(Non-)Random Sequences from (Non-)Random Permutations - Analysis of RC4 stream cipher*

Sourav Sen Gupta[1], Subhamoy Maitra[1], Goutam Paul[2], and Santanu Sarkar[1]

[1] Applied Statistics Unit, Indian Statistical Institute, Kolkata 700 108, India
[2] Dept. of Computer Science and Engg., Jadavpur University, Kolkata 700 032, India

Abstract. RC4 has been the most popular stream cipher in the history of symmetric key cryptography till date. Its internal state contains a pseudo-random permutation over all n-bit words (typically $n = 8$) and it attempts to generate a pseudo-random sequence of words by extracting elements of this permutation. Since more than last twenty years, numerous cryptanalytic results on RC4 stream cipher have been published. Many of these results are based on some non-random (biased) events involving the secret key or the state variables or the output sequence, or a combination of them.

Though biases based on the secret key is common in RC4 literature, none of the existing ones depends on the length of the secret key. In the first part of this paper, we report significant biases involving the length of the secret key, for the first time in the literature.

In the second part of the paper, theoretical proofs of some significant initial-round empirical biases observed by Sepehrdad, Vaudenay and Vuagnoux [SAC 2010] are presented. Another important result presented here is the derivation of the complete probability distribution of the first byte of RC4 output sequence, a problem left open for a decade since the observation by Mironov [CRYPTO 2002]. Further, the existence of positive biases towards zero for all the initial bytes 3 to 255 is proved and exploited towards a generalized broadcast attack on RC4 stream cipher.

The above biases discussed in this paper, like most of the existing biases in RC4 literature, are short-term and do not last after a few initial rounds. The last part of this paper investigates the long-term manifestation of short-term biases in RC4 output sequence. A careful analysis of the periodic structure of RC4 evolution proves the first long-term generalization of Mantin and Shamir's [FSE 2001] famous second-byte bias.

Keywords: Bias, Cryptography, Distinguisher, Probability Distribution, Pseudo-Random Permutation, Pseudo-Random Word, Random Sequences, RC4, Sequences, Stream Ciphers.

* This is a substantially revised and extended version of the papers [14] of FSE 2011 and [28] of SAC 2011. Sections 2 and 3.1 are based on [28] and Section 3.3 is based on [14] with major revision in the proof of Theorem 12. Sections 3.2 and 4 are completely new technical contributions in this paper.

1 Introduction

Stream ciphers constitute a major branch of Cryptology, especially in the domain of symmetric key ciphers. Stream ciphers claim to output a *pseudo-random sequence* of bits, called the *keystream*, and encryption is done by masking the plaintext (considered as a sequence of bits) by the keystream. The masking operation is just a simple XOR in general, and the ciphertext is also a sequence of bits of the same length as that of the plaintext. For ideal information theoretic 'perfect secrecy' of the scheme, it is desired that the masking is done using a one-time pad where a unique sequence of bits is used as a mask for each plaintext-ciphertext pair. In reality however, a one-time pad is not practical, as it requires a source of truly random sequence of bits to mask the plaintext in the encryption operation. Instead, a computational notion of secrecy is ensured by the pseudo-random output sequence generated by a stream cipher. Any non-random event in the internal state or the output sequence of a stream cipher is not desired from a cryptographic point of view, and a rigorous analysis of a stream cipher is performed to identify the presence of any such non-randomness in its design.

The most important and cryptographically significant goal of a stream cipher is to produce a pseudo-random sequence of bits or words using a fixed length secret key (or a secret key paired with an initialization vector). Over the last three decades of research and development in stream ciphers, a number of designs have been proposed and analyzed by the cryptology community. One of the main ideas for building a stream cipher relies on constructing a *pseudo-random permutation* and thereafter extracting a pseudo-random sequence from this permutation. Interestingly, even if the underlying permutation is pseudo-random, if the method of extracting the words from the permutation is not carefully designed, then it may be possible to identify certain biased events in the final output sequence of the cipher.

Till date, the most popular stream cipher among the cryptologists has been RC4, which is designed using the same principle of extracting pseudo-random words from pseudo-random permutations. This cipher gains its popularity for its intriguing simplicity that has made it widely accepted for numerous software and web applications. Our paper takes a critical look at some important non-random events of the RC4 stream cipher, thereby illustrating some key design vulnerabilities in the shuffle-exchange paradigm of stream ciphers.

1.1 RC4 stream cipher

RC4 is the most widely deployed commercial stream cipher, having applications in network protocols such as SSL, WEP, WPA and in Microsoft Windows, Apple OCE, Secure SQL etc. It was (allegedly) designed in 1987 by Ron Rivest. The design was a trade secret since then, and was anonymously posted on the web in 1994. Later, the description was verified by comparing the outputs of the posted design with those of the licensed systems using proprietary versions of the original cipher. The public design was never officially claimed to be the original cipher, and hence it is also called the 'Alleged RC4' or 'ARC4'.

The cipher consists of two major components, the Key Scheduling Algorithm (KSA) and the Pseudo-Random Generation Algorithm (PRGA). The internal state of RC4 contains a permutation of all n-bit words (typically, $n = 8$), i.e., a permutation of size $N = 2^n$ words (typically, $N = 256$). The key K is of the same size (N words) as well. However, the original secret key is of length typically between 5 to 32 words, and is repeated to form the expanded key K. The KSA produces the initial pseudo-random permutation of RC4 by scrambling an identity permutation using key K. The initial permutation S produced by the KSA acts as an input to the next procedure PRGA that generates the output sequence.

The practical form of the cipher is byte-oriented, i.e., it operates with $n = 8$ and hence $N = 256$. The RC4 algorithms KSA and PRGA are as shown in Fig. 1.

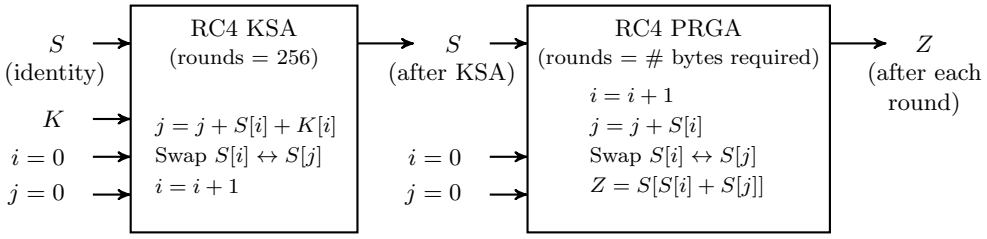

Fig. 1. Key-Scheduling Algorithm (KSA) and Pseudo-Random Generation Algorithm (PRGA) of RC4.

Notation: For round $t \geq 1$ of RC4 PRGA, we denote the indices by i_t, j_t, the keystream output byte by Z_t, and the permutations before and after the swap by S_{t-1} and S_t respectively. After t rounds of KSA, we denote the state variables by adding a superscript K to each variable. By S_0^K and S_0, we denote the initial permutations before KSA and PRGA respectively. Note that S_0^K is the identity permutation and $S_0 = S_N^K$ is the permutation obtained right after the completion of KSA. In this paper, all additions in the context of RC4 are to be considered modulo N.

1.2 An overview of RC4 cryptanalysis

The goal of RC4, like all stream ciphers, is to produce a pseudo-random sequence of bits from the internal permutation, which in turn should be pseudo-random. Hence, one of the main ideas for RC4 cryptanalysis is to investigate for *biases*, that is, statistical weaknesses that can be exploited to computationally distinguish the output sequence of RC4 from a truly random sequence of bytes with a considerable probability of success. The target of attack may be to exploit the non-randomness in the internal state of RC4, or the non-randomness of keyword-extraction from the internal permutation. Both ideas have been put to practice in various ways in the literature, and the main theme of attacks on RC4 has been in three directions, as follows.

1. **Key recovery attack:** Key recovery from permutation was first proposed in [24] and later studied in [3, 4, 10]. Key recovery from keystream output primarily exploits the use of RC4 in WEP and WPA. The analysis in [7,18] are applicable towards RC4 in WEP mode, and there are quite a few practical attacks [11,34,35] on the WEP protocol as well. After a practical breach [33] of WEP, the new variant WPA came into the picture. This too used RC4 as a backbone, and the most recent result published in [30] mounts a distinguishing attack on RC4 in WPA.

2. **State recovery attack:** The state-space of RC4 is around 2^{1700}. The first important state recovery attack was due to [12] that required a complexity of 2^{779}. After a series of small improvements [21, 31, 32], the best attack with complexity 2^{241} appeared in [19], after which the use of secret keys of length more than 30 bytes is not recommended any more.

3. **Biases and Distinguishers:** Most of the results are targeted towards specific short-term (involving only the initial few bytes of the output) biases and correlations [8,9,14,16,20,22,23,27,29], while there exist only a few results for long-term (prominent even after discarding an arbitrary number of initial bytes of the output) biases [2, 6, 17].

Fig. 2 gives a chronological summary of the important cryptanalytic results on RC4 till date.

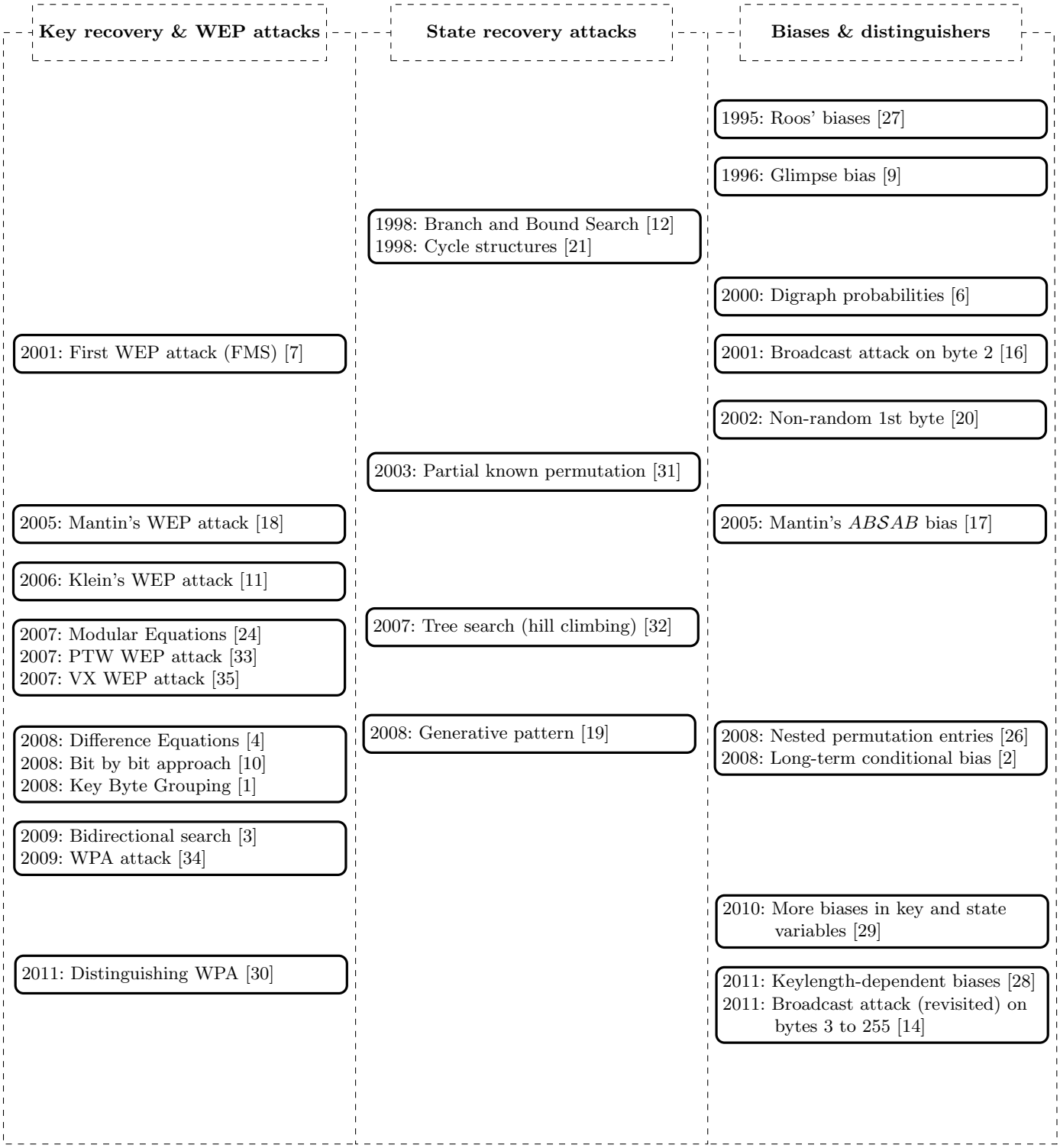

Fig. 2. A chronological summary of RC4 cryptanalysis.

1.3 Our contributions

In this paper, we extend and supplement the literature of RC4 cryptanalysis in three directions; we introduce the concept of keylength dependent biases, identify new short-term biases and investigate significant long-term manifestation of short-term biases in RC4. Sections 2, 3 and 4 contain the technical results of this paper, summarized as follows.

Section 2: In SAC 2010, Sepehrdad, Vaudenay and Vuagnoux [29] reported the empirical bias $\Pr(S_{16}[j_{16}] = 0 \mid Z_{16} = -16) = 0.038488$ without any proof. Our detailed investigation of this bias reveals that the number 16 comes from the secret key length of 16 bytes in which the experiments were performed in [29] and the same bias holds for any secret key length. Along the same line of investigation, in Section 2 of our paper, we observe and prove a family of related conditional biases involving secret key and its length. Moreover, we establish a strong correlation between the length l of the secret key and the l-th byte in the output sequence (typically for $5 \leq l \leq 30$), and thus propose a method to predict the keylength of the cipher by observing the output sequence. To the best of our knowledge, this is the *first set of keylength dependent biases* discovered in RC4.

Section 3: In this section, we investigate RC4 PRGA and discuss biases in the initial rounds.

1. In Section 3.1, we provide *theoretical proofs for some significant empirical biases* of RC4 involving the state variables in the initial rounds, that were reported by Sepehrdad et al. [29].
2. In CRYPTO 2002, Mironov [20] observed that the 1st byte Z_1 of RC4 output sequence has a negative bias towards zero, and also found an interesting non-uniform probability distribution (similar to a sine curve) for all other values of this byte. However, the theoretical proof remained open. In Section 3.2, we derive the *complete theoretical distribution of* Z_1.
3. In FSE 2001, Mantin and Shamir [16] showed that $\Pr(Z_2 = 0) \approx \frac{2}{N}$ in RC4 output sequence, whereas this should be $1/N$ in case of a random sequence of bytes. They also claimed that such bias does not exist in any other subsequent byte in the output sequence. Contrary to this claim, in Section 3.3, we establish that *all the bytes 3 to 255 of RC4 initial keystream are biased to zero*. In addition, we rigorously study the non-randomness of index j to find a strong bias of j_2 towards 4. We further use this bias to guess the internal state variable $S_2[2]$ from Z_2.

Section 4: Biases in initial rounds of RC4 has no effect if one throws away some initial bytes from the output sequence of RC4. This naturally motivates a quest for *long-term* biases in the RC4 output, if any exists. In Section 4, we seek for long-term manifestation of the short-term biases and find a *new long-term conditional bias* in RC4 output sequence, hitherto undiscovered in the literature.

Before presenting the technical contribution of this paper, we shall first study the background and some key technical concepts that are required for our results.

1.4 Preliminaries and technical background

In this paper, we focus on non-random (biased) events in RC4 and their cryptographic significance. To utilize a biased event towards an attack on the stream cipher, one needs to identify the bias in the very first place. In this section, we build the information theoretic tools towards this purpose.

Bias and distinguisher: For a stream cipher, if there is an event such that the probability of occurrence of the event is different from that in case of a uniformly random sequence of bits, the

event is said to be *biased*. If there exists a biased event based only on the bits of the output sequence, then such an event gives rise to a *distinguisher* for the cipher that can computationally differentiate between the output sequence of the stream cipher and a truly random sequence of bits. The efficiency of a distinguisher is judged by the number of samples required to identify the bias. For a distinguisher to successfully identify and exploit a bias, one requires to inspect a certain length of the output sequence so that one can collect sufficient number of samples for the event under consideration. The less is the number of samples required, the more is the efficiency of the distinguisher. The technical details are as follows.

Number of samples required to identify a bias: Let E be an event based on some key bits or internal state bits or keystream bits or a combination of them in a stream cipher. Suppose, $\Pr(E) = p$ for a uniformly random sequence of bits (i.e., for the output sequence of an ideal stream cipher) and $\Pr(E) = p(1+q)$ for the output sequence of the stream cipher under consideration. The cryptanalytic motivation of studying a stream cipher is to distinguish these two sequences (uniform random sequence and output sequence of the stream cipher) in terms of the difference in the above probabilities when $q \neq 0$. It requires the formal information theoretic notion of 'relative entropy' between two sequences.

The relative entropy between two discrete probability distributions $P(.)$ and $Q(.)$ is given by the Kullback-Leibler divergence [13]

$$D_{KL}(P||Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)},$$

where x runs over all the sample points. For the above-mentioned single event E with probabilities p and $p(1+q)$ in two different distributions $P(.)$ and $Q(.)$, the relative entropy is given by

$$\begin{aligned} & p \log_2 \left(\frac{p}{p(1+q)} \right) + (1-p) \log_2 \left(\frac{1-p}{1-p(1+q)} \right) \\ & \qquad = p \log_2 \left(1 - \frac{q}{1+q} \right) + (1-p) \log_2 \left(1 + \frac{pq}{1-p(1+q)} \right) \\ & \qquad \approx -p \left(\frac{q}{1+q} \right) + (1-p) \left(\frac{pq}{1-p(1+q)} \right) \approx pq^2. \end{aligned}$$

If P, Q are two distributions defined over the domain A and P', Q' are two other distributions defined over the domain B, then it can be shown that the overall relative entropy of the joint distributions PP' and QQ' is given by $D(PP'||QQ') = D(P||Q) + D(P'||Q')$. Applying this to n samples from the same distribution, the relative entropy is obtained as npq^2.

According to [5], the bound for false positive rate (α) and false negative rate (β) satisfy the following inequality.

$$npq^2 \geq \beta \log_2 \frac{\beta}{1-\alpha} + (1-\beta) \log_2 \frac{1-\beta}{\alpha}$$

For $\alpha = \beta$, this relation reduces to

$$n \geq \left(\frac{1}{pq^2} \right) \cdot (1-2\alpha) \log_2 \frac{1-\alpha}{\alpha}.$$

In our context, *false positive* means that the test sequence is actually from the stream cipher, but we decide it to be random and *false negative* means that the test sequence is actually random, but we decide it to be from the stream cipher.

Thus for a given false positive or negative rate α ($= \beta$), one needs roughly $O(1/pq^2)$ many samples to perform the distinguishing test. In particular, $n \geq 1/pq^2$ signifies $\alpha \approx 0.2227$, i.e., a success probability of approximately 0.7773. Since $0.7773 > 0.5$ is a reasonably good success probability, $O(1/pq^2)$ many samples are considered enough to reliably apply the distinguisher.

This gives an estimate of the number of samples needed to confirm a bias (either through simulation or from practical data). If the biased event is a function of the bits in the output sequence only, then the number of samples needed gives an estimate of the data complexity to mount a distinguishing attack. We shall use this notion of *sample complexity* while judging the effectiveness of any bias discussed throughout this paper.

Now we present the technical contribution of this paper in the following sections.

2 Biases based on secret key and its length

In this section, we present a family of biases in RC4 that are dependent on the length of the secret key. In SAC 2010, Sepehrdad et al. [29] discovered several correlations in RC4 PRGA using DFT based approach. A list of such biases was presented in [29, Fig. 10], and the authors commented:

> "After investigation, it seems that all the listed biases are artifact of a new conditional bias which is $\Pr[S'_{16}[j'_{16}] = 0 \mid Z_{16} = -16] = 0.038488$."

However, the authors also admitted that

> *"So far, we have no explanation about this new bias."*

In our notation of Section 1.1, the above event is denoted as $(S_{16}[j_{16}] = 0 \mid Z_{16} = -16)$. While exploring this conditional bias in RC4 PRGA, we could immediately observe two things:

1. The number 16 in the result comes from the keylength that is consistently chosen to be 16 in [29] for most of the experimentation. In its general form, the conditional bias should be stated as:

$$\Pr\left(S_l[j_l] = 0 \mid Z_l = -l\right) \approx \frac{10}{N}. \tag{1}$$

   It is surprising why this natural observation could not be identified earlier.
2. Along the same line of investigation, we could find a family of related conditional biases, stated in their general form as follows:

$$\Pr(Z_l = -l \mid S_l[j_l] = 0) \approx 10/N \tag{2}$$
$$\Pr(S_l[l] = -l \mid S_l[j_l] = 0) \approx 30/N \tag{3}$$
$$\Pr(t_l = -l \mid S_l[j_l] = 0) \approx 30/N \tag{4}$$
$$\Pr(S_l[j_l] = 0 \mid t_l = -l) \approx 30/N \tag{5}$$

Note that bias (2) follows almost immediately from bias (1), and biases (5) and (4) are related in a similar way. Moreover, bias (3) implies bias (4) as $t_l = S_l[l] + S_l[j_l] = -l$ under the given condition. However, we investigate even further to study the bias caused in Z_l due to the state variables.

2.1 Dependence of conditional biases on RC4 secret key

We found that all of the aforementioned conditional biases between the two events under consideration are related to the following third event that is dependent on the values and the length of the RC4 secret key.

$$\sum_{i=0}^{l-1} K[i] + \frac{l(l-1)}{2} \equiv -l \pmod{N}$$

We shall henceforth denote the above event by $(f_{l-1} = -l)$, following the notation of Paul and Maitra [26], and this event is going to constitute the base for most of the conditional probabilities we consider hereafter. We consider $\Pr(f_{l-1} = -l) \approx \frac{1}{N}$, assuming that f_{l-1} can take any value modulo N uniformly at random.

Extensive experimentation with different keylengths (100 million runs for each keylength $1 \leq l \leq 256$) revealed strong bias in all of the following events:

$$\Pr(S_l[j_l] = 0 \mid f_{l-1} = -l), \qquad \Pr(S_l[l] = -l \mid f_{l-1} = -l),$$
$$\Pr(t_l = -l \mid f_{l-1} = -l), \qquad \Pr(Z_l = -l \mid f_{l-1} = -l).$$

Each of the correlations (1), (2), (3), (4), and (5) is an artifact of these common keylength-based correlations in RC4 PRGA. In this section, we discuss and justify all these conditional biases.

To prove the observations in this section, we shall require the following existing results from the literature of key-correlation in RC4. These are the correlations observed by Roos [27] in 1995, which were later proved by Paul and Maitra [26].

Proposition 1. [26, Lemma 1] *If index j is pseudo-random at each KSA round, we have*

$$\Pr\left(j_{y+1}^K = f_y\right) \approx \left(1 - \frac{1}{N}\right)^{1+\frac{y(y+1)}{2}} + \frac{1}{N}.$$

Proposition 2. [26, Corollary 1] *On completion of KSA in the RC4 algorithm,*

$$\Pr(S_0[y] = f_y) = \Pr(S_N^K[y] = f_y) \approx \left(1 - \frac{y}{N}\right) \cdot \left(1 - \frac{1}{N}\right)^{\frac{y(y+1)}{2}+N} + \frac{1}{N}.$$

Proposition 3. [26, Corollary 1] *On completion of KSA, we have for $0 \leq y \leq 31$,*

$$\Pr(S_0[S_0[y]] = f_y) \approx \left[\frac{y}{N} + \frac{1}{N}\left[1 - \frac{1}{N}\right]^{2-y} + \left[1 - \frac{y}{N}\right]^2 \left[1 - \frac{1}{N}\right]\right] \left[1 - \frac{1}{N}\right]^{\frac{y(y+1)}{2}+2N-4}.$$

Note that in each of the above statements,

$$f_y = S_0^K\left[\sum_{x=0}^{y} S_0^K[x] + \sum_{x=0}^{y} K[x]\right] = \sum_{x=0}^{y} x + \sum_{x=0}^{y} K[x] = \sum_{x=0}^{y} K[x] + \frac{y(y+1)}{2},$$

and we shall henceforth use this shorthand notation f_y throughout the paper.

2.2 Proof of keylength dependent conditional biases

In this section, we will prove the four main conditional biases that we have observed. Each depends on the event $(f_{l-1} = -l)$, and can be justified as follows. In each of the following theorems, the notation '$x : A \xrightarrow{\alpha} B$' denotes that the value x transits from position A to position B with probability α.

Theorem 1. *Suppose that l is the length of the secret key used in the RC4 algorithm. Given $f_{l-1} = \sum_{i=0}^{l-1} K[i] + l(l-1)/2 = -l$, we have*

$$\Pr(S_l[j_l] = 0) \approx \frac{1}{N} + \left[1 - \frac{l}{N}\right]\left[1 - \frac{1}{N}\right]^{N+l-2}\left[\left[1 - \frac{1}{N}\right]^{1+\frac{l(l+1)}{2}} + \frac{1}{N}\right]$$

$$\Pr(S_{l-2}[l-1] = -l) \approx \frac{1}{N} + \left[1 - \frac{1}{N}\right]^{l-1}\left[\left[1 - \frac{l-1}{N}\right]\left[1 - \frac{1}{N}\right]^{N+\frac{l(l-1)}{2}} + \frac{1}{N}\right]$$

Proof. For proving the first conditional bias, we need to trace the value 0 over KSA and the first l rounds of PRGA. We start from $S_0^K[0] = 0$, as the initial state S_0^K of KSA is the identity permutation in RC4. The following gives the trace pattern for 0 through the complete KSA and l initial rounds of PRGA. We shall discuss some of the transitions in details.

$$0 : S_0^K[0] \xrightarrow{1} S_1^K[K[0]] \xrightarrow{p_1} S_l^K[K[0]] \xrightarrow{p_2} S_{l+1}^K[l] \xrightarrow{p_3} S_{l-1}[l] \xrightarrow{1} S_l[j_l]$$

Here $p_1 = \left(1 - \frac{l}{N}\right)\left(1 - \frac{1}{N}\right)^{l-1}$ denotes the probability that index $K[0]$ is not touched by i^K and j^K in the first l rounds of KSA, $p_2 = \left(1 - \frac{1}{N}\right)^{1+\frac{l(l+1)}{2}} + \frac{1}{N}$ denotes the probability $\Pr(j_{l+1}^K = f_l = K[0])$ (using Proposition 1) such that 0 is swapped from $S_l^K[K[0]]$ to $S_{l+1}^K[l]$, and $p_3 = \left(1 - \frac{1}{N}\right)^{N-2}$ denotes the probability that the location $S_{l+1}^K[l]$ containing 0 is not touched by i^K, j^K in the remaining $N - l - 1$ rounds of KSA or by i, j in the first $l - 1$ rounds of PRGA. So, this path gives a total probability of $p_1p_2p_3$. If this path does not hold, we assume that the event $(S_l[j_l] = 0)$ still holds at random, with probability $1/N$. Thus, the total probability is obtained as

$$\Pr(S_l[j_l] = 0) = p_1p_2p_3 + (1 - p_1p_2p_3) \cdot \frac{1}{N} = \frac{1}{N} + \left(1 - \frac{1}{N}\right) p_1p_2p_3.$$

We do a similar propagation tracking for the value $f_{l-1} = -l$ to prove the second result, and the main path for this tracking looks as follows.

$$-l : S_0^K[-l] \xrightarrow{p_4} S_0[l-1] \xrightarrow{p_5} S_{l-2}[l-1]$$

Here we get $p_4 = \Pr(S_0[l-1] = f_{l-1}) = \left(1 - \frac{l-1}{N}\right)\left(1 - \frac{1}{N}\right)^{N+\frac{l(l-1)}{2}} + \frac{1}{N}$ using Proposition 2 directly, and $p_5 = \left(1 - \frac{1}{N}\right)^{l-2}$ denotes the probability that the index $(l-1)$, containing $-l$, is not touched by i, j in the first $l-2$ rounds of PRGA. Similar to the previous proof, the total probability can be calculated as

$$\Pr(S_{l-2}[l-1] = -l) = p_4p_5 + (1 - p_4p_5) \cdot \frac{1}{N} = \frac{1}{N} + \left(1 - \frac{1}{N}\right) p_4p_5.$$

We get the claimed results by substituting p_1, p_2, p_3 and p_4, p_5 appropriately. □

If we substitute $l = 16$, the most common keylength for RC4, and $N = 256$, we get the probabilities of Theorem 1 of magnitude

$$\Pr(S_l[j_l] = 0 \mid f_{l-1} = -l) \approx \Pr(S_{l-2}[l-1] = -l \mid f_{l-1} = -l) \approx 50/N.$$

These are, to the best of our knowledge, *the best known key-dependent conditional biases in RC4 PRGA till date*. The estimates closely match the experiments we performed over 100 million runs with 16-byte keys. In the next theorem, we look at a few natural consequences of these biases.

Theorem 2. *Suppose that l is the length of the RC4 secret key. Given that $f_{l-1} = \sum_{i=0}^{l-1} K[i] + l(l-1)/2 = -l$, the probabilities $\Pr(S_l[l] = -l \mid f_{l-1} = -l)$ and $\Pr(t_l = -l \mid f_{l-1} = -l)$ are approximately*

$$\frac{1}{N} + \left(1 - \frac{1}{N}\right) \cdot \left[\frac{1}{N} + \left[1 - \frac{l}{N}\right]\left[1 - \frac{1}{N}\right]^{N+l-2}\left[\left[1 - \frac{1}{N}\right]^{1+\frac{l(l+1)}{2}} + \frac{1}{N}\right]\right] \cdot \left[\frac{1}{N} + \left[1 - \frac{1}{N}\right]^{l-1}\left[\left[1 - \frac{1}{N}\right]^{N-l} + \frac{1}{N}\right]\right]$$

Proof. Before proving the path for the target events, let us take a look at rounds $l-1$ and l of RC4 PRGA when $S_{l-2}[l-1] = -l$ and $S_{l-1}[l] = 0$. In this situation, we have the following propagation for the value $-l$.

$$-l : S_{l-2}[l-1] \xrightarrow{1} S_{l-1}[j_{l-1}] = S_{l-1}[j_l] \xrightarrow{1} S_l[l]$$

In the above path, the equality holds because $j_l = j_{l-1} + S_{l-1}[l] = j_{l-1} + 0$ as per the conditions. Again, we have $S_l[j_l] = S_{l-1}[l] = 0$, implying $t_l = S_l[l] + S_l[j_l] = -l + 0 = -l$ as well. This explains the same expression for the probabilities of the two events in the statement.

Note that we require both the events $(S_l[j_l] = 0 \mid f_{l-1} = -l)$ and $(S_{l-2}[l-1] = -l \mid f_{l-1} = -l)$ to occur simultaneously, and need to calculate the joint probability. Also note that there is a significant overlap between the tracking paths of these two events, as they both assume that the first l positions of the state S_0^K are not touched by j^K in the first l rounds of KSA (refer to the proof of Theorem 1 of this paper and proofs of [26, Theorem 1, Corollary 1] for details). In other words, if we assume the occurrence of event $(S_l[j_l] = 0 \mid f_{l-1} = -l)$ (with probability p_6, as derived in Theorem 1, say), then the precondition for $(S_{l-2}[l-1] = -l \mid f_{l-1} = -l)$ will be satisfied, and thus the modified conditional probability is $\Pr(S_{l-2}[l-1] = -l \mid S_l[j_l] = 0 \wedge f_{l-1} = -l) = \frac{1}{N} + \left[1 - \frac{1}{N}\right]^{l-1}\left[\left[1 - \frac{1}{N}\right]^{N-l} + \frac{1}{N}\right] = p_7$, say. Now, we can compute the joint probability of the two events as

$$\Pr(S_l[l] = -l \mid f_{l-1} = -l) = p_6 p_7 + (1 - p_6 p_7) \cdot \frac{1}{N} = \frac{1}{N} + \left(1 - \frac{1}{N}\right) \cdot p_6 p_7.$$

Substituting the values of p_6 and p_7, we obtain the desired result. Event $(t_l = -l)$ follows immediately from $(S_l[l] = -l)$, with the same conditional probability. □

Substituting $l = 16$ and $N = 256$, we get the probabilities of Theorem 2 of the magnitude $\Pr(S_l[l] = -l \mid f_{l-1} = -l) = \Pr(t_l = -l \mid f_{l-1} = -l) \approx 20/N$. These estimates closely match our experimental results taken over 100 million runs of RC4 with 16-byte keys.

The bias in $(Z_l = -l)$ is caused due to the event $f_{l-1}[l]$, but in a different path than the one we have discussed so far. We prove the formal statement next as Theorem 3.

Theorem 3. *Suppose that l is the length of the secret key of RC4. Given that $f_{l-1} = \sum_{i=0}^{l-1} K[i] + l(l-1)/2 = -l$, the probability $\Pr(Z_l = -l)$ is approximately*

$$\frac{1}{N} + \left[1 - \frac{1}{N}\right] \cdot \left[\frac{1}{N} + \left[1 - \frac{l}{N}\right]\left[1 - \frac{1}{N}\right]^{N+l-2}\left[\left[1 - \frac{1}{N}\right]^{1+l} + \frac{1}{N}\right]\right] \cdot \left[\frac{1}{N} + \left[1 - \frac{1}{N}\right]^{l+1} \Pr(S_0[S_0[l-1]] = f_{l-1})\right]$$

Proof. The proof is similar to that of Theorem 2 as both require $S_l[j_l] = S_{l-1}[l] = 0$ to occur first. Note that if $S_l[j_l] = S_{l-1}[l] = 0$, we will always have

$$Z_l = S_l[S_l[l] + S_l[j_l]] = S_l[S_{l-2}[l-1] + 0] = S_l[S_{l-2}[l-1]].$$

Thus the basic intuition is to use the path $S_0[S_0[l-1]] = f_{l-1} = -l$ to get

$$-l : S_0[S_0[l-1]] \xrightarrow{p_8} S_{l-2}[S_{l-2}[l-1]] \xrightarrow{p_9} S_l[S_{l-2}[l-1]]$$

In the above expression, $p_8 = \left(1 - \frac{1}{N}\right)^{l-2}$ and $p_9 = \left(1 - \frac{1}{N}\right)^2$ denote the probabilities of j not touching the state index that stores the value $-l$. This introduces a probability $\left(1 - \frac{1}{N}\right)^l$. Thus $\Pr(S_l[S_{l-2}[l-1]] = -l \mid f_{l-1} = -l)$ is cumulatively given by $\frac{1}{N} + \left[1 - \frac{1}{N}\right]^{l+1} \Pr(S_0[S_0[l-1]] = f_{l-1}) = p_{10}$, say. Note that one of the preconditions to prove [26, Theorem 4] is that the first $(l-1)$ places of state S_0^K remain untouched by j^K for the first $l-1$ rounds of KSA. This partially matches with the precondition to prove $\Pr(S_l[j_l] = 0 \mid f_{l-1} = -l)$ (see Theorem 1), where we require the same for first l places over the first l rounds of KSA. Thus we derive the formula for $\Pr(S_l[j_l] = 0 \mid S_0[S_0[l-1]] = -l \ \wedge \ f_{l-1} = -l)$ by modifying the result of Theorem 1 as $\frac{1}{N} + \left[1 - \frac{l}{N}\right]\left[1 - \frac{1}{N}\right]^{N+l-2}\left[\left[1 - \frac{1}{N}\right]^{1+l} + \frac{1}{N}\right] = p_{11}$, say. The final probability for $(Z_l = -l \mid f_{l-1} = -l)$ can now be computed as

$$\Pr(Z_l = -l \mid f_{l-1} = -l) = p_{10}p_{11} + (1 - p_{10}p_{11}) \cdot \frac{1}{N} = \frac{1}{N} + \left(1 - \frac{1}{N}\right) \cdot p_{10}p_{11}.$$

Substituting appropriate values for p_{10} and p_{11}, we get the desired result. □

Let us consider $\Pr(Z_l = -l \mid S_l[j_l] = 0) = \Pr(S_l[S_{l-2}[l-1]] = -l \mid S_l[j_l] = 0)$. From the proof of Theorem 3, it is evident that the events $(S_l[S_{l-2}[l-1]] = -l)$ and $(S_l[j_l] = 0)$ have no obvious connection. Yet, there exists a strong correlation between them, possibly due to some hidden events that cause them to co-occur with a high probability. We found that one of these hidden events is $(f_{l-1} = -l)$.

From the proofs of Theorems 1 and 3, we know that both the aforementioned events depend strongly on $(f_{l-1} = -l)$, but along two different paths, as follows.

$$0 : S_0^K[0] \xrightarrow{1} S_1^K[K[0]] \xrightarrow{p_1} S_l^K[K[0]] \xrightarrow{p_2} S_{l+1}^K[l] \xrightarrow{p_3} S_{l-1}[l] \xrightarrow{1} S_l[j_l]$$
$$-l : S_0^K[S_0^K[l-1]] \xrightarrow{p_{12}} S_0[S_0[l-1]] \xrightarrow{p_8} S_{l-2}[S_{l-2}[l-1]] \xrightarrow{p_9} S_l[S_{l-2}[l-1]]$$

Here p_{12} depends on the probability $\Pr(S_0[S_0[l-1]] = f_{l-1})$ from Proposition 3. Using these two paths, one may obtain the value of $\Pr(Z_l = -l \ \wedge \ S_l[j_l] = 0)$ as

$$\begin{aligned}\Pr(Z_l = -l \ \wedge \ S_l[j_l] = 0) &= \Pr(f_{l-1} = -l) \cdot \Pr(S_l[S_{l-2}[l-1]] = -l \ \wedge \ S_l[j_l] = 0 \mid f_{l-1} = -l) \\ &\quad + \Pr(f_{l-1} \neq -l) \cdot \Pr(S_l[S_{l-2}[l-1]] = -l \ \wedge \ S_l[j_l] = 0 \mid f_{l-1} \neq -l).\end{aligned}$$

As before, $\Pr(f_{l-1} = -l)$ can be taken as $1/N$. If one assumes that the aforementioned two paths are independent, the probabilities from Theorems 1 and 3 can be substituted in the above expression. If one further assumes that the events occur uniformly at random if $f_{l-1} \neq -l$, the values of $\Pr(S_l[j_l] = 0 \mid Z_l = -l)$ and $\Pr(Z_l = -l \mid S_l[j_l] = 0)$ turn out to be approximately $5/N$ each (for $l = 16$).

However, our experiments show that the two paths mentioned earlier are *not entirely independent*, and we obtain $\Pr(Z_l = -l \ \wedge \ S_l[j_l] = 0 \mid f_{l-1} = -l) \approx 5/N$. Moreover, the events are *not uniformly random* if $f_{l-1} \neq -l$; rather they are considerably biased for a range of values of f_{l-1} around $-l$ (e.g., for values like $-l + 1$, $-l + 2$ etc.). These hidden paths contribute towards the probability $\Pr(f_{l-1} \neq -l) \Pr(Z_l = -l \wedge S_l[j_l] = 0 \mid f_{l-1} \neq -l) \approx 5/N^2$. Through a careful treatment of the dependences and all the hidden paths, one would be able to justify the above observations, and obtain

$$\Pr(S_l[j_l] = 0 \mid Z_l = -l) \approx \Pr(Z_l = -l \mid S_l[j_l] = 0) \approx 10/N.$$

Similar techniques for analyzing dependences and hidden paths would work for all correlations reported in Equations (1), (2), (3), (4) and, (5).

We now shift our focus to $\Pr(Z_l = -l \mid f_{l-1} = -l)$ and its implications.

First of all, notice that the value of $\Pr(Z_l = -l \mid f_{l-1} = -l)$ depends on the value of $\Pr(S_0[S_0[l-1]] = f_{l-1})$. Proposition 3 gives an explicit formula for $\Pr(Z_l = -l \mid f_{l-1} = -l)$ for l up to 32. As l increases beyond 32, one may check by experimentation that this probability converges approximately to $1/N$. Thus, for $1 \leq l \leq 32$, one can use the formula from Proposition 3, and for $l > 32$, one may replace $\Pr(S_0[S_0[l-1]] = f_{l-1})$ by $1/N$ to approximately compute the distribution of $(Z_l = -l \mid f_{l-1} = -l)$ completely. In fact, after the state recovery attack by Maximov and Khovratovich [19], that is of time complexity around 2^{241}, choosing a secret key of length $l > 30$ is not meaningful. The value of $\Pr(Z_l = -l \mid f_{l-1} = -l)$ for some typical values of l are

$$12/N \ \text{ for } \ l = 5, \qquad 11/N \ \text{ for } \ l = 8, \qquad 7/N \ \text{ for } \ l = 16, \qquad 2/N \ \text{ for } \ l = 30.$$

In the list above, each conditional probability is quite high in magnitude compared to the natural probability of random occurrence. We try to exploit this bias in the next section to predict the length of RC4 secret key.

2.3 Keylength prediction from keystream

The huge conditional bias proved in Theorem 3 hints that there may be a related unconditional bias present in the event $Z_l = -l$ as well. In fact, New_007 in [29, Fig. 5] reports a bias in $(Z_i = -i)$ for $i = 0 \bmod 16$. The reported bias for $i = 16$ is $1.0411/N$. Notice that almost all experiments of [29] used the keylength $l = 16$, which encourages our speculation for an unconditional bias in $(Z_l = -l)$ for any general keylength l of RC4 secret key. Systematic investigation in this direction reveals the following result.

Theorem 4. *Suppose that l is the length of the secret key of RC4. The probability $\Pr(Z_l = -l)$ is given by*

$$\Pr(Z_l = -l) \approx \frac{1}{N} + [N \cdot \Pr(Z_l = -l \mid f_{l-1} = -l) - 1] \cdot \frac{1}{N^2}.$$

Proof. We provide a quick sketch of the proof to obtain a crude approximation of this bias in Z_l. Notice that we already have a path $(Z_l = -l \mid f_{l-1} = -l)$ with probability calculated in Theorem 3.

If we assume that for all other values of $f_{l-1} \neq -l$, the output Z_l can take the value $-l$ uniformly at random, we have

$$\begin{aligned}\Pr(Z_l = -l) \approx &\Pr(f_{l-1} = -l) \cdot \Pr(Z_l = -l \mid f_{l-1} = -l) \\ &+ \Pr(f_{l-1} \neq -l) \cdot \Pr(Z_l = -l \mid f_{l-1} \neq -l) \\ &= \frac{1}{N} \cdot \Pr(Z_l = -l \mid f_{l-1} = -l) + \left(1 - \frac{1}{N}\right) \cdot \frac{1}{N} \\ &= \frac{1}{N} + [N \cdot \Pr(Z_l = -l \mid f_{l-1} = -l) - 1] \cdot \frac{1}{N^2}.\end{aligned}$$

Thus we obtain the desired result. □

We have a closed form expression for $\Pr(Z_l = -l \mid f_{l-1} = -l)$ from Theorem 3 in cases where $1 \leq l \leq 32$ (using Proposition 3). We have also calculated some numerical values of this probability for $l = 5, 8, 16, 30$ and $N = 256$. Using those numeric approximations, the value of $\Pr(Z_l = -l)$ is

$$\begin{aligned} &1/N + 11/N^2 \text{ for } l = 5, && 1/N + 10/N^2 \text{ for } l = 8, \\ &1/N + 6/N^2 \text{ for } l = 16, && 1/N + 2/N^2 \text{ for } l = 30. \end{aligned}$$

The lower bound for $\Pr(Z_l = -l)$ within the typical range of keylength ($5 \leq l \leq 30$) is approximately $1/N + 1/N^2$, which is quite high and easily detectable. In experiments with 100 million runs and different keylengths, we have found that the probabilities are even higher than those mentioned above. This helps us in predicting the length of the secret key from the output, as follows.

1. Find the output byte Z_x biased towards $-x$. This requires $O(N^3)$ many samples as the bias is $O(1/N^2)$ for a base event with probability $1/N$ (recall the discussion in Section 1.4). A 'sample' in this case means the observation of keystream bytes Z_x for all $5 \leq x \leq 30$ for a specific key. The bias is computed by examining these keystream bytes with different keys, which are all of the same length l, say.
2. Check if the probability $\Pr(Z_x = -x)$ is equal or greater than the value proved in Theorem 4.
3. If the above statements hold for some $5 \leq x \leq 30$, the keylength can be predicted as $l = x$.

Although the bias in $Z_l = -l$ has been noticed earlier in the literature for specific keylengths, no attempts have been made for its generalization. Moreover, to the best of our knowledge, the prediction of keylength from the keystream has never been attempted. We have performed extensive experiments with varying keylengths to verify the practical feasibility of the prediction technique. This prediction technique proves to be successful for all keylengths within the typical usage range $5 \leq l \leq 30$. As already pointed out in Section 2.2, choosing a secret key of length $l > 30$ is not recommended. So, our *keylength prediction* effectively works for all practical values of the keylength.

3 Biases in initial rounds of RC4 PRGA

In this section, we study the short-term biases in RC4 keystream. First we discuss and prove some empirically observed biases involving the state variables and then we continue exploring the biases that involve only the keystream bytes.

3.1 Biases involving the state variables

In this section, we investigate some significant empirical biases discovered and reported in [29]. We provide theoretical justification only for the biases which are of the approximate order of $2/N$ or more, summarized in Table 1. Note that the authors of [29] denote the PRGA variables by primed indices. Moreover, the probabilities mentioned in the table are the ones observed in [29], and the values for 'biases at all rounds (round-dependent)' are the ones for $r = 3$. We provide general proofs and formulas for all of these biases.

Table 1. Significant biases observed in [29] and proved in this paper.

Type of Bias	Label as in [29]	Event	Probability
Bias at Specific Initial Rounds	New_004	$j_2 + S_2[j_2] = S_2[i_2] + Z_2$	$2/N$
	New_noz_007	$j_2 + S_2[j_2] = 6$	$2.37/N$
	New_noz_009	$j_2 + S_2[j_2] = S_2[i_2]$	$2/N$
	New_noz_014	$j_1 + S_1[i_1] = 2$	$1.94/N$
Bias at All Rounds (round-independent)	New_noz_001	$j_r + S_r[i_r] = i_r + S_r[j_r]$	$2/N$
	New_noz_002	$j_r + S_r[j_r] = i_r + S_r[i_r]$	$2/N$
Bias at All Rounds (round-dependent)	New_000	$S_r[t_r] = t_r$	$1.9/N$ at $r = 3$
	New_noz_004	$S_r[i_r] = j_r$	$1.9/N$ at $r = 3$
	New_noz_006	$S_r[j_r] = i_r$	$2.34/N$ at $r = 3$

In this target list, general biases refer to the ones occurring in all initial rounds of PRGA ($1 \le r \le N-1$), whereas the specific ones have been reported only for rounds 1 and 2 of PRGA. We do not consider the biases reported for rounds 0 mod 16 in this section, as they are of order $1/N^2$ or less. For the proofs and numeric probability calculations in this section, we require [15, Theorem 6.2.1], restated as Proposition 4 below.

Proposition 4. *At the end of RC4 KSA, for* $0 \le u \le N-1$, $0 \le v \le N-1$,

$$\Pr(S_0[u] = v) = \begin{cases} \frac{1}{N}\left[\left(\frac{N-1}{N}\right)^v + \left(1 - \left(\frac{N-1}{N}\right)^v\right)\left(\frac{N-1}{N}\right)^{N-u-1}\right] & \text{if } v \le u; \\ \frac{1}{N}\left[\left(\frac{N-1}{N}\right)^{N-u-1} + \left(\frac{N-1}{N}\right)^v\right] & \text{if } v > u. \end{cases}$$

If a pseudo-random permutation is taken as the initial state S_0 of RC4 PRGA, then we would have $\Pr(S_0[u] = v) = 1/N$ for all $0 \le u \le N-1$ and $0 \le v \le N-1$.

3.1.1 Bias at specific initial rounds

In this part of the paper, we prove the biases labeled New_noz_014, New_noz_007, New_noz_009 and New_004, as in [29, Fig. 3 and Fig. 4] and Table 1.

Theorem 5. *After the first round* ($r = 1$) *of RC4 PRGA,*

$$\Pr(j_1 + S_1[i_1] = 2) = \Pr(S_0[1] = 1) + \sum_{X \neq 1} \Pr(S_0[X] = 2 - X \ \wedge \ S_0[1] = X)$$

Proof. Note that $j_1 = S_0[1]$ and $S_1[i_1] = S_0[j_1]$. So, in the case $j_1 = S_0[1] = 1$, we will have $j_1 + S_0[j_1] = S_0[1] + S_0[1] = 2$ with probability 1. Otherwise, the probability turns out to be $\Pr(j_1 + S_0[j_1] = 2 \ \wedge \ j_1 = S_0[1] \neq 1) = \sum_{X \neq 1} \Pr(X + S_0[X] = 2 \ \wedge \ S_0[1] = X)$. Thus, the probability $\Pr(j_1 + S_1[i_1] = 2)$ can be written as

$$\Pr(j_1 + S_1[i_1] = 2) = \Pr(S_0[1] = 1) + \sum_{X \neq 1} \Pr(S_0[X] = 2 - X \wedge S_0[1] = X),$$

as desired. Hence the claimed result. □

If we consider the RC4 permutation after the KSA, the probabilities involving S_0 in the expression for $\Pr(j_1 + S_1[i_1] = 2)$ should be evaluated using Proposition 4 and the joint probability should be estimated in the same manner as in Section 3.2.3, giving a total probability of approximately $1.937/N$ for $N = 256$. This closely matches the observed value $1.94/N$. If we assume that RC4 PRGA starts with a truly pseudo-random initial state S_0, the probability turns out to be approximately $2/N - 1/N^2 \approx 1.996/N$ for $N = 256$, i.e., almost twice that of a random occurrence.

Theorem 6. *After the second round ($r = 2$) of RC4 PRGA, the following probability relations hold between the index j_2 and the state variables $S_2[i_2], S_2[j_2]$.*

$$\Pr\left(j_2 + S_2[j_2] = 6\right) \approx \Pr(S_0[1] = 2) + \sum_{X \text{ even}, X \neq 2} (2/N) \cdot \Pr(S_0[1] = X) \tag{6}$$

$$\Pr\left(j_2 + S_2[j_2] = S_2[i_2]\right) \approx 2/N - 1/N^2 \tag{7}$$

$$\Pr\left(j_2 + S_2[j_2] = S_2[i_2] + Z_2\right) \approx 2/N - 1/N^2 \tag{8}$$

Proof. In Equation (6), we have $j_2 + S_2[j_2] = (j_1 + S_1[2]) + S_1[i_2] = S_0[1] + 2 \cdot S_1[2]$. In this expression, note that if $S_0[1] = 2$, then one must have the positions 1 and 2 swapped in the first round of PRGA, and thus $S_1[2] = S_0[1] = 2$ as well. This provides one path for $j_2 + S_2[j_2] = S_0[1] + 2 \cdot S_1[2] = 2 + 2 \times 2 = 6$, with probability $\Pr(S_0[1] = 2) \cdot 1 \approx \frac{1}{N}$. If on the other hand, $S_0[1] = X \neq 2$, we have $\Pr(j_2 + S_2[j_2] = 6 \ \wedge \ S_0[1] \neq 2) = \sum_{X \neq 2} \Pr(X + 2 \cdot S_1[2] = 6 \ \wedge \ S_0[1] = X)$. Note that the value of X is bound to be even and for each such value of X, the variable $S_1[2]$ can take 2 different values to satisfy the equation $2 \cdot S_1[2] = 6 - X$. Thus, we have

$$\sum_{X \neq 2} \Pr(2 \cdot S_1[2] = 6 - X \ \wedge \ S_0[1] = X) \approx \sum_{X \text{ even}, X \neq 2} \frac{2}{N} \cdot \Pr(S_0[1] = X).$$

Combining the two disjoint cases $S_0[1] = 2$ and $S_0[1] \neq 2$, we get Equation (6).

In case of Equation (7), we have a slightly different condition $S_0[1] + 2 \cdot S_1[2] = S_2[i_2] = S_1[j_2] = S_1[S_0[1] + S_1[2]]$. In this expression, if we have $S_1[2] = 0$, then the left hand side reduces to $S_0[1]$ and the right hand side becomes $S_1[S_0[1] + S_1[2]] = S_1[S_0[1]] = S_1[j_1] = S_0[i_1] = S_0[1]$ as well. This provides a probability $1/N$ path for the condition to be true. In all other cases with $S_1[2] \neq 0$, we can approximate the probability for the condition as $1/N$, and hence approximate the total probability $\Pr(j_2 + S_2[j_2] = S_2[i_2])$ as

$$\begin{aligned}\Pr(j_2 + S_2[j_2] = S_2[i_2] \ \wedge \ S_1[2] = 0) + \Pr(j_2 + S_2[j_2] = S_2[i_2] \ \wedge \ S_1[2] \neq 0) \\ \approx \frac{1}{N} + \left(1 - \frac{1}{N}\right) \cdot \frac{1}{N} = \frac{2}{N} - \frac{1}{N^2}.\end{aligned}$$

Finally, for Equation (8), the main observation is that this is almost identical to the condition of Equation (7) apart from the inclusion of Z_2. But our first path of $S_1[2] = 0$ in the previous

case also provides us with $Z_2 = 0$ with probability 1 (this path was first observed by Mantin and Shamir [16]). Thus, we have $\Pr(j_2 + S_2[j_2] = S_2[i_2] + Z_2 \ \wedge \ S_1[2] = 1) \approx \frac{1}{N} \cdot 1$. In all other cases with $S_1[2] \neq 0$, we assume the conditions to match uniformly at random, and therefore have

$$\Pr(j_2 + S_2[j_2] = S_2[i_2] + Z_2) \approx \frac{1}{N} \cdot 1 + \left(1 - \frac{1}{N}\right) \cdot \frac{1}{N} = \frac{2}{N} - \frac{1}{N^2}.$$

Hence the desired results of Equations (6), (7) and (8). □

In case of Equation (6), if we assume S_0 to be the initial state for RC4 PRGA, and substitute all probabilities involving S_0 using Proposition 4, we get the total probability equal to $2.36/N$ for $N = 256$. This value closely match the observed probability $2.37/N$. If we suppose that S_0 is pseudo-random, we will get probability $2/N - 2/N^2 \approx 1.992/N$ for Equation (6). The theoretical results are summarized in Table 2 along with the experimentally observed probabilities of [29].

Table 2. Theoretical and observed biases at specific initial rounds of RC4 PRGA.

Label [29]	Event	Observed Probability (reported in [29])	Theoretical Probability	
			S_0 of RC4	Random S_0
New_noz_014	$j_1 + S_1[i_1] = 2$	$1.94/N$	$1.937/N$	$1.996/N$
New_noz_007	$j_2 + S_2[j_2] = 6$	$2.37/N$	$2.363/N$	$1.992/N$
New_noz_009	$j_2 + S_2[j_2] = S_2[i_2]$	$2/N$	$1.996/N$	$1.996/N$
New_noz_004	$j_2 + S_2[j_2] = S_2[i_2] + Z_2$	$2/N$	$1.996/N$	$1.996/N$

3.1.2 Round-independent biases at all initial rounds

In this section, we turn our attention to the biases labeled New_noz_001 and New_noz_002 in [29], both of which continue to persist in all initial rounds ($1 \leq r \leq N - 1$) of RC4 PRGA.

Theorem 7. *At any initial round $1 \leq r \leq N - 1$ of RC4 PRGA, the following two relations hold between the indices i_r, j_r and the state variables $S_r[i_r], S_r[j_r]$.*

$$\Pr(j_r + S_r[j_r] = i_r + S_r[i_r]) \approx 2/N \tag{9}$$

$$\Pr(j_r + S_r[i_r] = i_r + S_r[j_r]) \approx 2/N \tag{10}$$

Proof. For both the events mentioned above, we shall take the path $i_r = j_r$. Notice that $i_r = j_r$ occurs with probability $1/N$ and in that case both the events mentioned above hold with probability 1. In the case where $i_r \neq j_r$, we rewrite the events as $S_r[j_r] = (i_r - j_r) + S_r[i_r]$ and $S_r[j_r] = (j_r - i_r) + S_r[i_r]$. Here we already know that $S_r[j_r] \neq S_r[i_r]$, as $j_r \neq i_r$ and S_r is a permutation. Thus in case $i_r \neq j_r$, the values of $S_r[i_r]$ and $S_r[j_r]$ can be chosen in $N(N-1)$ ways (drawing from a permutation without replacement) to satisfy the relations stated above. This gives the total probability for each event approximately as

$$\Pr(j_r = i_r) \cdot 1 + \sum_{j_r \neq i_r} \frac{1}{N(N-1)} = \frac{1}{N} + (N-1) \cdot \frac{1}{N(N-1)} = \frac{2}{N}.$$

Hence the claimed result for Equations (9) and (10). □

The probabilities for New_noz_001 and New_noz_002 proved in Theorem 7 do not vary with change in r (i.e., they continue to persist at the same order of $2/N$ at any arbitrary round of PRGA), and our theoretical results match the probabilities reported in [29, Fig. 2].

3.1.3 Round-dependent biases at all initial rounds

Next, we consider the biases that are labeled as New_000, New_noz_004 and New_noz_006 in [29, Fig. 2]. We prove the biases for rounds 3 to 255 in RC4 PRGA, and we show that all of these decrease in magnitude with increase in r, as observed experimentally in the original paper. Before proving the observation New_noz_006 of [29] in Theorem 8, let us first prove a necessary technical result.

Lemma 1. *After the first round of RC4 PRGA, the probability* $\Pr(S_1[t] = r)$ *is*

$$\Pr(S_1[t] = r) = \begin{cases} \sum_{X=0}^{N-1} \Pr(S_0[1] = X \ \wedge \ S_0[X] = r), & t = 1; \\ \Pr(S_0[1] = r) + \sum_{w \neq r} \Pr(S_0[1] = w \ \wedge \ S_0[r] = r), & t = r; \\ \sum_{w \neq t} \Pr(S_0[1] = w \ \wedge \ S_0[t] = r), & t \neq 1, r. \end{cases}$$

Proof. After the first round of RC4 PRGA, we obtain the state S_1 from the initial state S_0 through a single swap operation between the positions $i_1 = 1$ and $j_1 = S_0[i_1] = S_0[1]$. Thus, all other positions of S_0 remain the same apart from these two. This gives us the value of $S_1[t]$ as follows: $S_1[t] = S_0[S_0[1]]$ if $t = 1$, $S_1[t] = S_0[1]$ if $t = S_0[1]$, and $S_1[t] = S_0[t]$ in all other cases. Now, we can compute the probabilities $\Pr(S_1[t] = r)$ based on the probabilities for S_0, which are in turn derived from Proposition 4. We have three cases:

- Case $t = 1$. In this case, using the recurrence relation $S_1[1] = S_0[S_0[1]]$, we can write

$$\Pr(S_1[1] = r) = \sum_{X=0}^{N-1} \Pr(S_0[1] = X \ \wedge \ S_0[X] = r).$$

- Case $t = r$. In this situation, if $S_0[1] = r$, we will surely have $S_1[r] = r$ as these are the positions swapped in the first round, and if $S_0[1] \neq r$, the position $t = r$ remains untouched and $S_1[r] = r$ is only possible if $S_0[r] = r$. Thus,

$$\Pr(S_1[r] = r) = \Pr(S_0[1] = r) + \Pr(S_0[1] \neq r \ \wedge \ S_0[r] = r).$$

- Case $t \neq 1, r$. In all other cases where $t \neq 1, r$, it can either take the value $S_0[1]$ with probability $\Pr(S_0[1] = t)$, or not. If $t = S_0[1]$, the value $S_0[t]$ will get swapped with $S_0[1] = t$ itself, i.e., we will get $S_1[t] = t \neq r$ for sure. Otherwise, the value $S_1[t]$ remains the same as $S_0[t]$. Hence,

$$\Pr(S_1[t] = r) = \Pr(S_0[1] \neq t \ \wedge \ S_0[t] = r).$$

Combining all the above cases together, we obtain the desired result. □

Note that estimation of the joint probabilities in Lemma 1 should be done using Proposition 4 in the same manner as in Section 3.2.3. Now we can state and prove the main result.

Theorem 8. *For PRGA rounds* $r \geq 3$*, value of* $\Pr(S_r[j_r] = i_r)$ *is approximately*

$$\Pr(S_1[r] = r) \left[1 - \frac{1}{N}\right]^{r-2} + \sum_{t=2}^{r-1} \sum_{k=0}^{r-t} \frac{\Pr(S_1[t] = r)}{k! \cdot N} \left[\frac{r-t-1}{N}\right]^k \left[1 - \frac{1}{N}\right]^{r-3-k}$$

Proof. Let us start from the PRGA state S_1, that is, the state that has been updated once in the PRGA (we refer to the state after KSA by S_0). We know that the event $\Pr(S_1[r] = r)$ is positively biased for all r, and hence the natural path for investigation is the effect of the event $(S_1[r] = r)$ on $(S_{r-1}[r] = r)$, i.e, on $(S_r[j_r] = i_r)$. Notice that there can be two cases, as follows.

Case I. In the first case, suppose that $(S_1[r] = r)$ after the first round, and the r-th index is not disturbed for the next $r - 2$ state updates. Notice that index i varies from 2 to $r - 1$ during these period, and hence never touches the r-th index. Thus, the index r will retain its state value r if index j does not touch it. The probability of this event is $\left(1 - \frac{1}{N}\right)^{r-2}$ over all the intermediate rounds. Hence the first part of the probability is

$$\Pr(S_1[r] = r) \left(1 - \frac{1}{N}\right)^{r-2}.$$

Case II. In the second case, suppose that $S_1[r] \neq r$ and $S_1[t] = r$ for some $t \neq r$. In such a case, only a swap between the positions r and t during rounds 2 to $r - 1$ of PRGA can make the event $(S_{r-1}[r] = r)$ possible. Notice that if t does not fall in the *path of* i, that is, if the index i does not touch the t-th location, then the value at $S_1[t]$ can only go to some position behind i, and this can never reach $S_{r-1}[r]$, as i can only go up to $(r - 1)$ during this period. Thus we must have $2 \leq t \leq r - 1$ for $S_1[t]$ to reach $S_{r-1}[r]$. Note that the way $S_1[t]$ can move to the r-th position may be either a one hop or a multi-hop route.

- In the easiest case of single hop, we require j not to touch t until i touches t, and $j = r$ when $i = t$, and j not to touch r for the next $r - t - 1$ state updates. Total probability comes to be

$$\Pr(S_1[t] = r) \left(1 - \frac{1}{N}\right)^{t-2} \cdot \frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^{r-t-1} = \Pr(S_1[t] = r) \cdot \frac{1}{N} \left(1 - \frac{1}{N}\right)^{r-3}.$$

- Suppose that it requires $(k+1)$ hops to reach from $S_1[t]$ to $S_{r-1}[r]$. Then the main issue to note is that the transfer will never happen if the position t swaps with any index which does not lie in the future *path of* i. Again, this path of i starts from $\frac{r-t-1}{N}$ for the first hop and decreases approximately to $\frac{r-t-1}{lN}$ at the l-th hop. We would also require j not to touch the position r for the remaining $(r - 3 - k)$ number of rounds. Combining all, we get the second part of the probability as

$$\Pr(S_1[t] = r) \left[\prod_{l=1}^{k} \frac{r-t-1}{lN}\right] \left[1 - \frac{1}{N}\right]^{r-3-k} = \frac{\Pr(S_1[t] = r)}{k! \cdot N} \left[\frac{r-t-1}{N}\right]^{k} \left[1 - \frac{1}{N}\right]^{r-3-k}.$$

Finally, note that the number of hops $(k + 1)$ is bounded from below by 1 and from above by $(r - t + 1)$, depending on the initial gap between t and r positions. Considering the sum over t and k with this consideration, we get the desired expression for $\Pr(S_{r-1}[r] = r)$. □

Remark 1. In proving Theorem 8, we use the initial condition $S_1[r] = r$ to branch out the probability paths, and not $S_0[r] = r$ as in [14, Lemma 1]. This is because the probability of $S[r] = r$ takes a leap from around $1/N$ in S_0 to about $2/N$ in S_1, and this turns out to be the actual cause behind the bias in $S_{r-1}[r] = r$. The correction by moving to the base distribution of S_1 from that of S_0 eventually corrects the mismatches observed in the graphs of [14]. We shall discuss this issue in more details in Section 3.3.

Fig. 3 illustrates the experimental observations (each data point represents the average obtained from over 100 million experimental runs with 16-byte key in each case) and the theoretical values for the distribution of $\Pr(S_r[j_r] = i_r)$ over the initial rounds $3 \leq r \leq 255$ of RC4 PRGA. It is evident that our theoretical formula, as proved in Theorem 8, matches the experimental observations.

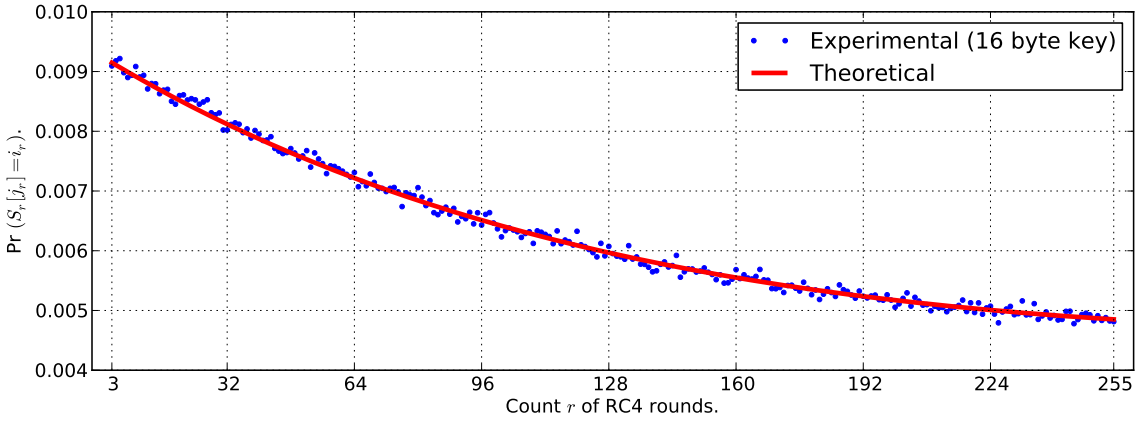

Fig. 3. Distribution of $\Pr(S_r[j_r] = i_r)$ for initial rounds $3 \leq r \leq 255$ of RC4 PRGA.

Next we take a look at the other two round-dependent biases of RC4, observed in [29]. We can state the related result in Theorem 9 (corresponding to observations New_noz_004 and New_000).

Theorem 9. *For PRGA rounds* $r \geq 3$*, the probabilities* $\Pr(S_r[i_r] = j_r)$ *and* $\Pr(S_r[t_r] = t_r)$ *are approximately*

$$\frac{r}{N^2} + \sum_{X=r}^{N-1} \frac{1}{N} \left[\Pr(S_1[X] = X) \left[1 - \frac{1}{N}\right]^{r-2} + \sum_{t=2}^{r-1} \sum_{k=0}^{r-t} \frac{\Pr(S_1[t] = r)}{k! \cdot N} \left[\frac{r-t-1}{N}\right]^k \left[1 - \frac{1}{N}\right]^{r-3-k} \right]$$

The proof of this result follows the same logic as in the proof of Theorem 8. A proof sketch is presented as follows. For this proof sketch, we consider the variables j_r and t_r to be pseudo-random variables that can take any value between 0 to 255 with probability $1/N$. The reader may note that this is a crude approximation, especially for small values of r, and causes minor mismatch with the experimental observations in the final result.

Proof-sketch for $\Pr(S_r[i_r] = j_r)$*:* For this probability computation, we first rewrite the event as $(S_{r-1}[j_r] = j_r)$ to make it look similar to $S_{r-1}[r] = r$, as in Theorem 8. The only difference is that we were concentrating on a fixed index r in Theorem 8 instead of a variable index j_r. This produces two cases.

Case I. First, suppose that j_r assumes a value $X \geq r$. In this case, the probability calculation can be split in two paths, one in which $S_1[X] = X$ is assumed, and the other in which $S_1[X] \neq X$. If we assume $S_1[X] = X$, the probability of $(S_{r-1}[X] = X)$ becomes $\Pr(S_1[X] = X) \left[1 - \frac{1}{N}\right]^{r-2}$, similar to the logic in Theorem 8. If we suppose that $S_1[t] = X$ was the initial state, then one may notice the following two sub-cases:

- The probability for this path is identical to that in Theorem 8 if $2 \leq t \leq r - 1$.
- The probability is 0 in case $t \geq r$, as in this case the value X will always be behind the position of $i_r = r$, whereas $X > r$ as per assumption, i.e., the value X can never reach index X from t.

Assuming $\Pr(j_r = X) = 1/N$, this gives

$$\sum_{X=r}^{N-1} \frac{1}{N} \left[\Pr(S_1[X] = X) \left[1 - \frac{1}{N}\right]^{r-2} + \sum_{t=2}^{r-1} \sum_{k=0}^{r-t} \frac{\Pr(S_1[t] = r)}{k! \cdot N} \left[\frac{r-t-1}{N}\right]^k \left[1 - \frac{1}{N}\right]^{r-3-k} \right].$$

Case II. In the second case, we assume that j_r takes a value X between 0 to $r-1$. Approximately this complete range is touched by index i for sure, and may also be touched by index j. Thus, with probability approximately 1, the index $j_r = X$ is touched by either of the indices. Simplifying all complicated computations involving the initial position of value X and the exact location of index X in this case, we shall assume that the approximate value of $\Pr(S_{r-1}[X] = X)$ is $1/N$. Thus, the total contribution of Case II, assuming $\Pr(j_r = X) = 1/N$, is given by

$$\sum_{X=0}^{r-1} \Pr(j_r = X) \cdot \Pr(S_{r-1}[X] = X) \approx \sum_{X=0}^{r-1} \frac{1}{N} \cdot \frac{1}{N} = \frac{r}{N^2}.$$

Adding the contributions of the two disjoint cases I and II, we obtain the total probability for $(S_r[i_r] = j_r)$ as desired. One may investigate Case II in more details to incorporate all intertwined sub-cases, and obtain a better closed form expression for the probability.

Proof-sketch for $\Pr(S_r[t_r] = t_r)$*:* In this case, notice that t_r is just another random variable like j_r, and may assume all values from 0 to 255 with approximately the same probability $1/N$. Thus we can approximate $\Pr(S_r[t_r] = t_r)$ by $\Pr(S_{r-1}[j_r] = j_r)$ with a high confidence margin to obtain the desired expression. This approximation is particularly close for higher values of r because the effect of a single state change $S_{r-1} \rightarrow S_r$ is low in such a case. For smaller values of r, one may approximate $\Pr(S_{r-1}[t_r] = t_r)$ by $\Pr(S_{r-1}[j_r] = j_r)$ and critically analyze the effect of the r-th round of PRGA thereafter. However, in spite of the approximations we made, one may note that the theoretical values closely match the experimental observations (averages taken over 100 million runs of RC4 with 16-byte key), as shown in Fig. 4.

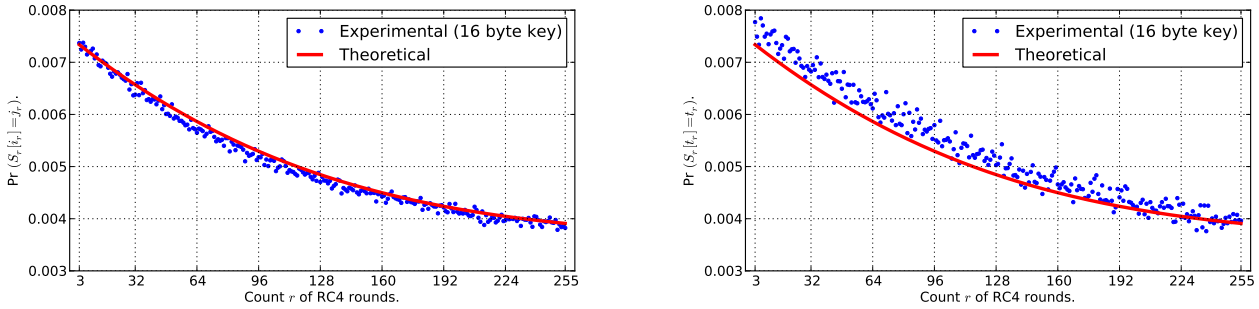

Fig. 4. Distributions of $\Pr(S_r[i_r] = j_r)$ and $\Pr(S_r[t_r] = t_r)$ for initial rounds $3 \leq r \leq 255$ of RC4 PRGA.

Fig. 4 illustrates the experimental observations (averages taken over 100 million runs with 16-byte key) and the theoretical values for the distributions of $\Pr(S_r[i_r] = j_r)$ and $\Pr(S_r[t_r] = t_r)$ over the initial rounds $3 \leq r \leq 255$ of RC4 PRGA. It is evident that our theoretical formulas approximately match the experimental observations in both the cases; the cause of the little deviation is explained in the proof sketch above.

Note: Apart from the biases proved so far, all other unconditional biases reported in [29] are of order $1/N^2$ or less, and we omit their analysis in this paper.

3.2 Probability distribution of Z_1

In this section, we theoretically derive the complete probability distribution of the first keystream byte Z_1 of RC4, as observed by Mironov [20, Fig. 6] in CRYPTO 2002. The first thing to note is that Z_1 has a negative bias towards the value 0. In fact, even if the initial permutation of PRGA, i.e., S_0, is taken to be a uniformly random permutation, this negative bias in $(Z_1 = 0)$ persists. After almost ten years, we present the theoretical proof of this observation in the next subsection.

3.2.1 Negative bias in Z_1 towards zero

Note that PRGA begins with $i_0 = j_0 = 0$. In the first round, $i_1 = 1$ and $j_1 = S_0[i_1] = S_0[1]$. In proving the main result, we shall utilize the existence of a special path in which Z_1 can never be equal to zero.

Theorem 10. *Assume that the initial permutation S_0 of RC4 PRGA is randomly chosen from the set of all permutations of $\{0, 1, \ldots, N-1\}$. Then the probability that the first output byte of RC4 keystream is 0 is approximately $1/N - 1/N^2$.*

Proof. First, suppose that $S_0[j_1] = S_0[S_0[1]] = 0$ and $S_0[1] \neq 1$. In this case, the first output byte Z_1 is 0 with probability 0 (see Fig. 5).

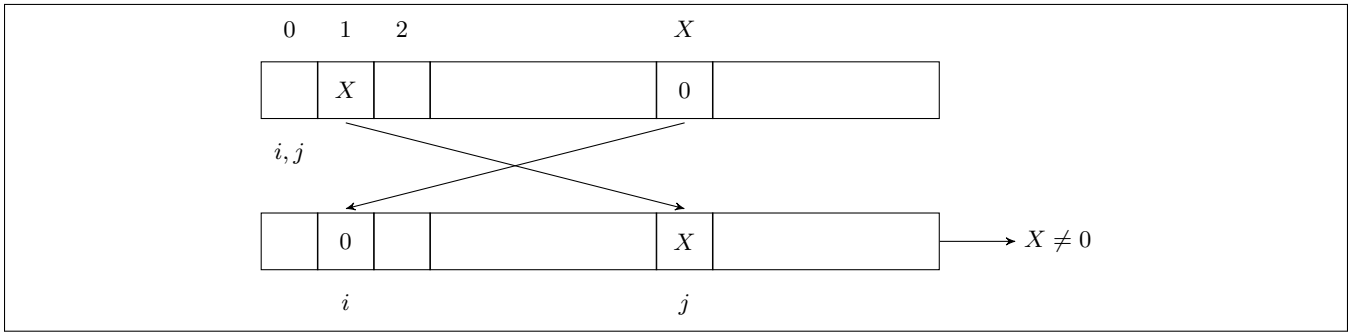

Fig. 5. The first round of RC4 when $S_0[S_0[1]] = 0$ and $S_0[1] \neq 1$.

Let j_1 be equal to $X \neq 1$. After the first swap, $S_1[1] = S_0[X] = 0$ and $S_1[X] = S_0[1] = X$. Now, $Z_1 = S_1[S_1[1] + S_1[X]] = S_1[0 + X] = X$. If Z_1 were 0, one must have $X = 0$. But this is not possible as X and 0 belong to two different locations in the initial permutation S_0, as shown in Fig. 5. Thus,

$$\Pr(Z_1 = 0 \mid S_0[j_1] = 0) = 0.$$

If $S_0[j_1] \neq 0$, output byte Z_1 can be considered uniformly random, and thus we have

$$\Pr(Z_1 = 0 \mid S_0[j_1] \neq 0) \approx 1/N.$$

As a result, the total probability that the first output byte is 0 is given by

$$\begin{aligned}\Pr(Z_1 = 0) &= \Pr(Z_1 = 0 \mid S_0[j_1] = 0) \cdot \Pr(S_0[j_1] = 0) \\ &\quad + \Pr(Z_1 = 0 \mid S_0[j_1] \neq 0) \cdot \Pr(S_0[j_1] \neq 0) \\ &\approx 0 \cdot 1/N + 1/N \cdot (1 - 1/N) = 1/N - 1/N^2.\end{aligned}$$

Thus, Z_1 has a negative bias of $1/N^2$ towards the value 0, as claimed. □

From Theorem 10, we immediately get a distinguisher of RC4 that can effectively distinguish the output keystream of the cipher from a random sequence of bytes. Let X and Y be the distributions corresponding to random stream and RC4 keystream respectively and define $E : (Z_1 = 0)$. Then the bias proved above can be written as $p(1 + q)$, where $p = 1/N$ and $q = -1/N$. According to Section 1.4, the number of samples required to distinguish RC4 from random sequence of bits with a constant probability of success in this case is $O(N^3)$.

3.2.2 Complete distribution of Z_1

In this section, we turn our attention to the complete probability distribution of the first byte Z_1. We can state and prove the main result in this direction as follows.

Theorem 11. *The probability distribution of the first output byte of RC4 keystream is*

$$\Pr(Z_1 = v) \approx \begin{cases} \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = 0) \\ \quad + \sum_{X \neq 0,1} \sum_{Y \in \mathcal{T}_0} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X + Y] = 0), & \text{if } v = 0; \\ \\ \Pr(S_0[1] = 0 \ \wedge \ S_0[0] = 1) \\ \quad + \sum_{X \neq 1} \sum_{Y \in \mathcal{T}_1} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X + Y] = 1), & \text{if } v = 1; \\ \\ \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) + \Pr(S_0[1] = v \ \wedge \ S_0[v] = 0) \\ \quad + \Pr(S_0[1] = 1 - v \ \wedge \ S_0[1 - v] = v) \\ \quad + \sum_{X \neq 1,v} \sum_{Y \in \mathcal{T}_v} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X + Y] = v), & \text{otherwise,} \end{cases}$$

where $v \in \{0, \ldots, N - 1\}$ denotes the output, set $\mathcal{T}_0$ denotes $\{0, 1, \ldots, N - 1\} \setminus \{0, X, 1 - X\}$, set $\mathcal{T}_1$ denotes $\{0, 1, \ldots, N - 1\} \setminus \{0, 1, X, 1 - X\}$, set $\mathcal{T}_v$ denotes $\{0, 1, \ldots, N - 1\} \setminus \{0, X, 1 - X, v\}$, and all the sums range over 0 to $N - 1$, unless otherwise specified.

Proof. The first output byte Z_1 can be explicitly written as follows:

$$Z_1 = S_1[S_1[i_1] + S_1[j_1]] = S_1[S_0[j_1] + S_0[i_1]] = S_1[S_0[S_0[1]] + S_0[1]] = S_1[Y + X], \text{ say,}$$

where we denote $j_1 = S_0[1]$ by X and $S_0[S_0[1]] = S_0[X]$ by Y. We know that state S_1 is different from S_0 in at most two places, $i_1 = 1$ and $j_1 = X$. Thus, we need to treat separately the cases $X + Y = i_1 = 1$ and $X + Y = j_1 = X$, as we have particular values of Z_1 in these cases, as follows.

$$\begin{aligned} (X + Y = X \ \Leftrightarrow \ Y = 0) \quad &\Rightarrow \quad Z_1 = S_1[X] = S_1[j_1] = S_0[i_1] = S_0[1] = X \\ (X + Y = 1 \ \Leftrightarrow \ Y = 1 - X) \quad &\Rightarrow \quad Z_1 = S_1[1] = S_1[i_1] = S_0[j_1] = S_0[X] = Y \end{aligned}$$

Moreover, if $X = 1$, there is no swap at the first round of RC4, and hence S_1 is identical to S_0. In this case, we have

$$(X = 1 \ \Leftrightarrow \ Y = X) \quad \Rightarrow \quad Z_1 = S_1[X + Y] = S_0[X + Y] = S_0[1 + 1] = S_0[2].$$

One may verify that this is an exhaustive list of special cases, and in all other circumstances, we would have $Z_1 = S_1[X + Y] = S_0[X + Y]$. Considering all the special cases as discussed above, we

obtain the cases for $\Pr(Z_1 = v)$ as follows.

$$\Pr(Z_1 = v) \approx \begin{cases} \Pr(S_0[2] = v) & \text{if } (X = 1 \Leftrightarrow Y = X); \\ \Pr(X = S_0[1] = v) & \text{if } (X + Y = X \Leftrightarrow Y = 0); \\ \Pr(Y = S_0[X] = v) & \text{if } (X + Y = 1 \Leftrightarrow Y = 1 - X); \\ \Pr(S_0[X + Y] = v) & \text{in all other cases.} \end{cases}$$

Note that $Y = 0$ fixes $X = v$ in the second case and $Y = 1 - X$ fixes $Y = v$, i.e., $X = 1 - v$ in the third case. Furthermore, we have two further restrictions based on the cases where X or Y equals the output v:

- If $X = v$ and $Y \neq 0$, then after the first swap, v resides in index v, but $X + Y \neq v$ in this case. This gives us an impossible pair $[v, Y]$ for all $Y \neq 0$.
- If $Y = v$ and $X \neq 1 - v$, then after the first swap, v resides in index 1, but $X + Y \neq 1$ for sure. So, this constitutes another impossible pair $[X, v]$ for all $X \neq 1 - v$.

Note that the only possible pairs $[v, 0]$ and $[1 - v, v]$ have already been considered earlier as special cases. Hence, the most general form for the probability can be written as

$$\begin{aligned} \Pr(Z_1 = v) \approx\ & \Pr(S_0[1] = 1 \ \wedge\ S_0[2] = v) + \Pr(S_0[1] = v \ \wedge\ S_0[v] = 0) \\ & + \Pr(S_0[1] = 1 - v \ \wedge\ S_0[1 - v] = v) \\ & + \sum_{X \neq 1, v} \sum_{Y \neq 0, X, 1-X, v} \Pr(S_0[1] = X \ \wedge\ S_0[X] = Y \ \wedge\ S_0[X + Y] = v) \end{aligned}$$

Now, state S_0 being a permutation, some of these probabilities are 0 when v takes particular values. So, we consider the following cases depending on the value of v.

Case I. When $v = 0$, the probability $\Pr(S_0[1] = v \ \wedge\ S_0[v] = 0) = \Pr(S_0[1] = 0 \ \wedge\ S_0[0] = 0)$ is 0 as two different places of the permutation S_0 can not hold the same value. Similarly, $Pr(S_0[1] = 1 - v \ \wedge\ S_0[1 - v] = v) = Pr(S_0[1] = 1 \ \wedge\ S_0[1] = 0) = 0$. Moreover, the condition $Y \neq 0$ takes into account $Y \neq v$, and thus we have

$$\begin{aligned} \Pr(Z_1 = 0) \approx\ & \Pr(S_0[1] = 1 \ \wedge\ S_0[2] = 0) \\ & + \sum_{X \neq 0, 1} \sum_{Y \neq 0, X, 1-X} \Pr(S_0[1] = X \ \wedge\ S_0[X] = Y \ \wedge\ S_0[X + Y] = 0) \end{aligned}$$

Case II. Suppose that $v = 1$. In this case $\Pr(S_0[1] = 1 \wedge S_0[2] = v) = \Pr(S_0[1] = 1 \wedge S_0[2] = 1) = 0$ and $\Pr(S_0[1] = v \ \wedge\ S_0[v] = 0) = \Pr(S_0[1] = 1 \ \wedge\ S_0[1] = 0) = 0$. Moreover, the conditions $X \neq 1$ and $X \neq v$ are identical. Thus we have

$$\begin{aligned} \Pr(Z_1 = 1) \approx\ & \Pr(S_0[1] = 0 \ \wedge\ S_0[0] = 1) \\ & + \sum_{X \neq 1} \sum_{Y \neq 0, 1, X, 1-X} \Pr(S_0[1] = X \ \wedge\ S_0[X] = Y \ \wedge\ S_0[X + Y] = 1) \end{aligned}$$

Case III. If $v \neq 0, 1$, we have no conflicts or special conditions as in the previous cases, and hence the general form of the probability holds.

Combining all three cases, we obtain the desired theoretical probability distribution for the first output byte Z_1, where we represent the condition $Y \neq 0, X, 1 - X$ by $Y \in \mathcal{T}_0$, condition $Y \neq 0, 1, X, 1 - X$ by $Y \in \mathcal{T}_1$, and condition $Y \neq 0, X, 1 - X, v$ by $Y \in \mathcal{T}_v$. □

3.2.3 Estimation of the joint probabilities and numeric values

We consider two special cases while computing the numeric values of $\Pr(Z_1 = v)$. First, we investigate RC4 PRGA where S_0 is fed from the output of RC4 KSA, as in practice. Next, we probe into the scenario when the initial permutation S_0 is pseudo-random. The latter case hints at the long-term manifestation of this short-term bias in the first byte.

Recall that the observed pattern in [20, Fig. 6] depicts a negative bias at $(Z_1 = 0)$, a slight negative bias at $(Z_1 = 1)$, and a 'sine-curve-like' distribution of probabilities $\Pr(Z_1 = v)$ for $2 \leq v \leq 255$. We observed that this sine-curve pattern in the probabilities is an outcome of the non-randomness in S_0 generated by the KSA routine, which can be modeled using Proposition 4.

Assume that the initial permutation S_0 of RC4 PRGA is constructed from the regular KSA, i.e., the probabilities $\Pr(S_0[u] = v)$ follow the distribution mentioned in Proposition 4. However, we require the joint probabilities like $\Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X + Y] = v)$ in our formula derived in Theorem 11, and these seem to be way more complicated to calculate in precise terms. So, we devise the following estimates for these joint probabilities.

- Consider the joint probability of two events: $\Pr(S_0[a] = X \ \wedge \ S_0[b] = Y)$ where $a \neq b$ and $X \neq Y$. Then we can represent this by

$$\Pr(S_0[a] = X \ \wedge \ S_0[b] = Y) = \Pr(S_0[a] = X) \cdot \Pr(S_0[b] = Y \mid S_0[a] = X).$$

  The first term is straight-forward to estimate from Proposition 4. For the second term, we note that given $S_0[a] = X$, it is impossible to have $S_0[b] = X$. However, the sum of the $N - 1$ conditionals $\Pr(S_0[b] = Y \mid S_0[a] = X)$ over all $Y \neq X$ would still be 1. Thus, the sum of the differences $\Delta P_Y = \Pr(S_0[b] = Y \mid S_0[a] = X) - \Pr(S_0[b] = Y)$ over all the $N - 1$ values of $Y \neq X$ would exactly balance the missing probability $\Pr(S_0[b] = X)$. For simplicity, we distribute $\Pr(S_0[b] = X)$ equally over all the $N - 1$ many ΔP_Y's and hence estimate the second term as

$$\Pr(S_0[b] = Y \mid S_0[a] = X) \approx \Pr(S_0[b] = Y) + \frac{\Pr(S_0[b] = X)}{N - 1}.$$

  Note that a special case of our estimation strategy when applied to uniformly random permutation gives the conditional $P(S_0[b] = Y \mid S_0[a] = X)$ as $\frac{1}{N} + \frac{1/N}{N-1} = \frac{1}{N-1}$, as expected.

- Similarly, for the joint probability of three events: $\Pr(S_0[a] = X \ \wedge \ S_0[b] = Y \ \wedge \ S_0[c] = Z)$, we can represent it by

$$\Pr(S_0[a] = X) \cdot \Pr(S_0[b] = Y \mid S_0[a] = X) \cdot \Pr(S_0[c] = Z \mid S_0[b] = Y \ \wedge \ S_0[a] = X).$$

  While the first term is straight-forward to estimate from Proposition 4, and the second term is estimated as before, the third term can be approximated as

$$\Pr(S_0[c] = Z \mid S_0[b] = Y \ \wedge \ S_0[a] = X) \approx \Pr(S_0[c] = Z) + \frac{\Pr(S_0[c] = Y)}{N - 2} + \frac{\Pr(S_0[c] = X)}{N - 2},$$

  so that the sum of all conditional probabilities over $Z \neq X, Y$ gives 1, as expected.

Then the theoretical values of $\Pr(Z_1 = v)$, calculated using Theorem 11 and Proposition 4, along with the estimations for joint probabilities discussed above, closely match the experimental observations. Fig. 6 shows the theoretical and experimental probability distributions of Z_1, where the experimental data is generated over 100 million runs of RC4 PRGA using 16 byte secret keys. The figure clearly shows that our theoretical justification closely matches the data obtained from the experiments. This further justifies the observation by Mironov [20].

Remark 2. As an alternative to the *additive correction* described above for estimating the conditionals, one may consider *multiplicative correction* by normalizing the probabilities as follows:

- Estimate $\Pr(S_0[b] = Y \mid S_0[a] = X)$ as $\frac{\Pr(S_0[b]=Y)}{1-\Pr(S_0[b]=X)}$.
- Estimate $\Pr(S_0[c] = Z \mid S_0[b] = Y \wedge S_0[a] = X)$ as $\frac{\Pr(S_0[c]=Z)}{1-\Pr(S_0[c]=Y)-\Pr(S_0[c]=X)}$.

Note that for uniformly random permutation, $\Pr(S_0[b] = Y \mid S_0[a] = X) = \frac{1/N}{1-1/N} = \frac{1}{N-1}$, as expected. However, we find that the final numeric values of $\Pr(Z_1 = v)$ estimated using the two different models (additive and multiplicative) almost coincide and the graphs fall right on top of one another. The same holds for all the probability expressions in this paper that depend on similar conditional estimates.

If S_0 is pseudo-random, our experiments also illustrated the fact that the sine-curve variation in probabilities for Z_1 is not present if RC4 PRGA starts with a pseudo-random initial permutation S_0. This can be theoretically justified as follows.

If the initial permutation S_0 of RC4 PRGA is considered to be pseudo-random, then we would have $p_0[u, v] = \Pr(S_0[u] = v) \approx 1/N$ for all u, v. In the case of a pseudo-random permutation, the joint probabilities can be computed directly (our strategy of adjustments of the conditionals described earlier is applicable here also and leads to the same joint probabilities as those obtained from the direct computations). Substituting all the relevant probability values, we get

$$\Pr(Z_1 = 0) \approx \Pr(Z_1 = 1) \approx \frac{1}{N} - \frac{1}{N(N-1)} \quad \text{and}$$
$$\Pr(Z_1 = v) \approx \frac{1}{N} + \frac{1}{N(N-1)(N-2)} \quad \text{for } 2 \le v \le 255,$$

which is almost a uniform distribution for $2 \le v \le 255$. This supports our claim that KSA causes the 'sine-curve-like' distribution of the first output byte. Fig. 6 shows the graph for this distribution.

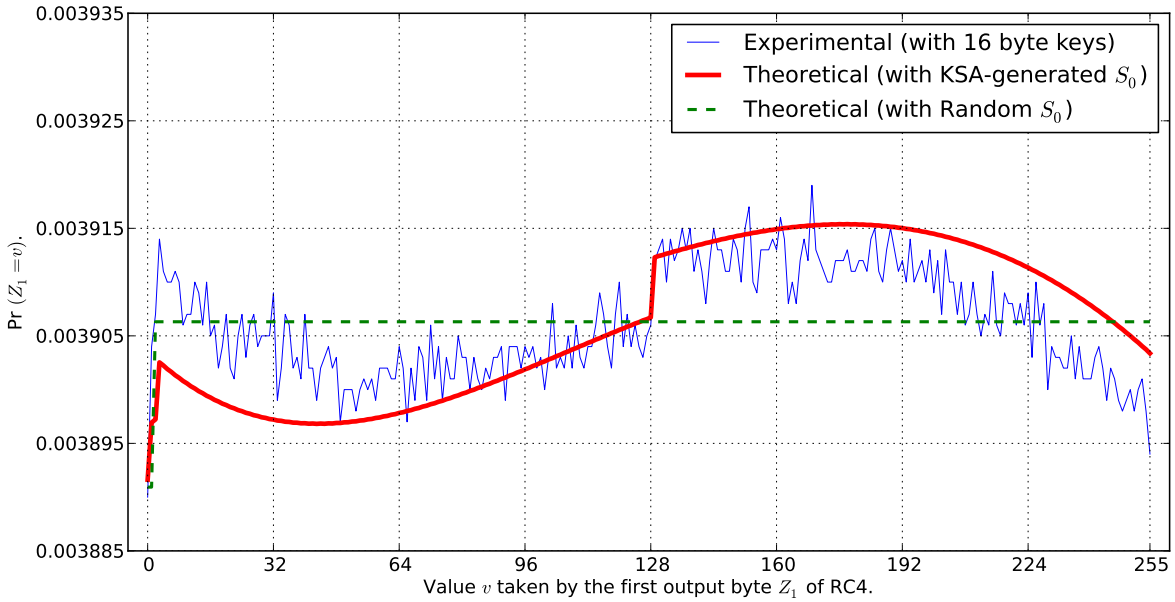

Fig. 6. The probability distribution of the first output byte Z_1.

Remark 3. Theorem 10 is the special case $v = 0$ of Theorem 11 and hence may seem redundant. However, we like to point out that the former has a simple and straightforward proof assuming S_0 to be random and the latter has a rigorous general proof without any assumption on S_0. The result of Theorem 10 signifies that this negative bias is not an artifact of non-random S_0 produced by RC4 KSA, rather it would be present, even if one starts PRGA with a uniformly random permutation.

3.3 Biases of keystream bytes 3 to 255 towards zero

In FSE 2001, Mantin and Shamir [16] proved the famous $2/N$ bias towards the value 0 for the second byte of RC4 keystream. In addition, they made the following claims.

MS-Claim 1: $\Pr(Z_r = 0) = \frac{1}{N}$ at PRGA rounds $3 \leq r \leq 255$.
MS-Claim 2: $\Pr(Z_r = 0 \mid j_r = 0) > \frac{1}{N}$ and $\Pr(Z_r = 0 \mid j_r \neq 0) < \frac{1}{N}$ for $3 \leq r \leq 255$. These two biases cancel each other to produce no bias in the event $(Z_r = 0)$ in rounds 3 to 255.

MS-Claim 2 was made to justify MS-Claim 1 in [16]. In this section, contrary to MS-Claim 1, we show (in Theorem 12) that $\Pr(Z_r = 0) > \frac{1}{N}$ for all rounds r from 3 to 255. The immediate implications are that we find 253 new distinguishers of RC4, and that the validity of MS-Claim 2 is questionable. In this context, we rigorously analyze the work of [16] to refute the aforementioned claims, and to study the (non)-randomness of j in PRGA.

We show that all the initial 253 bytes of RC4 keystream from round 3 to 255 are biased to zero. To prove the main result, we will require the following corollary of Theorem 8. This corollary follows from the fact that $S_r[j_r] = S_{r-1}[i_r] = S_{r-1}[r]$.

Corollary 1. *For $r \geq 3$, the probability $\Pr(S_{r-1}[r] = r)$ is approximately*

$$\Pr(S_1[r] = r) \left[1 - \frac{1}{N}\right]^{r-2} + \sum_{t=2}^{r-1} \sum_{k=0}^{r-t} \frac{\Pr(S_1[t] = r)}{k! \cdot N} \left[\frac{r-t-1}{N}\right]^k \left[1 - \frac{1}{N}\right]^{r-3-k}$$

Now, we can state our main theorem on the bias of RC4 initial bytes.

Theorem 12. *For $3 \leq r \leq 255$, the probability that the r-th RC4 keystream byte is equal to 0 is*

$$\Pr(Z_r = 0) \approx \frac{1}{N} + \frac{c_r}{N^2}.$$

where c_r is given by $\frac{N}{N-1}\left[N \cdot \Pr(S_{r-1}[r] = r) - 1\right]$ with $\Pr(S_{r-1}[r] = r)$ as in Corollary 1.

Proof. We prove the result by decomposing the event $(Z_r = 0)$ into two mutually exclusive and exhaustive cases, as follows.

$$\Pr(Z_r = 0) = \Pr\left(Z_r = 0 \wedge S_{r-1}[r] = r\right) + \Pr\left(Z_r = 0 \wedge S_{r-1}[r] \neq r\right) \tag{11}$$

Now we consider the events $(Z_r = 0 \wedge S_{r-1}[r] = r)$ and $(Z_r = 0 \wedge S_{r-1}[r] \neq r)$ individually to calculate their probabilities. In this direction, note that

$$Z_r = S_r[S_r[i_r] + S_r[j_r]] = S_r[S_r[r] + S_{r-1}[i_r]] = S_r[S_r[r] + S_{r-1}[i_r]] = S_r[S_r[r] + S_{r-1}[r]].$$

This expression for Z_r will be used in various effects throughout the paper.

Calculation of $\Pr(Z_r = 0 \wedge S_{r-1}[r] = r)$**:** In this case $S_{r-1}[r] = r$, and thus we have the probability

$$\begin{aligned}\Pr(Z_r = 0 \wedge S_{r-1}[r] = r) &= \Pr(S_r[S_r[r] + r] = 0 \wedge S_{r-1}[r] = r) \\ &= \sum_{x=0}^{N-1} \Pr(S_r[x + r] = 0 \wedge S_r[r] = x \wedge S_{r-1}[r] = r) \\ &= \sum_{x=0}^{N-1} \Pr(S_r[x + r] = 0 \wedge S_r[r] = x) \cdot \Pr(S_{r-1}[r] = r) \qquad (12)\end{aligned}$$

The last expression results from the assumption that the events $(S_r[x + r] = 0)$ and $(S_r[r] = x)$ are both independent from $(S_{r-1}[r] = r)$, as a state update has occurred in the process. Note that $S_{r-1}[r] = r$ is one of the values that gets swapped to produce the new state S_r (location $[r]$ denotes $[i_r]$ at this stage), and this is why we can claim the independence of $S_r[r]$ and $S_{r-1}[r]$. Otherwise, if a location $[s]$ is not same as $[i_r]$ or $[j_r]$, then $S_r[s]$ would be the same as $S_{r-1}[s]$, even after the state update.

Now, let us compute $\Pr(S_r[x+r] = 0 \wedge S_r[r] = x) = \Pr(S_r[x+r] = 0) \cdot \Pr(S_r[r] = x \mid S_r[x+r] = 0)$ independently. In this expression, if there exists any bias in the event $(S_r[x + r] = 0)$, then it must propagate from a similar bias in $(S_0[x + r] = 0)$, as was the case for $(S_{r-1}[r] = r)$ in Corollary 1. However, $\Pr(S_0[x + r] = 0) = \frac{1}{N}$ by Proposition 4, and thus we can safely assume $S_r[x + r]$ to be random as well. This provides us with $\Pr(S_r[x + r] = 0) = \frac{1}{N}$.

For $\Pr(S_r[r] = x \mid S_r[x + r] = 0)$, observe that when $x = 0$, the indices $[x + r]$ and $[r]$ in the state S_r point to the same location, and the events $(S_r[x + r] = S_r[r] = 0)$ and $(S_r[r] = x = 0)$ denote identical events. Thus in this case, $\Pr(S_r[r] = x \mid S_r[x + r] = 0) = 1$. In cases where $x \neq 0$, the indices $[x+r]$ and $[r]$ refer to two distinct locations in the permutation S_r, obviously containing different values. In this case,

$$\Pr(S_r[r] = x \mid S_r[x + r] = 0) = \Pr(S_r[r] = x \mid x \neq 0) = \frac{1}{N - 1}.$$

For justifying the randomness of $S_r[r]$ for $x \neq 0$, one may simply observe that the location $[r] = [i_r]$ is the one that got swapped to generate state S_r from the previous state, and thus the randomness assumption of $S_r[r]$ is based on the randomness assumption of j_r, which is validated for $r \geq 3$ later in Section 3.3.4.

According to the discussion above, we obtain

$$\Pr(S_r[x + r] = 0 \wedge S_r[r] = x) = \begin{cases} \frac{1}{N} \cdot 1 = \frac{1}{N} & \text{if } x = 0, \\ \frac{1}{N} \cdot \frac{1}{N-1} = \frac{1}{N(N-1)} & \text{if } x \neq 0. \end{cases} \qquad (13)$$

Substituting these probability values in Equation (12), we get

$$\begin{aligned}&\Pr(Z_r = 0 \wedge S_{r-1}[r] = r) \\ &= \Pr(S_{r-1}[r] = r) \left[\sum_{x=0}^{N-1} \Pr(S_r[x + r] = 0 \wedge S_r[r] = x)\right] \\ &= \Pr(S_{r-1}[r] = r) \cdot \left[\frac{1}{N} + \sum_{x=1}^{N-1} \frac{1}{N(N-1)}\right] \\ &= \Pr(S_{r-1}[r] = r) \cdot \left[\frac{1}{N} + (N-1) \cdot \frac{1}{N(N-1)}\right] = \Pr(S_{r-1}[r] = r) \cdot \frac{2}{N}. \qquad (14)\end{aligned}$$

Calculation of $\Pr(Z_r = 0 \wedge S_{r-1}[r] \neq r)$**:** Similar to the previous case, we can derive the probability as follows:

$$\begin{aligned}\Pr(Z_r = 0 \wedge S_{r-1}[r] \neq r) &= \sum_{y \neq r} \Pr(S_r[S_r[r] + y] = 0 \wedge S_{r-1}[r] = y) \\ &= \sum_{y \neq r} \sum_{x=0}^{N-1} \Pr(S_r[x + y] = 0 \wedge S_r[r] = x \wedge S_{r-1}[r] = y)\end{aligned}$$

An interesting situation occurs if $x = r - y$. In this case, on one hand, we obtain $S_r[x+y] = S_r[r] = 0$ for the first event, while on the other hand, we get $S_r[r] = x = r - y \neq 0$ for the second event (note that $y \neq r$). This poses a contradiction (event with probability of occurrence 0), and hence we get

$$\begin{aligned}&\Pr(Z_r = 0 \wedge S_{r-1}[r] \neq r) \\ &= \sum_{y \neq r} \sum_{x \neq r-y} \Pr(S_r[x + y] = 0 \wedge S_r[r] = x \wedge S_{r-1}[r] = y) \\ &= \sum_{y \neq r} \sum_{x \neq r-y} \Pr(S_r[x + y] = 0 \wedge S_r[r] = x) \cdot \Pr(S_{r-1}[r] = y), \end{aligned} \tag{15}$$

where the last expression results from the fact that the events $(S_r[x + y] = 0)$ and $(S_r[r] = x)$ are both independent from $(S_{r-1}[r] = y)$, as a state update has occurred in the process, and $S_{r-1}[r]$ got swapped during that update.

Similar to the derivation of Equation (13), we obtain

$$\Pr(S_r[x + y] = 0 \wedge S_r[r] = x) = \begin{cases} 0 & \text{if } x = 0, \\ \dfrac{1}{N(N-1)} & \text{if } x \neq 0. \end{cases} \tag{16}$$

The only difference occurs in the case $x = 0$. In this situation, simultaneous occurrence of the events $(S_r[x + y] = S_r[y] = 0)$ and $(S_r[r] = x = 0)$ pose a contradiction as the two locations $[y]$ and $[r]$ of S_r are distinct (note that $y \neq r$), and they can not hold the same value 0 as the state S_r is a permutation. In all other cases $(x \neq 0)$, the argument is identical to that in the previous derivation.

Substituting the values above in Equation (15), we get

$$\begin{aligned}&\Pr(Z_r = 0 \wedge S_{r-1}[r] \neq r) \\ &= \sum_{y \neq r} \Pr(S_{r-1}[r] = y) \left[\sum_{x \neq r-y} \Pr(S_r[x + y] = 0 \wedge S_r[r] = x) \right] \\ &= \sum_{y \neq r} \Pr(S_{r-1}[r] = y) \left[0 + \sum_{\substack{x \neq r-y \\ x \neq 0}} \frac{1}{N(N-1)} \right] \\ &= \sum_{y \neq r} \Pr(S_{r-1}[r] = y) \left[(N-2) \cdot \frac{1}{N(N-1)} \right] \\ &= \frac{N-2}{N(N-1)} \sum_{y \neq r} \Pr(S_{r-1}[r] = y) \\ &= \frac{N-2}{N(N-1)} \cdot (1 - \Pr(S_{r-1}[r] = r)) = \frac{N-2}{N(N-1)} \cdot (1 - \Pr(S_{r-1}[r] = r)) \end{aligned} \tag{17}$$

Calculation for $\Pr(Z_r = 0)$**:** Combining the probabilities from Equation (14) and Equation (17) in the final expression of Equation (11), we obtain the following.

$$\begin{aligned}\Pr(Z_r = 0) &= \Pr(S_{r-1}[r] = r) \cdot \frac{2}{N} + \frac{N-2}{N(N-1)} \cdot (1 - \Pr(S_{r-1}[r] = r)) \\ &= \frac{\Pr(S_{r-1}[r] = r)}{N-1} + \frac{N-2}{N(N-1)} = \frac{1}{N} + \frac{1}{N-1} \cdot \left(\Pr(S_{r-1}[r] = r) - \frac{1}{N}\right) \qquad (18)\end{aligned}$$

Thus, $\Pr(Z_r = 0) \approx \frac{1}{N} + \frac{c_r}{N^2}$ with $c_r = \frac{N}{N-1}\left[N \cdot \Pr(S_{r-1}[r] = r) - 1\right]$, as required. □

In Theorem 12, we have presented the bias in the probability $\Pr(Z_r = 0)$ in terms of the parameter c_r, which in turn is a function of r and $\Pr(S_{r-1}[r] = r)$. But we are more interested in observing the bias for specific rounds of RC4 PRGA, namely within the interval $3 \leq r \leq 255$. Thus, we are interested in obtaining numerical bounds on the bias for this specific interval. The next result is a corollary of Theorem 12 that provides exact numeric bounds on $\Pr(Z_r = 0)$ within the interval $3 \leq r \leq 255$, depending on the corresponding bounds of c_r within the same interval.

Corollary 2. *For* $3 \leq r \leq 255$*, the probability that the* r*-th RC4 keystream byte is equal to 0 is bounded as follows*

$$\frac{1}{N} + \frac{1.347168}{N^2} \;\geq\; \Pr(Z_r = 0) \;\geq\; \frac{1}{N} + \frac{0.242811}{N^2}.$$

Proof. We calculated all values of c_r (as in Theorem 12) for the range $3 \leq r \leq 255$, and checked that c_r is a decreasing function in r where $3 \leq r \leq 255$ (refer to Fig. 7). Therefore we obtain

$$\max_{3 \leq r \leq 255} c_r = c_3 = 1.347168 \quad \text{and} \quad \min_{3 \leq r \leq 255} c_r = c_{255} = 0.242811.$$

Hence the result on the bounds of $\Pr(Z_r = 0)$, depending on the bounds of c_r. □

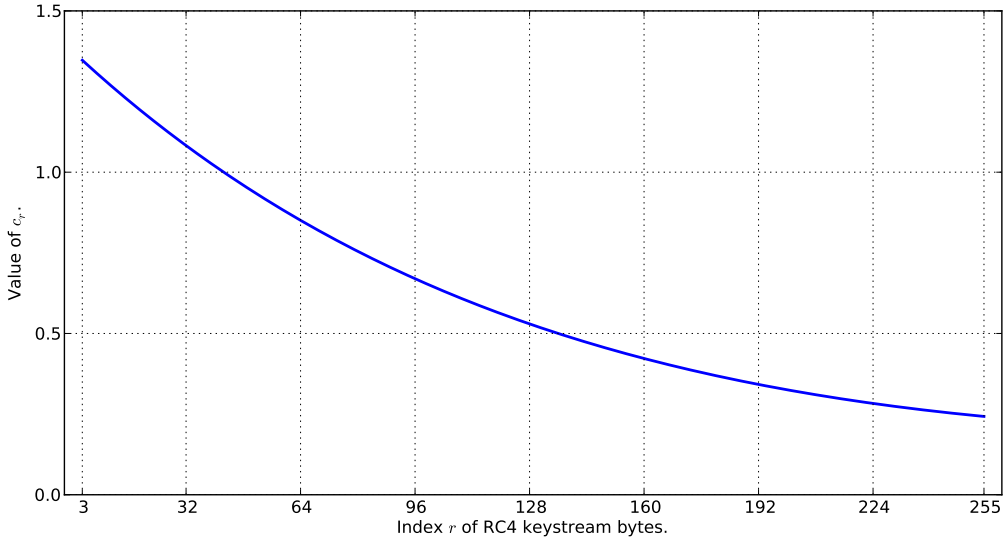

Fig. 7. Value of c_r versus r during RC4 PRGA ($3 \leq r \leq 255$).

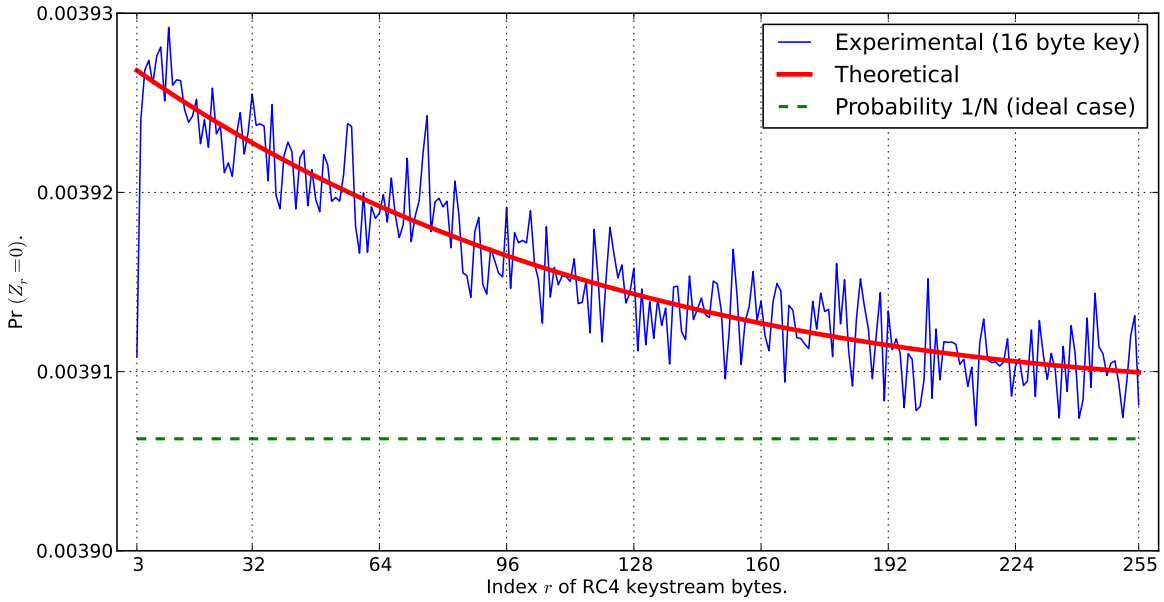

Fig. 8. $\Pr(Z_r = 0)$ versus r during RC4 PRGA ($3 \leq r \leq 255$).

Fig. 8 depicts a comparison between the theoretically derived vs. experimentally obtained values of $\Pr(Z_r = 0)$ versus r, where $3 \leq r \leq 255$. The experimentation has been carried out with 1 billion trials, each trial with a randomly generated 16 byte key.

Following the notation of Section 1.4, let P and Q denote the distributions corresponding to *random sequence* and *RC4 keystream* respectively, and E_r denote the event $(Z_r = 0)$ for $r = 3$ to 255. Writing $p = \frac{1}{N}$ and $q = \frac{c_r}{N}$, to distinguish RC4 keystream from random sequence based on the event $(Z_r = 0)$, one would need number of samples of the order of

$$\left(\frac{1}{N}\right)^{-1} \left(\frac{c_r}{N}\right)^{-2} \sim O(N^3).$$

Combined distinguisher: We can combine the effect of all these distinguishers by counting the number of zeros in the initial keystream of RC4, according to Theorem 13, as follows.

Theorem 13. *The expected number of 0's is approximately 0.990652 in RC4 rounds 3 to 255.*

Proof. Let X_r be a random variable taking values $X_r = 1$ if $Z_r = 0$, and $X_r = 0$ otherwise. Hence, the total number of 0's in rounds 3 to 255 is given by

$$C = \sum_{r=3}^{255} X_r.$$

We have $E(X_r) = \Pr(X_r = 1) = \Pr(Z_r = 0)$ from Theorem 12. By linearity of expectation,

$$E(C) = \sum_{r=3}^{255} E(X_r) = \sum_{r=3}^{255} \Pr(Z_r = 0).$$

Substituting the numeric values of the probabilities $\Pr(Z_r = 0)$ from Theorem 12, we get $E(C) \approx 0.990652$. Hence the result. □

For a random sequence of bytes, this expectation is $E(C) = 253/256 = 0.98828125$. Thus, the expectation for RC4 is approximately 0.24% higher than that for the random case. The inequality of this expectation in RC4 keystream compared to that in a random sequence of bytes may also be used to design a distinguisher.

3.3.1 Critical analysis of the event $(Z_r = 0)$ given j_r

Recall the expression for $\Pr(Z_r = 0)$ from Theorem 12:

$$\Pr(Z_r = 0) = \frac{1}{N} + \frac{1}{N-1} \cdot \left(\Pr(S_{r-1}[r] = r) - \frac{1}{N} \right) \approx \frac{1}{N} + \frac{c_r}{N^2}. \tag{19}$$

In the expression for $\Pr(S_{r-1}[r] = r)$, as in Corollary 1, we see that $\left(\frac{N-1}{N}\right)^{r-1} > \frac{1}{N}$ for all $3 \leq r \leq 255$. Thus, there is always a *positive* bias in $\Pr(S_{r-1}[r] = r)$, and in turn in $\Pr(Z_r = 0)$. Further, for any $r \geq 1$, we can write

$$\Pr(Z_r = 0) = \Pr(j_r = 0) \cdot \Pr(Z_r = 0 \mid j_r = 0) + \Pr(j_r \neq 0) \cdot \Pr(Z_r = 0 \mid j_r \neq 0). \tag{20}$$

One may note that MS-Claim 2 of Mantin and Shamir [16] essentially states that $\Pr(Z_r = 0 \mid j_r = 0) = \frac{1}{N} + a_r$ and $\Pr(Z_r = 0 \mid j_r \neq 0) = \frac{1}{N} - b_r$ for $3 \leq r \leq 255$, where both $a_r, b_r > 0$. Plugging these values in Equation (20), we have

$$\frac{1}{N} + \frac{c_r}{N^2} = \frac{1}{N}\left(\frac{1}{N} + a_r\right) + \left(1 - \frac{1}{N}\right)\left(\frac{1}{N} - b_r\right) \quad \text{for } 3 \leq r \leq 255.$$

Simplifying the above equation, we get $a_r = \frac{c_r}{N} + (N-1)b_r$. Thus, if MS-Claim 2 is correct, then we must have

$$\Pr(Z_r = 0 \mid j_r = 0) = \frac{1}{N} + \frac{c_r}{N} + (N-1)b_r = \frac{1 + c_r}{N} + (N-1)b_r,$$

where $0.98490994 \geq c_r \geq 0.36757467$ for $3 \leq r \leq 255$ (from Corollary 2). However, extensive experiments have confirmed that $\Pr(Z_r = 0 \mid j_r = 0) \approx \frac{1}{N}$, thereby refuting MS-Claim 2 of [16].

3.3.2 Guessing state information using the bias in Z_r

Mantin and Shamir [16] used the bias of the second byte of RC4 keystream to guess some information regarding $S_0[2]$, based on the following.

$$\Pr(S_0[2] = 0 \mid Z_2 = 0) = \frac{\Pr(S_0[2] = 0)}{\Pr(Z_2 = 0)} \cdot \Pr(Z_2 = 0 \mid S_0[2] = 0) \approx \frac{1/N}{2/N} \cdot 1 = \frac{1}{2}.$$

Note that in the above expression, no randomness assumption is required to obtain $\Pr(S_0[2] = 0) = 1/N$. This probability is exact and can be derived by substituting $u = 2, v = 0$ in Proposition 4. Hence, on every occasion we obtain $Z_2 = 0$ in the keystream, we can guess $S_0[2]$ with probability $1/2$, and this is significantly more than a random guess with probability $1/N$.

In this section, we use the biases in bytes 3 to 255 (observed in Theorem 12) to extract similar information about the state array S_{r-1} using the RC4 keystream byte Z_r. In particular, we try to explore the conditional probability $\Pr(S_{r-1}[r] = r \mid Z_r = 0)$ for $3 \leq r \leq 255$, as follows.

$$\Pr(S_{r-1}[r] = r \mid Z_r = 0) \;=\; \frac{\Pr(Z_r = 0 \wedge S_{r-1}[r] = r)}{\Pr(Z_r = 0)} \;\approx\; \frac{\Pr(S_{r-1}[r] = r) \cdot \frac{2}{N}}{\frac{1}{N} + \frac{c_r}{N^2}}$$

In the above expression, c_r is as in Theorem 12. One may write

$$\Pr(S_{r-1}[r] = r) = \frac{1}{N} + \frac{c_r}{N} - \frac{c_r}{N^2},$$

using Equation (18) from the proof of Theorem 12, and thereby obtain

$$\begin{aligned}\Pr(S_{r-1}[r] = r \mid Z_r = 0) &\approx \frac{\left(\frac{1}{N} + \frac{c_r}{N} - \frac{c_r}{N^2}\right) \cdot \frac{2}{N}}{\frac{1}{N} + \frac{c_r}{N^2}} \\ &= 2 \cdot \left(\frac{1}{N} + \frac{c_r}{N} - \frac{c_r}{N^2}\right) \cdot \left(1 + \frac{c_r}{N}\right)^{-1} \approx \frac{2}{N} + \frac{2c_r}{N}.\end{aligned}$$

From the expression for $\Pr(S_{r-1}[r] = r \mid Z_r = 0)$ derived above, one can guess $S_{r-1}[r]$ with probability more than twice of the probability of a random guess, every time we obtain $Z_r = 0$ in the RC4 keystream. In Fig. 9, we plot the theoretical probabilities

$$\Pr(S_{r-1}[r] = r \mid Z_r = 0) = 2 \cdot \left(\frac{1}{N} + \frac{c_r}{N} - \frac{c_r}{N^2}\right) \cdot \left(1 + \frac{c_r}{N}\right)^{-1}$$

against r for $3 \leq r \leq 255$, and the corresponding experimental values observed by running the RC4 algorithm 1 billion times with randomly selected 16 byte keys. It clearly shows that all the experimental values are also greater than $2/N$, as desired.

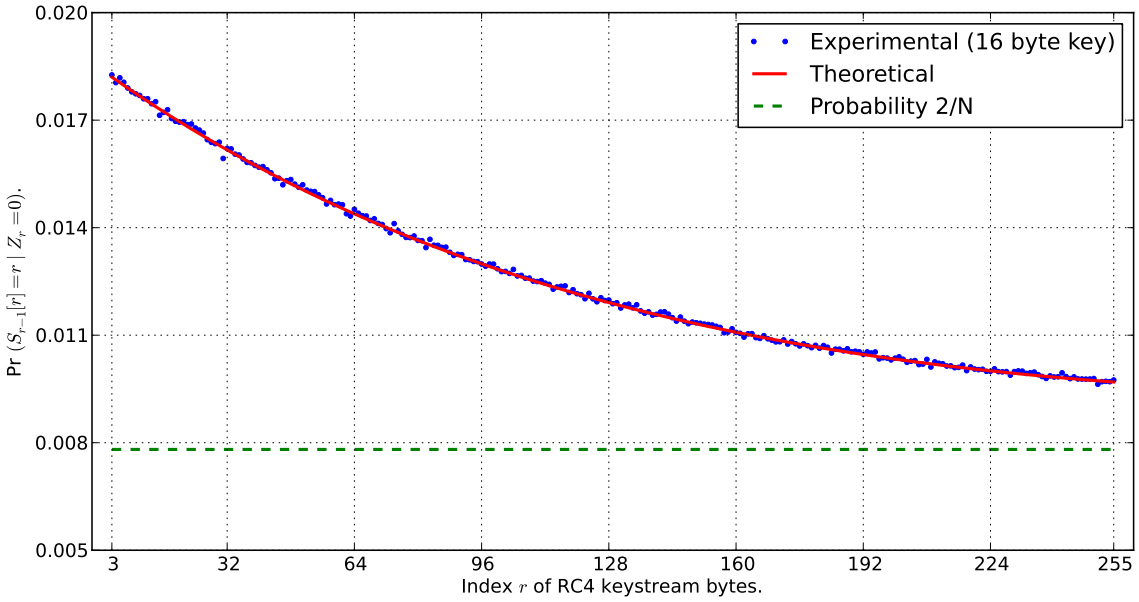

Fig. 9. $\Pr(S_{r-1}[r] = r \mid Z_r = 0)$ versus r during RC4 PRGA ($3 \leq r \leq 255$).

3.3.3 Attacking the RC4 broadcast scheme

Let us now revisit the famous attack of Mantin and Shamir [16] on broadcast RC4. As mentioned in their paper,

> "A classical problem in distributed computing is to allow N Byzantine generals to coordinate their actions when up to one third of them can be traitors. The problem is solved by a multi-round protocol in which each general broadcasts the same plaintext (which initially consists of either "Attack" or "Retreat") to all the other generals, where each copy is encrypted under a different key agreed in advance between any two generals."

In [16], the authors propose a practical attack against an RC4 implementation of the broadcast scheme, based on the bias observed in the second keystream byte. They prove that an enemy that collects $k = \Omega(N)$ number of ciphertexts corresponding to the same plaintext M, can easily deduce the second byte of M, by exploiting the bias in Z_2.

In a similar line of action, we may exploit the bias observed in bytes 3 to 255 of the RC4 keystream to mount a similar attack on RC4 broadcast scheme. Notice that we obtain a bias of the order of $1/N^2$ in each of the bytes Z_r where $3 \leq r \leq 255$. Thus, roughly speaking, if the attacker obtains about N^3 ciphertexts corresponding to the same plaintext M (from the broadcast scheme), then he can check the frequency of occurrence of bytes to deduce the r-th ($3 \leq r \leq 255$) byte of M.

The most important point to note is that this technique will work for each r where $3 \leq r \leq 255$, and hence will reveal *all the 253 initial bytes* (number 3 to 255 to be specific) of the plaintext M. We can formally state our result (analogous to [16, Theorem 3]) as follows.

Theorem 14. *Let M be a plaintext,and let $C_1, C_2, \ldots, C_k$ be the RC4 encryptions of M under k uniformly distributed keys. Then if $k = \Omega(N^3)$, the bytes 3 to 255 of M can be reliably extracted from $C_1, C_2, \ldots, C_k$.*

Proof. Recall from Theorem 12 that $\Pr(Z_r = 0) \approx \frac{1}{N} + \frac{c_r}{N^2}$ for all $3 \leq r \leq 255$ in the RC4 keystream. Thus, for each encryption key chosen during broadcast, the r-th plaintext byte $M[r]$ has probability $\frac{1}{N} + \frac{c_r}{N^2}$ to be XOR-ed with 0.

Due to the bias of Z_r towards zero, $\frac{1}{N} + \frac{c_r}{N^2}$ fraction of the r-th ciphertext bytes will have the same value as the r-th plaintext byte, with a higher probability. When $k = \Omega(N^3)$, the attacker can identify the most frequent character in $C_1[r], C_2[r], \ldots, C_k[r]$ as $M[r]$ with constant probability of success. □

The attack on broadcast RC4 is applicable to many modern Internet protocols (such as group emails encrypted under different keys, group-ware multi-user synchronization etc.). Note that Mantin and Shamir's attack [16] works at the byte level. It can recover only the second byte of the plaintext under some assumptions. On the other hand, our attack can recover additional 253 bytes (namely, bytes 3 to 255) of the plaintext.

3.3.4 Non-randomness of j in PRGA

During the PRGA round of RC4 algorithm, two indices are used; the first is i (deterministic) and the second is j (pseudo-random). Index i starts from 0 and increments by 1 (modulo N) at the beginning of each iteration, whereas j depends on the values of i and $S[i]$ simultaneously. The pseudo-randomness of the internal state S triggers the pseudo-randomness in j. In this section, we attempt to understand the pseudo-random behavior of j more clearly.

In RC4 PRGA, we know that for $r \geq 1$, $i_r = r \bmod N$ and $j_r = j_{r-1} + S_{r-1}[i_r]$, starting with $j_0 = 0$. Thus, we can write the values assumed by j at different rounds of PRGA as follows.

$$\begin{aligned}
j_1 &= j_0 + S_0[i_1] = 0 + S_0[1] = S_0[1], \\
j_2 &= j_1 + S_1[i_2] = S_0[1] + S_1[2], \\
j_3 &= j_2 + S_2[i_3] = S_0[1] + S_1[2] + S_2[3], \\
&\vdots \\
j_r &= j_{r-1} + S_{r-1}[i_r] = S_0[1] + S_1[2] + \cdots + S_{r-1}[r] = \sum_{x=1}^{r} S_{x-1}[x],
\end{aligned}$$

where $1 \leq r \leq N-1$, and all the additions are performed modulo N, as usual.

Non-randomness of j_1: In the first round of PRGA, $j_1 = S_0[1]$ follows a probability distribution which is determined by S_0, the internal state array after the completion of KSA. According to Proposition 4, we have

$$\Pr(j_1 = v) = \Pr(S_0[1] = v) = \begin{cases} \frac{1}{N} & \text{if } v = 0; \\ \frac{1}{N}\left(\frac{N-1}{N} + \frac{1}{N}\left(\frac{N-1}{N}\right)^{N-2}\right) & \text{if } v = 1; \\ \frac{1}{N}\left(\left(\frac{N-1}{N}\right)^{N-2} + \left(\frac{N-1}{N}\right)^{v}\right) & \text{if } v > 1. \end{cases}$$

This clearly tells us that j_1 is *not* random. This is also portrayed in Fig. 10.

Non-randomness of j_2: In the second round of PRGA however, we have $j_2 = S_0[1] + S_1[2]$, which demonstrates better randomness, as discussed next. Note that we have the following in terms of probability for j_2.

$$\Pr(j_2 = v) = \Pr(S_0[1] + S_1[2] = v) = \sum_{w=0}^{N-1} \Pr(S_0[1] = w \wedge S_1[2] = v - w) \tag{21}$$

In the above expression, $(v - w)$ is performed modulo N, like all arithmetic operations in RC4. The following cases may arise with respect to Equation (21).

Case I. Suppose that $j_1 = S_0[1] = w = 2$. Then, we will have $S_1[i_2] = S_1[2] = S_1[j_1] = S_0[i_1] = S_0[1] = 2$. In this case,

$$\Pr(j_2 = v) = \begin{cases} \Pr(S_0[1] = 2) & \text{if } v = 4, \\ 0 & \text{otherwise.} \end{cases}$$

Case II. Suppose that $j_1 = S_0[1] = w \neq 2$. Then $S_0[2]$ will not get swapped in the first round, and hence we will have $S_1[2] = S_0[2]$. In this case,

$$\Pr(S_0[1] = w \wedge S_1[2] = v - w) = \Pr(S_0[1] = w \wedge S_0[2] = v - w).$$

Let us substitute the results obtained from these cases to Equation (21) to obtain

$$\Pr(j_2 = v) = \begin{cases} \Pr(S_0[1] = 2) + \sum\limits_{\substack{w=0 \\ w \neq 2}}^{N-1} \Pr(S_0[1] = w \wedge S_0[2] = v - w), & \text{if } v = 4; \\ \sum\limits_{\substack{w=0 \\ w \neq 2}}^{N-1} \Pr(S_0[1] = w \wedge S_0[2] = v - w), & \text{if } v \neq 4. \end{cases} \tag{22}$$

Equation (22) completely specifies the exact probability distribution of j_2, where each of the probabilities $\Pr(S_0[x] = y)$ can be substituted by their exact values from Proposition 4 with the adjustment as in Section 3.2.3 for estimating the joint probabilities. However, the expression suffices to exhibit the non-randomness of j_2 in the RC4 PRGA, having a large bias for $v = 4$. We found that the theoretical values corresponding to the probability distribution of j_2 (as in Equation (22)) match almost exactly with the experimental data plotted in Fig. 10. For the sake of clarity, we do not show the theoretical curve in Fig. 10.

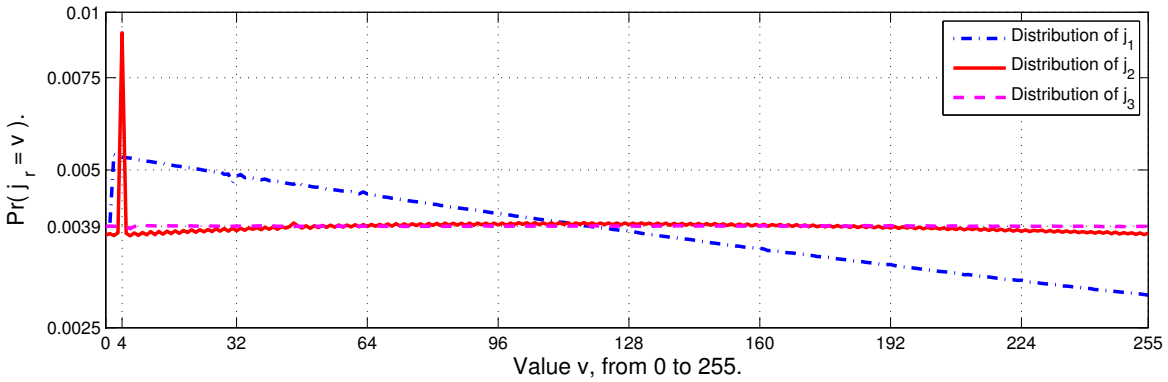

Fig. 10. Probability distribution of j_r for $1 \leq r \leq 3$.

Let us now evaluate $\Pr(j_2 = 4)$ independently:

$$\Pr(j_2 = 4) = \Pr(S_0[1] = 2) + \sum_{\substack{w=0 \\ w \neq 2}}^{N-1} \Pr(S_0[1] = w \wedge S_0[2] = 4 - w)$$

$$= \frac{1}{N}\left[\left(\frac{N-1}{N}\right)^{N-2} + \left(\frac{N-1}{N}\right)^{2}\right] + \sum_{\substack{w=0 \\ w \neq 2}}^{N-1} \Pr(S_0[1] = w \wedge S_0[2] = 4 - w)$$

Following Proposition 4 and the same strategy of estimating the joint probabilities as in Section 3.2.3, the summation term in the above expression evaluates approximately to $0.965268/N$ for $N = 256$. Thus, we get

$$\Pr(j_2 = 4) \approx \frac{1}{N}\left[\left(\frac{N-1}{N}\right)^{N-2} + \left(\frac{N-1}{N}\right)^{2}\right] + \frac{0.965268}{N} \approx \frac{7/3}{N}.$$

This verifies our experimental observation, as depicted in Fig. 10.

Guessing state information using the bias in j_2: It is also feasible to use this bias of j_2 to guess certain information about the RC4 state S_2. In particular, we shall focus on the event $(S_2[i_2] = 4 - Z_2)$ or $(S_2[2] = 4 - Z_2)$, and prove the following bias for this event.

Theorem 15. *After completion of the second round of RC4 PRGA, the state variable $S_2[2]$ equals the value $4 - Z_2$ with probability*

$$\Pr\left(S_2[2] = 4 - Z_2\right) \approx \frac{1}{N} + \frac{4/3}{N^2}.$$

Proof. First, note that we can write Z_2 in terms of the state variables as follows

$$Z_2 = S_2[S_2[i_2] + S_2[j_2]] = S_2[S_1[j_2] + S_1[i_2]] = S_2[S_1[j_2] + S_1[2]].$$

Thus, we can write the probability of the target event ($S_2[2] = 4 - Z_2$) as follows

$$\begin{aligned}\Pr(S_2[2] = 4 - Z_2) &= \Pr(S_2[i_2] = 4 - S_2[S_1[j_2] + S_1[2]]) \\ &= \Pr(S_1[j_2] = 4 - S_2[S_1[j_2] + S_1[2]]) \\ &= \Pr(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4)\end{aligned}$$

The idea is to exploit the bias in the event ($j_2 = 4$) to obtain the bias in the probability mentioned above. Thus, we decompose the target event into two mutually exclusive and exhaustive cases:

$$\begin{aligned}(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4) = &(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 = 4) \\ &\cup (S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 \neq 4)\end{aligned}$$

First event $(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 = 4)$**:** The probability for the first event can be calculated as follows.

$$\begin{aligned}&\Pr\left(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 = 4\right) \\ &= \Pr(S_1[4] + S_2[S_1[4] + S_1[2]] = 4 \wedge j_2 = 4) \\ &= \sum_{y=0}^{N-1} \Pr(S_1[4] + S_2[y] = 4 \wedge S_1[4] + S_1[2] = y \wedge j_2 = 4) \\ &= \sum_{y=0}^{N-1} \Pr(S_1[4] + S_2[y] = 4 \wedge S_1[4] + S_1[2] = y) \cdot \Pr(j_2 = 4) \\ &= \Pr(j_2 = 4) \sum_{y=0}^{N-1} \Pr(S_1[4] + S_2[y] = 4 \wedge S_1[4] + S_1[2] = y)\end{aligned}$$

In the last expression, the values taken from S_1 are independent of the value of j_2, and thus the events $(S_1[4] + S_2[y] = 4)$ and $(S_1[4] + S_1[2] = y)$ are both independent of the event $(j_2 = 4)$. Also note that if $y = 4$, we obtain

$$S_1[4] + S_2[y] = S_1[4] + S_2[4] = S_1[4] + S_2[j_2] = S_1[4] + S_1[i_2] = S_1[4] + S_1[2],$$

which results in the events $(S_1[4] + S_2[y] = 4)$ and $(S_1[4] + S_1[2] = y)$ being identical. In all other cases, we have $S_1[4] + S_2[y] \neq S_1[4] + S_1[2]$ and thus the values are chosen distinctly independent at random. Hence, we obtain

$$\Pr(S_1[4] + S_2[y] = 4 \wedge S_1[4] + S_1[2] = y) = \begin{cases} \frac{1}{N} & \text{if } y = 4; \\ \frac{1}{N(N-1)} & \text{if } y \neq 4. \end{cases}$$

The probabilities in the above expression are verified through experimentation by running the RC4 algorithm 1 billion times, choosing a 16 byte key uniformly at random in each run. The probability for the first event turns out to be

$$\begin{aligned}&\Pr\left(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 = 4\right) \\ &= \Pr(j_2 = 4) \cdot \left[\frac{1}{N} + \sum_{y \neq 4} \frac{1}{N(N-1)}\right] \\ &= \frac{7/3}{N} \cdot \left[\frac{1}{N} + (N-1) \cdot \frac{1}{N(N-1)}\right] = \frac{7/3}{N} \cdot \frac{2}{N}.\end{aligned}$$

Second event $(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 \neq 4)$**:** For the second event, the probability calculation can be performed in a similar fashion, as follows.

$$\begin{aligned}
&\Pr\left(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 \neq 4\right) \\
&= \sum_{x \neq 4} \Pr(S_1[x] + S_2[S_1[x] + S_1[2]] = 4 \wedge j_2 = x) \\
&= \sum_{x \neq 4} \sum_{y=0}^{N-1} \Pr(S_1[x] + S_2[y] = 4 \wedge S_1[x] + S_1[2] = y \wedge j_2 = x)
\end{aligned}$$

Note that the case $y = x$ poses an interesting situation. On one hand, we obtain

$$S_1[x] + S_2[y] = S_1[x] + S_2[x] = S_1[x] + S_2[j_2] = S_1[x] + S_1[i_2] = S_1[x] + S_1[2] = 4,$$

while on the other hand, we get $S_1[x] + S_1[2] = x \neq 4$. We rule out the case $y = x$ from the probability calculation due to this contradiction, and get

$$\begin{aligned}
&\Pr\left(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 \neq 4\right) \\
&= \sum_{x \neq 4} \sum_{y \neq x} \Pr(S_1[x] + S_2[y] = 4 \wedge S_1[x] + S_1[2] = y \wedge j_2 = x) \\
&= \sum_{x \neq 4} \sum_{y \neq x} \Pr(S_1[x] + S_2[y] = 4 \wedge S_1[x] + S_1[2] = y) \cdot \Pr(j_2 = x).
\end{aligned}$$

As before, in the last expression, the values taken from S_1 are independent of the value of j_2, and thus the events $(S_1[x] + S_2[y] = 4)$ and $(S_1[x] + S_1[2] = y)$ are both independent of the event $(j_2 = x)$.

Another interesting case occurs if $y = 4$ in the above calculation. In this case, one one hand, we have $S_1[x] + S_2[4] = 4$, while one the other hand we get $S_1[x] + S_1[2] = 4$. One may notice that $S_1[4]$ is a value that does not get swapped to obtain the state S_2. This is because the only two values to get swapped at this stage are from the locations $[i_2] = [2]$ and $[j_2] = [x] \neq [4]$. Thus, $S_2[4] = S_1[4]$ and we get $S_1[x] + S_1[4] = 4$ and $S_1[x] + S_1[2] = 4$, indicating $S_1[4] = S_1[2]$. As S_1 is a permutation, this case is not possible, and all other cases deal with two distinct locations of the permutation S_1. Therefore, we obtain

$$\Pr(S_1[x] + S_2[y] = 4 \wedge S_1[x] + S_1[2] = y) = \begin{cases} 0 & \text{if } y = 4; \\ \frac{1}{N(N-1)} & \text{otherwise.} \end{cases}$$

In turn, we obtain the probability of the second event as follows.

$$\begin{aligned}
&\Pr\left(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4 \wedge j_2 \neq 4\right) \\
&= \sum_{x \neq 4} \Pr(j_2 = x) \sum_{y \neq x} \Pr(S_1[x] + S_2[y] = 4 \wedge S_1[x] + S_1[2] = y) \\
&= \sum_{x \neq 4} \Pr(j_2 = x) \left[0 + \sum_{\substack{y \neq x \\ y \neq 4}} \frac{1}{N(N-1)} \right] = \sum_{x \neq 4} \Pr(j_2 = x) \left[(N-2) \cdot \frac{1}{N(N-1)} \right] \\
&= \frac{N-2}{N(N-1)} \sum_{x \neq 4} \Pr(j_2 = x) = \frac{N-2}{N(N-1)} \cdot (1 - \Pr(j_2 = 4)) = \frac{N-2}{N(N-1)} \cdot \left(1 - \frac{7/3}{N^2}\right).
\end{aligned}$$

Calculation for $\Pr(S_2[2] = 4 - Z_2)$: Combining the probabilities for the first and second events, we obtain the final probability as

$$\Pr(S_2[2] = 4 - Z_2) = \frac{7/3}{N^2} \cdot \frac{2}{N} + \frac{N-2}{N(N-1)} \cdot \left(1 - \frac{7/3}{N^2}\right) \approx \frac{1}{N} + \frac{4/3}{N^2}.$$

Hence the desired probability for the event $(S_2[2] = 4 - Z_2)$. □

Thus, one can guess the value of $S_2[i_2] = S_2[2]$ with probability greater than that of a random guess (probability $1/N$). For $N = 256$, the result matches with our experimental data generated from 1 billion runs of RC4 with randomly selected 16 byte keys.

Randomness of j_r for $r \geq 3$: Along the same line of analysis as in the case of j_2, it is possible to compute the explicit probability distributions of $j_r = \sum_{x=1}^{r} S_{x-1}[x]$ for $3 \leq r \leq 255$ as well. We do not present the expressions $\Pr(j_r = v)$ for $r \geq 3$ to avoid complication. However, it turns out that $j_r = \sum_{x=1}^{r} S_{x-1}[x]$ becomes closer to be random as r increase. The probability distributions of j_1, j_2 and j_3 are shown in Fig. 10, where the experiments have been run over 1 billion trials of RC4 PRGA, with randomly generated keys of size 16 bytes.

One may note that the randomness in j_2 is more than that of j_1 (apart from the case $v = 4$), and j_3 is almost uniformly random. This trend continues for the later rounds of PRGA as well. However, we do not plot the graphs for the probability distributions of j_r with $r \geq 4$, as these distributions are almost identical to that of j_3, i.e., almost uniformly random in behavior.

4 Long-term manifestation of short-term biases in RC4

The biases discussed so far are prevalent in the initial bytes of the RC4 output sequence, and are generally referred to as the short-term biases of the cipher. It is a common practice to discard a few hundred initial bytes of the output sequence to avoid these biases, and this motivates the search for long-term biases in RC4 that are present even after discarding an arbitrary number of initial bytes.

There has only been a handful of results in this direction till date. The first set of results was proposed by Fluhrer and McGrew [6] in 2000, and the biases depend upon the frequency of occurrence of certain digraphs in the RC4 keystream. Each of these are biases of magnitude $O(1/N^3)$ for corresponding base events with probability $1/N^2$. Later in 2005, Mantin [17] improved these to obtain the best long-term distinguisher of RC4 till date, also known as the $AB\mathcal{S}AB$ distinguisher. This bias is of magnitude $O(e^{(-4-8G)/N}/N^3)$ for a base event with probability $1/N^2$, and it depends on the repetition of digraphs in the keystream after a gap of G words. In 2008, Basu et al. [2] identified another conditional long-term bias, depending on the relationship between two consecutive bytes in the output sequence. This bias turned out to be weaker, as it is of magnitude $O(1/N^3)$ for a base event with probability $1/N$.

However, in each of these cases, the long-term biases were completely different from any short-term bias in RC4 that we know of, and depended on digraphs or relations between consecutive bytes. Our motivation to search for long-term biases start from a systematic investigation of long-term manifestations of the known short-term biases, if there is any at all. In this direction, it is natural to initiate the study with the most prominent short-term biases of RC4, namely, the biases of the first two keystream bytes towards zero. The negative bias in the first byte Z_1 towards 0 is approximately of the magnitude $1/N^2$, as observed by Mironov [20] in 2002, and proved in Section 3.2. The positive bias in the second byte Z_2 towards 0 is approximately $1/N$, as proved by Mantin and Shamir [16] in 2001. We first study their long-term propagation characteristics, as follows.

4.1 Direct long-term manifestation of short-term biases

The main motivation to look for long-term manifestations of short-term biases arises from the following structure for viewing the PRGA cycle of RC4:

$$\{1, 2, \ldots, 255\} \bigcup \{256\} \bigcup \{257, 258, \ldots, 511\} \bigcup \{512\} \bigcup \cdots,$$

where each long period is exactly 255 rounds, and the single rounds in between act as buffers to initiate the next PRGA cycle. It is only the initial period that lacks the buffer round in front. The proposed structure is illustrated in Fig. 11, where $\mathcal{A}$ and $\mathcal{B}$ denote the main cycles of PRGA.

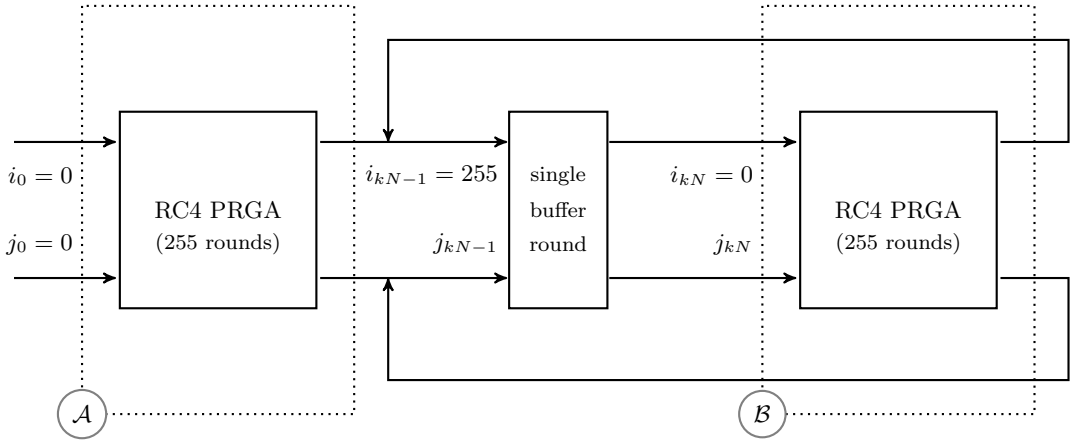

Fig. 11. The cycle structure considered for RC4 PRGA.

In this new structure of RC4 PRGA, we immediately observe the following.

- Each main cycle $\mathcal{A}$ or $\mathcal{B}$ of RC4 PRGA starts with $i_0 = 0$.
- First main cycle $\mathcal{A}$ of RC4 PRGA has $j_0 = 0$ to start with, and each main cycle $\mathcal{B}$ thereafter starts with initial pseudo-random index j_{kN}.
- All characteristics of the first main cycle will be repeated if $j_{kN} = 0$ in any other cycle of PRGA.

One may restate the third point to obtain the most important observation for our results:

When $j_{kN} = 0$, the main cycle $\mathcal{B}$ will be identical to the initial cycle $\mathcal{A}$ of RC4 PRGA.

Recall that the biases in the initial cycle $\mathcal{A}$ for events $[Z_1 = 0]$ (proved in Section 3.2) and $[Z_2 = 0]$ (proved in [16, Theorem 1]) does not depend on the *non-randomness of the initial permutation* S_0 due to KSA. Rather, they both depend only on the *non-randomness of byte-extraction* in PRGA, given the initial conditions $i_0 = 0$ and $j_0 = 0$. Thus, if $j_{kN} = 0$, these biases will be present in the events $[Z_{kN+1} = 0]$ and $[Z_{kN+2} = 0]$ of the main cycle $\mathcal{B}$ as well. This motivates us to investigate further for long-term biases in the bytes Z_{kN+1} and Z_{kN+2}.

Investigation of long-term bias in Z_{kN+1}: The negative bias in Z_1 directly propagates to all cycles of PRGA that start with $j_{kN} = 0$. Note that the event $[j_{kN} = 0]$ occurs with probability $1/N$ as j_{kN} is uniformly random. Theorem 10 directly implies (also evident from Theorem 11 for KSA-generated S_0) that

$$\Pr[Z_{kN+1} = 0 \mid j_{kN} = 0] = 1/N - 1/N^2. \tag{23}$$

When $j_{kN} \neq 0$, we find that $[Z_{kN+1} = 0]$ happens only due to a random association, and thus

$$\begin{aligned}\Pr[Z_{kN+1} = 0] &\approx \Pr[Z_{kN+1} = 0 \mid j_{kN} = 0] \cdot \Pr[j_{kN} = 0] + \Pr[Z_{kN+1} = 0 \mid j_{kN} \neq 0] \cdot \Pr[j_{kN} \neq 0] \\ &\approx (1/N - 1/N^2) \cdot 1/N + 1/N \cdot (1 - 1/N) = 1/N - 1/N^3.\end{aligned}$$

The probability computed above gives us a long-term bias for the event $[Z_{kN+1} = 0]$. However, it is a weak bias, as the magnitude of the bias is only $O(1/N^3)$ for a base event with probability $1/N$.

Investigation of long-term bias in Z_{kN+2}: Similar to the previous case, the positive bias in Z_2 directly propagates to later rounds if $j_{kN} = 0$. Mantin and Shamir's observation, as in [16, Theorem 1] implies that

$$\Pr[Z_{kN+2} = 0 \mid j_{kN} = 0] \approx 2/N - 1/N^2. \tag{24}$$

However, considering the case $j_{kN} \neq 0$, we observe that Z_{kN+2} does not take the value 0 by uniform random association. In particular, when $S_{kN}[2] = 0$, the value of Z_{kN+2} is 0 with probability 0. This case is illustrated in Fig. 12.

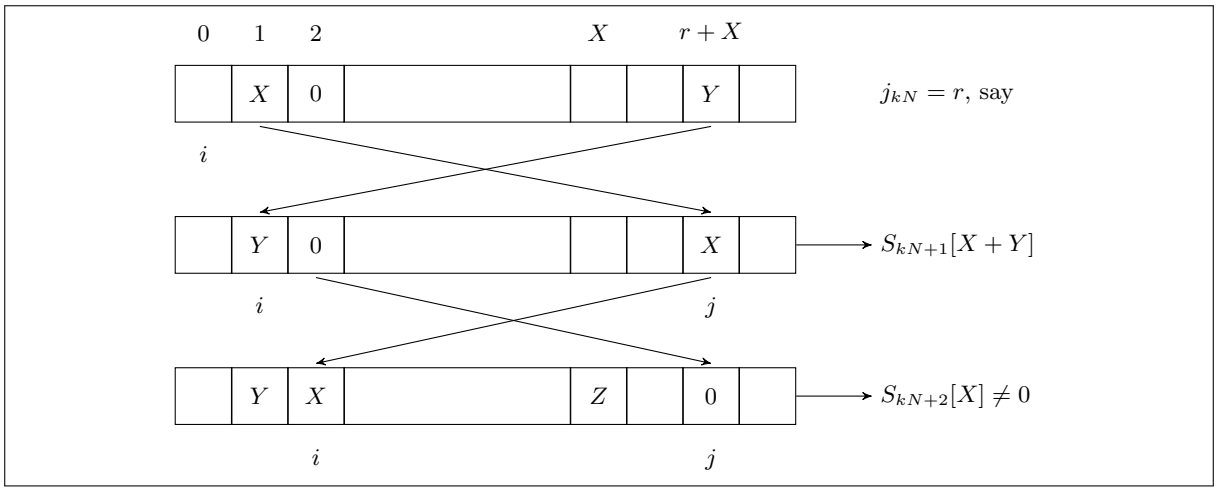

Fig. 12. The first two rounds of RC4 main cycle when $S_{kN}[2] = 0$ and $S_{kN}[1] \neq 2$.

Suppose that $j_{kN} = r \neq 0$ at some cycle of RC4 PRGA. Then for $S_{kN}[1] = X$ and $S_{kN}[2] = 0$, the swap sequence is as shown in Fig. 12, and we finally obtain 0 in the $(r+X)$-th location of S_{kN+2}. But the output byte at this stage is emitted from location X of the same permutation S_{kN+2}, and thus it can not be 0 (note that the locations X and $r+X$ are different as $r \neq 0$). This case arises for approximately $1/N$ fraction of the keys (as $S_{kN}[2]$ is uniformly random), and for the other $(1-1/N)$ fraction of keys, we may obtain $Z_{kN+2} = 0$ when $j_{kN} \neq 0$. Considering both cases,

$$\begin{aligned}\Pr[Z_{kN+2} = 0 \wedge j_{kN} \neq 0] &= \sum_{r \neq 0} \sum_{S_{kN}[2] \neq 0} \Pr[Z_{kN+2} = 0 \wedge j_{kN} = r] \\ &= (1 - 1/N) \sum_{r \neq 0} \Pr[Z_{kN+2} = 0 \mid j_{kN} = r] \cdot \Pr[j_{kN} = r] \\ &\approx (1 - 1/N) \cdot (N - 1) \cdot (1/N) \cdot (1/N) = 1/N - 2/N^2 + 1/N^3. \end{aligned} \tag{25}$$

Adding the contributions from the two cases $j_{kN} = 0$ and $j_{kN} \neq 0$, we obtain

$$\Pr[Z_{kN+2} = 0] \approx (2/N - 1/N^2) \cdot 1/N + (1/N - 2/N^2 + 1/N^3) = 1/N.$$

It is interesting to note that the second byte exhibits a strong bias in the first PRGA cycle of RC4 (as in [16]), but behaves almost uniformly random thereafter for all subsequent cycles.

Investigation of long-term bias in Z_{kN}: Note that byte Z_0 does not occur in the initial round of PRGA, as the process starts from the byte Z_1 itself. However, considering the cycle structure of PRGA, the byte Z_{kN} refers to the output of the buffer round (henceforth referred to as round number kN). Lemma 2 establishes a conditional bias for Z_{kN}, which is similar to that of Z_{kN+2}.

Lemma 2. *For any integer $k \geq 1$, assume that the permutation S_{kN} is randomly chosen from the set of all possible permutations of $\{0, 1, \ldots, N-1\}$. Then*

$$\Pr[Z_{kN} = 0 \mid j_{kN} = 0] \approx 2/N - 1/N^2.$$

Proof. We have $i_{kN} = 0$ in each PRGA cycle. When $j_{kN} = 0$ (this happens with probability $1/N$), no swap takes place and the output is $Z_{kN} = S_{kN}[2 \cdot S_{kN}[0]]$.

Two cases may arise from here. If $S_{kN}[0] = 0$, then $Z_{kN} = S_{kN}[0] = 0$ for sure. Otherwise if $S_{kN}[0] \neq 0$, the output Z_{kN} takes the value 0 only due to random association. Combining the cases,

$$\Pr[Z_{kN} = 0 \mid j_{kN} = 0] \approx 1/N \cdot 1 + (1 - 1/N) \cdot 1/N = 2/N - 1/N^2.$$

Hence the desired result. □

Similar to the computation of $\Pr[Z_{kN+2} = 0 \wedge j_{kN} \neq 0]$, one may also compute

$$\Pr[Z_{kN} = 0 \wedge j_{kN} \neq 0] \approx 1/N - 2/N^2 + 1/N^3 \quad \text{and} \quad \Pr[Z_{kN} = 0 \mid j_{kN} \neq 0] \approx 1/N - 1/N^2 \quad (26)$$

Adding the contributions from the two mutually exclusive cases $j_{kN} = 0$ and $j_{kN} \neq 0$, we obtain

$$\Pr[Z_{kN} = 0] \approx (2/N - 1/N^2) \cdot 1/N + (1/N - 2/N^2 + 1/N^3) = 1/N. \quad (27)$$

Thus, alike Z_{kN+2}, the buffer-round output Z_{kN} does not exhibit any significant bias towards zero.

4.2 Conditional long-term manifestation of short-term biases

From the previous section, we find that in the long run, Z_{kN} and Z_{kN+2} do not show any bias towards zero and Z_{kN+1} has a slightly negative bias towards zero. However, from Equation (24) and Lemma 2 we see that both Z_{kN} and Z_{kN+2} have a significant bias towards 0, whenever $j_{kN} = 0$, and so does Z_{kN+1}. This motivates us to relate pairs of bytes together by eliminating the condition on the hidden state variable j_{kN}.

We first focus on the consecutive pair of bytes (Z_{kN}, Z_{kN+1}) and (Z_{kN+1}, Z_{kN+2}) to find a conditional relation. This leads to the discovery of the following long-term biases in the RC4 keystream.

$$\Pr[Z_{kN+1} = 0 \mid Z_{kN} = 0] \approx 1/N + 1/N^2, \qquad \Pr[Z_{kN+2} = 0 \mid Z_{kN+1} = 0] \approx 1/N + 2/N^2.$$

The above biases can also be derived as special cases of the digraph $(0, 0)$ bias observed in [6].

Long-term bias involving non-consecutive bytes: Note that the above biases (and all others listed in [6]) are based on relations between consecutive bytes, and they do not consider *bytes with a gap in between*. We investigate the relation between Z_{kN} and Z_{kN+2} with a single-byte gap, and obtain a new long-term bias. In this section, by combining Equation (24) and Lemma 2, we show that the event $(Z_{kN+2} = 0 \mid Z_{kN} = 0)$ is positively biased. This is a hitherto-undiscovered long-term bias in RC4 that originates mainly from the long-term manifestation of Mantin and Shamir's second byte bias [16]. *To the best of our knowledge, this is the first long-term bias of RC4 that involves non-consecutive bytes of the output sequence.*

New long-term bias in RC4: The following technical result presents our new conditional bias between the non-consecutive output bytes Z_{kN} and Z_{kN+2}, as discussed earlier.

Theorem 16. *For any integer $k \geq 1$, assume that the permutation S_{kN} is randomly chosen from the set of all possible permutations of $\{0, \ldots, N-1\}$. Then*

$$\Pr[Z_{kN+2} = 0 \mid Z_{kN} = 0] \approx 1/N + 1/N^2.$$

Proof. Let us first compute the joint probability $\Pr[Z_{kN+2} = 0 \wedge Z_{kN} = 0]$, which is equal to

$$\Pr[Z_{kN+2} = 0 \wedge Z_{kN} = 0 \wedge j_{kN} = 0] + \Pr[Z_{kN+2} = 0 \wedge Z_{kN} = 0 \wedge j_{kN} \neq 0].$$

Given $j_{kN} = 0$, the random variables Z_{kN+2} and Z_{kN} can be considered independent. Using equation (24) and Lemma 2, we get the following for this part.

$$\begin{aligned}
&\Pr\left[Z_{kN+2} = 0 \wedge Z_{kN} = 0 \wedge j_{kN} = 0\right] \\
&= \Pr[Z_{kN+2} = 0 \mid j_{kN} = 0] \cdot \Pr[Z_{kN} = 0 \mid j_{kN} = 0] \cdot \Pr[j_{kN} = 0] \\
&\approx (2/N - 1/N^2) \cdot (2/N - 1/N^2) \cdot (1/N) \approx 4/N^3 - 4/N^4.
\end{aligned}$$

From the second part, using equation (25) and equation (26), one has

$$\begin{aligned}
&\Pr\left[Z_{kN+2} = 0 \wedge Z_{kN} = 0 \wedge j_{kN} \neq 0\right] \\
&= \Pr[Z_{kN+2} = 0 \mid j_{kN} \neq 0] \cdot \Pr[Z_{kN} = 0 \mid j_{kN} \neq 0] \cdot \Pr[j_{kN} \neq 0] \\
&\approx (1/N - 1/N^2)^2 \cdot (1 - 1/N) \approx 1/N^2 - 3/N^3 + 3/N^4.
\end{aligned}$$

Adding the two expressions, we have $\Pr[Z_{kN+2} = 0 \wedge Z_{kN} = 0]$ as

$$\begin{aligned}
&\Pr\left[Z_{kN+2} = 0 \wedge Z_{kN} = 0 \wedge j_{kN} = 0\right] + \Pr[Z_{kN+2} = 0 \wedge Z_{kN} = 0 \wedge j_{kN} \neq 0] \\
&\approx (4/N^3 - 4/N^4) + (1/N^2 - 3/N^3 + 3/N^4) \approx 1/N^2 + 1/N^3.
\end{aligned}$$

Hence $\Pr[Z_{kN+2} = 0 \mid Z_{kN} = 0] = \Pr[Z_{kN+2} = 0 \wedge Z_{kN} = 0] / \Pr[Z_{kN} = 0] \approx 1/N + 1/N^2$. □

The RC4 triangle of conditional biases: Fig. 13 depicts a trio of long-term biases between the bytes, as discussed above. An arrow directed from event A to event B denotes the conditional event $[B \mid A]$, and the value on the arrow denotes the positive bias of the corresponding conditional event with respect to the probability $1/N$ of random association. Note that the grey arrows in the figure represent the biases involving consecutive bytes, and the red arrow denotes the new bias involving non-consecutive bytes of the output, as proved in the previous section.

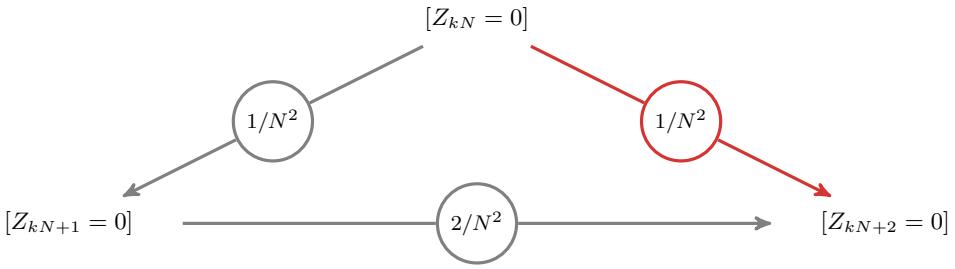

Fig. 13. The RC4 Triangle: Significant long-term biases in the RC4 keystream.

This new bias is of the order of $1/N^2$ for the base event with random probability of occurrence $1/N$. This indicates that we have obtained a long-term bias which is more significant in comparison to the best existing long-term biases of RC4 presented in [6, 17], which are around $O(1/N^3)$ for a base event with probability $1/N^2$. To identify the new bias, one shall require approximately $O(N^3)$ (i.e., around 2^{24} for $N = 256$) conditional samples, which is equivalent to approximately $O(N^4)$ bytes of output, as we consider only periodic intervals of length N to extract each sample.

The new long-term bias presented in this paper is the *first result* to observe a conditional bias between two non-consecutive bytes (Z_{kN}, Z_{kN+2}). The gap between the related bytes in this case is one, and we could not find any other significant long-term bias with this gap. An interesting direction for experimentation and analysis would be to look for similar long-term biases with larger gaps between the related bytes in the keystream. In the next section, we conclude the paper by summarizing our contributions and by proposing a few potential directions for future research.

5 Conclusion

In this paper, we have explored several classes of non-random events in RC4 - from key correlations to keystream-based distinguishers, and from short term biases to long-term non-randomness.

Key-dependent non-randomness: In practice, RC4 uses a small secret key of length l that is typically much less than the permutation size N. Hence each secret key byte is repeated at least $\lfloor N/l \rfloor$ times in the KSA. This is the source of several key-correlations and biases in the keystream. However, there were no biases reported in the literature that depends on the length l of the secret key. In this paper, we demonstrate the *first keylength-dependent biases* in the lineage of RC4 cryptanalysis.

Short-term non-randomness: The permutation after the RC4 KSA is non-random. This is the source of many biases in the initial keystream bytes, including the recent observations by Sepehrdad et al. [29], the sine-curve like probability distribution of the first byte observed by Mironov [20], and the second-byte bias observed by Mantin and Shamir [16]. In this paper, we prove all significant empirical biases observed in [29] and also provide the *first justification* for the sine-curve distribution of the first byte. We also extend the observation of [16] to all initial bytes 3 to 255 in the RC4 keystream, and hence *generalize the broadcast attack to recover all initial bytes.*

Long-term non-randomness: It is generally believed that the initial biases disappear if one discards the first few hundred bytes from the output sequence of RC4. However, we propose the idea that the short-term initial biases may have significant long-term manifestations. Our claim is supported by the discovery of a *new long-term bias* in the RC4 keystream that originates from a long-term periodic property of the second-byte bias. This discovery also generalizes the digraph patterns observed by Fluhrer and McGrew [6] by introducing *conditional relations between non-consecutive bytes.*

Future direction: In the search for non-random events in RC4, or other stream ciphers in general, our results open up the following interesting directions of research.

- What are the implications of using a secret key with length relatively small compared to the internal secret state of the cipher? How is the keylength related to the biases?
- Is there a general pattern in the non-random events generated from the initial non-random state produced by the KSA? Can we find more short-term biases in this direction?
- How does one generalize the concept of digraph biases to related bytes with arbitrary gaps in between? Are there more long-term biases of this kind in the RC4 output sequence?

References

1. M. Akgün, P. Kavak, and H. Demirci, "New Results on the Key Scheduling Algorithm of RC4," in *INDOCRYPT '08*, vol. 5365 of *LNCS*, pp. 40–52, 2008.
2. R. Basu, S. Ganguly, S. Maitra, and G. Paul, "A Complete Characterization of the Evolution of RC4 Pseudo Random Generation Algorithm," *Journal of Mathematical Cryptology (de Gruyter)*, vol. 2, no. 3, pp. 257–289, 2008.
3. R. Basu, S. Maitra, G. Paul, and T. Talukdar, "On Some Sequences of the Secret Pseudo-random Index j in RC4 Key Scheduling," in *AAECC '09*, vol. 5527 of *LNCS*, pp. 137–148, 2009.
4. E. Biham and Y. Carmeli, "Efficient Reconstruction of RC4 Keys from Internal States," in *FSE '08*, vol. 5086 of *LNCS*, pp. 270–288, 2008.
5. R. E. Blahut, *Principles and Practice of Information Theory*. Addisson-Wesley, 1983.
6. S. R. Fluhrer and D. A. McGrew, "Statistical Analysis of the Alleged RC4 Keystream Generator," in *FSE '00*, vol. 1978 of *LNCS*, pp. 19–30, 2000.
7. S. R. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," in *SAC '01*, vol. 2259 of *LNCS*, pp. 1–24, 2001.
8. J. D. Golic, "Linear Statistical Weakness of Alleged RC4 Keystream Generator," in *EUROCRYPT '97*, vol. 1233 of *LNCS*, pp. 226–238, 1997.
9. R. J. Jenkins, "ISAAC and RC4," 1996. Published on the Internet at `http://burtleburtle.net/bob/rand/isaac.html`.
10. S. Khazaei and W. Meier, "On Reconstruction of RC4 Keys from Internal States," in *MMICS '08*, vol. 5393 of *LNCS*, pp. 179–189, 2008.
11. A. Klein, "Attacks on the RC4 stream cipher," *Des. Codes Cryptography*, vol. 48, no. 3, pp. 269–286, 2008.
12. L. R. Knudsen, W. Meier, B. Preneel, V. Rijmen, and S. Verdoolaege, "Analysis Methods for (Alleged) RC4," in *ASIACRYPT '98*, vol. 1514 of *LNCS*, pp. 327–341, 1998.
13. S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Stats.*, vol. 22, pp. 49–86, 1951.
14. S. Maitra, G. Paul, and S. Sen Gupta, "Attack on Broadcast RC4 Revisited," in *FSE '11*, vol. 6733 of *LNCS*, pp. 199–217, 2011.
15. I. Mantin, "Analysis of the stream cipher RC4," Master's thesis, The Weizmann Institute of Science, Israel, 2001. Available at `http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html`.
16. I. Mantin and A. Shamir, "A Practical Attack on Broadcast RC4," in *FSE '01*, vol. 2355 of *LNCS*, pp. 152–164, 2002.
17. I. Mantin, "Predicting and Distinguishing Attacks on RC4 Keystream Generator," in *EUROCRYPT '05*, vol. 3494 of *LNCS*, pp. 491–506, 2005.
18. I. Mantin, "A Practical Attack on the Fixed RC4 in the WEP Mode," in *ASIACRYPT '05*, vol. 3788 of *LNCS*, pp. 395–411, 2005.
19. A. Maximov and D. Khovratovich, "New State Recovery Attack on RC4," in *CRYPTO '08*, vol. 5157 of *LNCS*, pp. 297–316, 2008.
20. I. Mironov, "(Not So) Random Shuffles of RC4," in *CRYPTO '02*, vol. 2442 of *LNCS*, pp. 304–319, 2002.
21. S. Mister and S. E. Tavares, "Cryptanalysis of RC4-like Ciphers," in *SAC'98*, vol. 1999 of *LNCS*, pages 131–143, 1998.
22. S. Paul and B. Preneel, "Analysis of Non-fortuitous Predictive States of the RC4 Keystream Generator," in *INDOCRYPT '03*, vol. 2904 of *LNCS*, pp. 52–67, 2003.
23. S. Paul and B. Preneel, "A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher," in *FSE '04*, vol. 3017 of *LNCS*, pp. 245–259, 2004.
24. G. Paul and S. Maitra, "Permutation After RC4 Key Scheduling Reveals the Secret Key," in *SAC '07*, vol. 4876 of *LNCS*, pp. 360–377, 2007.
25. G. Paul, S. Rathi, and S. Maitra, "On non-negligible bias of the first output byte of RC4 towards the first three bytes of the secret key," *Des. Codes Cryptography*, vol. 49, no. 1-3, pp. 123–134, 2008.
26. G. Paul and S. Maitra, "On biases of permutation and keystream bytes of RC4 towards the secret key," *Cryptography and Communications*, vol. 1, no. 2, pp. 225–268, 2009.
27. A. Roos, "A class of weak keys in the RC4 stream cipher." Two posts in sci.crypt, message-id `43u1eh$1j3@hermes.is.co.za` and `44ebge$llf@hermes.is.co.za`, 1995. Available at `http://www.impic.org/papers/WeakKeys-report.pdf`.
28. S. Sen Gupta, S. Maitra, G. Paul and S. Sarkar. "Proof of Empirical RC4 Biases and New Key Correlations," in *SAC '11*, vol. 7118 of *LNCS*, pp. 151–168, 2011.
29. P. Sepehrdad, S. Vaudenay, and M. Vuagnoux, "Discovery and Exploitation of New Biases in RC4," in *SAC '10*, vol. 6544 of *LNCS*, pp. 74–91, 2011.
30. P. Sepehrdad, S. Vaudenay, and M. Vuagnoux, "Statistical Attack on RC4 - Distinguishing WPA," in *EUROCRYPT '11*, vol. 6632 of *LNCS*, pp. 343–363, 2011.
31. Y. Shiraishi, T. Ohigashi, and M. Morii, "An Improved Internal-state Reconstruction Method of a Stream Cipher RC4," in *Communication, Network, and Information Security*, Track 440–088, December 10-12, New York, USA, 2003.
32. V. Tomasevic, S. Bojanic, and O. Nieto-Taladriz, "Finding an internal state of RC4 stream cipher," *Information Sciences*, vol. 177, pp. 1715–1727, 2007.
33. E. Tews, R.-P. Weinmann, and A. Pyshkin, "Breaking 104 Bit WEP in Less Than 60 Seconds," in *WISA '07*, vol. 4867 of *LNCS*, pp. 188–202, 2007.
34. E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *WISEC '09*, pp. 79–86, ACM, 2009.
35. S. Vaudenay and M. Vuagnoux, "Passive-Only Key Recovery Attacks on RC4," in *SAC '07*, vol. 4876 of *LNCS*, pp. 344–359, 2007.
36. D. A. Wagner, "My RC4 weak keys," 1995. (`http://www.cs.berkeley.edu/~daw/my-posts/my-rc4-weak-keys`)