{16}].
OUTPUT: 3-Dimentions array.

1. a = b = a1 = r = 0
2. For (x = 1 to 2^{10})
2.1. Call Algorithm 1: KSA
2.2. For (x = 1 to 2^{32})
2.2.1. a = (a + 1) mod 256
2.2.2. b = (b + State[a]) mod 256
2.2.3. Swap (State[a], State[b])
2.2.4. Gen-Key = State[(State[a] + State[b]) mod 256]
2.2.5. A[r][Gen-Key][a1] = A[r][Gen-Key][a1] +1
2.2.6. Deducting new key with 16 bytes from each generated key to be new secret key.
2.2.7. While (x % n ==0)
r = Gen-Key
2.2.8. a1 = (a1 + 1) mod 256

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024

3. Output: A[r][Gen-Key][a1].

Figures 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11 show the results for running above algorithm that measures the distributions of keystream bytes (Zr, Zr+n) to discover possible non-consecutive double-byte biases for RC4 in the first 32 bytes, Y-axis determines the frequents of each pair of values when the pointer is '0'.

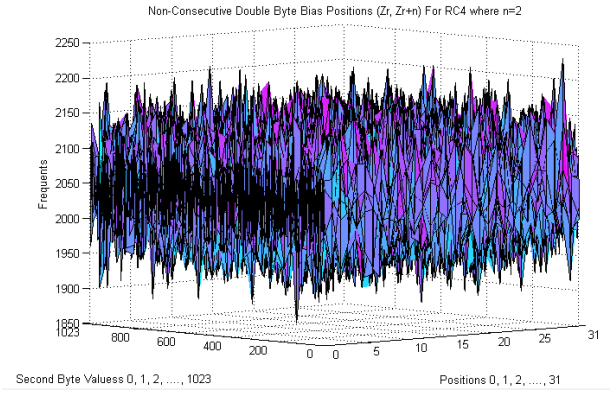

Figure 1: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=2.

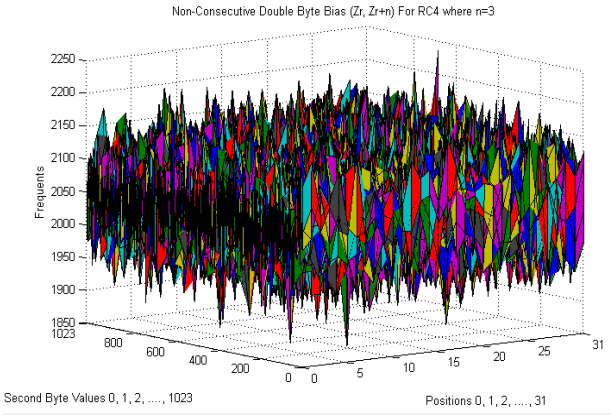

Figure 2: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=3.

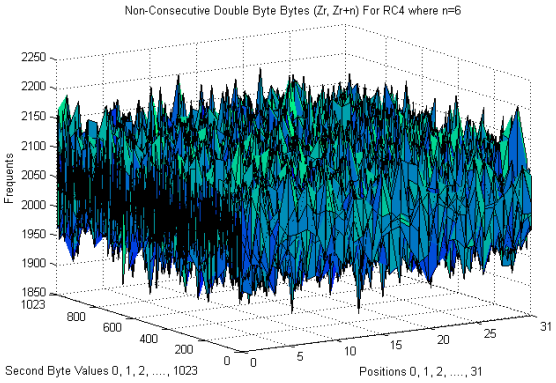

Figure 3: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=6.

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024

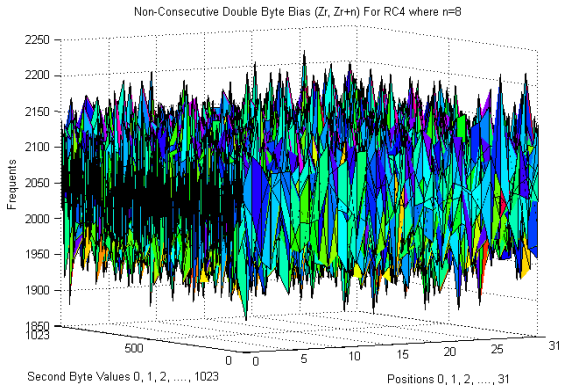

Figure 4: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=8.

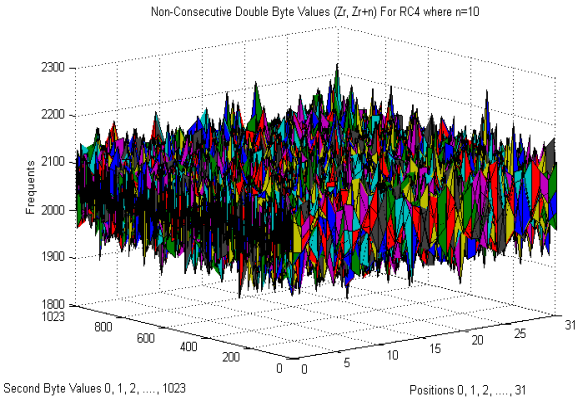

Figure 5: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=10.

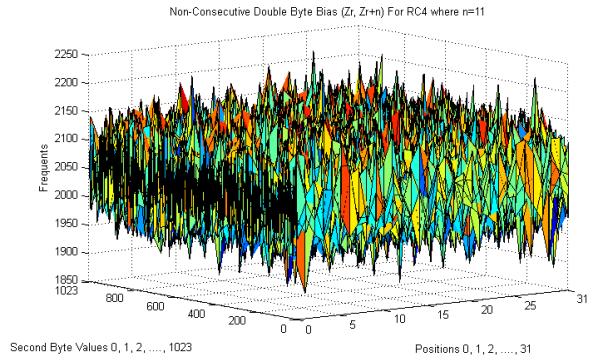

Figure 6: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=11.

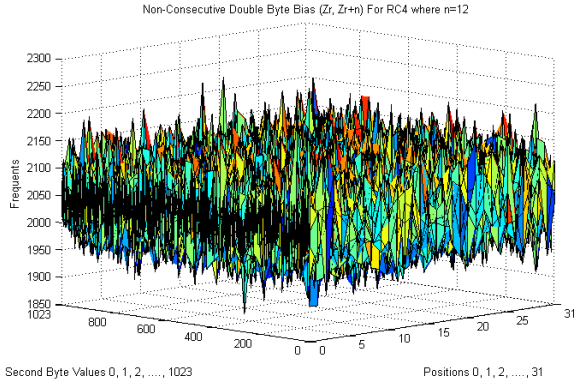

Figure 7: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=12.

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024

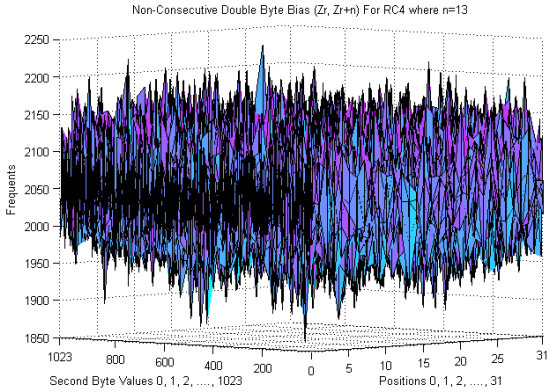

Figure 8: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=13.

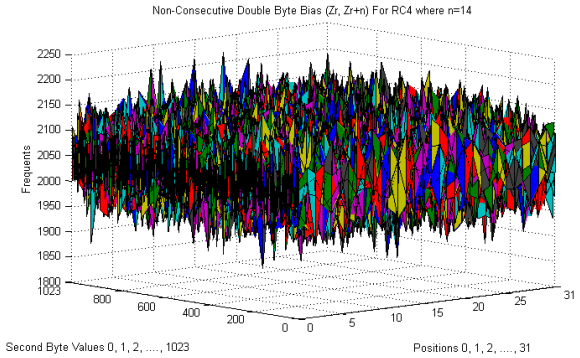

Figure 9: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=14.

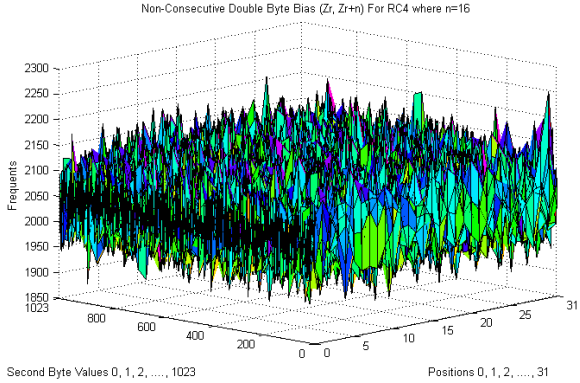

Figure 10: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=16.

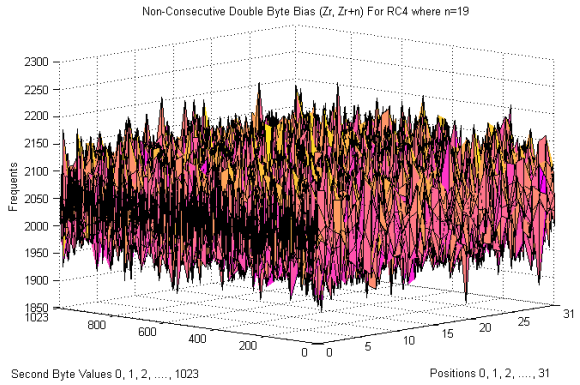

Figure 11: Non-Consecutive Double-Byte Biases (Zr, Zr+n) for RC4 where n=19.

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024

6. Conclusion

RC4 is an important encryption cipher that can be used for information secrecy on many communication networks as it fast and simple in implementation, but it has some weaknesses in its key stream bytes that these bytes are biased to some different values of the private key. RC4 biases are now quarried for making practical attacks on TLS. In this work, the analysis of RC4 algorithm is done for the first 32 positions by using new designed efficient algorithms. Moreover, shown a new set of non-consecutive double byte biases in the RC4 keystream bytes. As a future work, the proposed algorithms may be applied on 256 bytes by using parallel processors.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] Robshaw, Matthew, and Olivier Billet, eds. New stream cipher designs: the eSTREAM finalists. Vol. 4986. Springer, 2008..
[2] Darch Abed Dawar, A. (2024). Enhancing Wireless Security and Privacy: A 2-Way Identity Authentication Method for 5G Networks. International Journal of Mathematics, Statistics, and Computer Science, 2, 183–198. https://doi.org/10.59543/ijmscs.v2i.9073
[3] Prasithsangaree, Phongsak, and Prashant Krishnamurthy. "Analysis of energy consumption of RC4 and AES algorithms in wireless LANs." GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489). Vol. 3. IEEE, 2003..
[4] Karahan, Mehmet. New attacks RC4A and VMPC. MS thesis. Bilkent Universitesi (Turkey), 2015.
[5] Paul, Goutam. "Structural weakness of the key scheduling of RC4." Jadavpur university: IFW 2000 (2007): 4000..
[6] Hammood, Maytham M., Kenji Yoshigoe, and Ali M. Sagheer. "RC4-2S: RC4 stream cipher with two state tables." Information Technology Convergence: Security, Robotics, Automations and Communication. Dordrecht: Springer Netherlands, 2013. 13-20..
[7] Khine, Lae Lae. "A new variant of RC4 stream cipher." International Journal of Physical and Mathematical Sciences 3.2 (2009): 152-155..
[8] Mantin, Itsik, and Adi Shamir. "A practical attack on broadcast RC4." International workshop on fast software encryption. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.
[9] Fluhrer, Scott R., and David A. McGrew. "Statistical analysis of the alleged RC4 keystream generator." Fast Software Encryption: 7th International Workshop, FSE 2000 New York, NY, USA, April 10–12, 2000 Proceedings 7. Springer Berlin Heidelberg, 2001.
[10] AlFardan, Nadhem J., et al. "On the security of RC4 in TLS and WPA." USENIX Security Symposium. Vol. 173. 2013.
[11] Hammood, Maytham M., and Kenji Yoshigoe. "Previously overlooked bias signatures for RC4." 2016 4th International Symposium on Digital Forensic and Security (ISDFS). IEEE, 2016.
[12] Searan, S. M., A. M. Sagheer, and M. M. Hammood. "Analyzing of RC4 Algorithm Based on Its Single and Double Byte Bias by Using New Algorithms." International Conference on Change, Innovation, Informatics and Disruptive Technology, London–UK. 2016.
[13] Hammood, Maytham M., Kenji Yoshigoe, and Ali M. Sagheer. "RC4 stream cipher with a random initial state." Information Technology Convergence: Security, Robotics, Automations and Communication. Springer Netherlands, 2013.
[14] Garman, Christina, Kenneth G. Paterson, and Thyla Van der Merwe. "Attacks Only Get Better: Password Recovery Attacks Against {RC4} in {TLS}." 24th USENIX Security Symposium (USENIX Security 15). 2015.
[15] Maitra, Subhamoy, and Goutam Paul. "Analysis of RC4 and proposal of additional layers for better security margin." Progress in Cryptology-INDOCRYPT 2008: 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Proceedings 9. Springer Berlin Heidelberg, 2008.
[16] Orumiehchiha, Mohammad Ali, et al. "Cryptanalysis of RC4 (n, m) Stream Cipher." Proceedings of the 6th International Conference on Security of Information and Networks. 2013.
[17] Maitra, Subhamoy, and Goutam Paul. "New form of permutation bias and secret key leakage in keystream bytes of RC4." Fast Software Encryption: 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers 15. Springer Berlin Heidelberg, 2008.

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024

[18] Hammood, Maytham M., Kenji Yoshigoe, and Ali M. Sagheer. "Enhancing security and speed of RC4." International Journal of Computing and Network Technology 3.02 (2015).
[19] Roos, Andrew. "A class of weak keys in the RC4 stream cipher." (1995).
[20] Pardeep, P., and P. K. Pateriya. "PC 1-RC4 and PC 2-RC4 algorithms: Pragmatic enrichment algorithms to enhance RC4 stream cipher algorithm." International Journal of Computer Science and Network 1.3 (2012): 2277-5420.
[21] Ohigashi, Toshihiro, et al. "How to recover any byte of plaintext on RC4." International Conference on Selected Areas in Cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
[22] Searan, Sura M., and Ali M. Sagheer. "Modification of RC4 algorithm by using two state tables and initial state factorial." International Journal of Computer Network and Information Security 8.12 (2016): 1-8.

Doi : https://doi.org/10.54216/JCIM.130206
Received: July 04, 2023 Revised: November 17, 2023 Accepted: February 18, 2024