Attack on Broadcast RC4 Revisited

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

[1] Applied Statistics Unit, Indian Statistical Institute,
Kolkata 700 108, India.
{subho, souravsg_r}@isical.ac.in
[2] Department of Computer Science and Engineering,
Jadavpur University, Kolkata 700 032, India.
goutam.paul@ieee.org

Abstract. In this paper, contrary to the claim of Mantin and Shamir (FSE 2001), we prove that there exist biases in the initial bytes (3 to 255) of the RC4 keystream towards zero. These biases immediately provide distinguishers for RC4. Additionally, the attack on broadcast RC4 to recover the second byte of the plaintext can be extended to recover the bytes 3 to 255 of the plaintext given $\Omega(N^3)$ many ciphertexts. Further, we also study the non-randomness of index j for the first two rounds of PRGA, and identify a strong bias of j_2 towards 4. This in turn provides us with certain state information from the second keystream byte.

Keywords: Bias, Broadcast RC4, Cryptanalysis, Distinguishing Attack, Keystream, RC4, Stream Cipher.

1 Introduction

RC4, designed by Ron Rivest for RSA Data Security in 1987, is the most popular commercial stream cipher algorithm. There are two components of the RC4 algorithm, namely, the Key Scheduling Algorithm (KSA) and the Pseudo-Random Generation Algorithm (PRGA), that are presented in Algorithm 1 and Algorithm 2 respectively. Given a secret key k of size l bytes (typically, $5 \leq l \leq 16$), an array K of size N bytes (typically, $N = 256$) is created to hold the key such that $K[y] = k[y \bmod l]$ for any $y \in [0, N-1]$. The KSA uses this secret key to scramble a permutation S of $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$, initialized as the identity permutation. After that, the PRGA generates keystream bytes to be bitwise XOR-ed with the plaintext. The indices i (deterministic) and j (pseudo-random) are used to point to the locations of S. All additions in the KSA and PRGA routines of RC4 algorithm are performed modulo N.

Since the advent of RC4 in 1987, it has faced rigorous analysis over the years due to its simple structure. Extensive research has been conducted to identify weaknesses of RC4 in terms of the KSA as well as the PRGA. There are several important results in cryptanalysis of RC4 where the initial bytes are not of concern. The most prominent recent works in this direction are the distinguisher proposed by Mantin [4] (based on the occurrence of strings of the

Input: Secret Key K.
Output: S-Box S generated by K.

for $i = 0, \ldots, N-1$ **do**
  $S[i] = i$;
end

Initialize counter: $j = 0$;

for $i = 0, \ldots, N-1$ **do**
  $j = j + S[i] + K[i]$;
  Swap $S[i] \leftrightarrow S[j]$;
end

Algorithm 1: KSA

Input: S-Box S, output of KSA.
Output: Random stream Z generated from S.

Initialize the counters: $i = j = 0$;

while *TRUE* **do**
  $i = i + 1$;
  $j = j + S[i]$;
  Swap $S[i] \leftrightarrow S[j]$;
  Output $Z = S[S[i] + S[j]]$;
end

Algorithm 2: PRGA

pattern $ABTAB$ with A, B bytes and T a string of bytes of small length), and the state recovery attack presented by Maximov and Khovratovich [6].

However, the major portion of the literature in RC4 cryptanalysis involves results related to initial keystream bytes of PRGA [2, 5] (also see the references therein). To get rid of these problems, one may throw away some initial bytes of RC4 PRGA as suggested in [3, 7]. But it may not be easy to modify the actual implementations immediately by throwing away some initial keystream bytes, since RC4 is already in use in many commercial applications. Thus the cryptanalytic results related to the initial bytes are still of importance. Moreover, these results are always of theoretical significance in terms of studying one of the most popular stream ciphers. The trend continues, including the most recent biases in this direction [8] that relates the initial keystream bytes, state variables and secret key of RC4. Recently, another paper [9] accepted at Eurocrypt 2011 exploited the known biases of RC4 (mostly involving the initial bytes) to provide distinguishers against WEP and WPA. Using related idea, this paper also proposes the best key recovery attack against WPA till date.

Notation. Let S_r, i_r, j_r, z_r denote the state, index i, index j, and the keystream byte respectively, after r (≥ 1) rounds of PRGA have been performed. Let S_0 denote the state just before the PRGA starts, i.e., right after the KSA ends. Further, let $p_{r,x}$ denote the probability $\Pr(S_r[x] = x)$, after r rounds of PRGA, where $r \geq 1$ and $0 \leq x \leq N-1$.

Motivation and Contribution. In FSE 2001, Mantin and Shamir [5] published the best known distinguishing attack on RC4 based on the bias of the second byte towards zero. This result states that if the initial permutation is randomly chosen from the set of all $(N!)$ permutations of $\mathbb{Z}_N$, then $\Pr(z_2 = 0) \approx \frac{2}{N}$ in RC4 keystream, whereas this should be $\frac{1}{N}$ in case of a random stream of bytes.

In [5, Section 3.2], after the description of the bias in the event $(z_2 = 0)$, the following statement has been made:

> "One could expect to see a similar (but weaker) bias towards 0 at all the other outputs z_t with $t = 0 \bmod n$, since in $1/N^2$ of these cases $S_t[2] = 0$ and $j = 0$, which would give rise to the same situation. However, extensive experiments have shown that this weaker bias at later rounds does not exist. By carefully analyzing this situation one can show that for any $j \neq 0$, the output is zero with a slight negative bias, and the total contribution of these negative biases exactly cancels the positive bias derived from $j = 0$. The only time we don't have this cancellation effect is at the beginning of the execution, when j always starts as 0 rather than as a uniformly distributed random value."

The main two claims implied by the above statement are as follows.

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, when combined, 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 [5]. In the current work, contrary to MS-Claim 1, we show (Theorem 1) 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. This motivates us to analyze the work of [5] to refute the aforementioned claims, and to study the (non)-randomness of j in PRGA. It is quite surprising that this issue has never been identified over the last decade.

The bias in the second byte was used in [5] to mount a distinguisher. We use our newly discovered biases to construct a class of 253 new distinguishers corresponding to the initial 253 keystream bytes z_r for $r \in \{3, 4, \ldots, 255\}$.

In addition, we study the non-randomness of index j rigorously to find a strong bias of j_2 towards 4. We can use this bias to guess the internal state variable $S_2[2]$ from the value of keystream byte z_2. Very recently, the results published in [8] claimed an exhaustive search for biases in all possible linear combinations of the state variables and the RC4 keystream bytes. However, our result concerning the bias of j_2 towards 4 is not covered in [8].

The literature of RC4 cryptanalysis, developed over more than two decades, is quite rich. In context of this paper, we have only referred to the publications which have direct relevance with our work. The reader may look into the references therein for a more detailed overview.

During the proposition and proof of our results in this paper, we shall require the following well known result in RC4 cryptanalysis from the existing literature. This appears in [3, Theorem 6.3.1], and we can restate the result as follows.

Proposition 1 ([3]). *At the end of KSA, for* $0 \leq u \leq N-1$, $0 \leq v \leq N-1$,

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

Remark 1. As Proposition 1 reveals, the underlying assumption of Mantin and Shamir [5] regarding the randomness of the initial permutation is violated in practice. This non-randomness in the permutation for the initial state of PRGA gives rise to the biases that we report in this paper.

2 Bytes 3 to 255 of PRGA are Biased to Zero

In this section we show that all the initial 253 bytes of RC4 keystream from round 3 to 255 are biased to zero. To prove the main theorem, we require the following technical result.

Lemma 1. *For $r \geq 3$, the probability that $S_{r-1}[r] = r$ is*

$$p_{r-1,r} \approx p_{0,r} \cdot \left[\left(\frac{N-1}{N} \right)^{r-1} - \frac{1}{N} \right] + \frac{1}{N}.$$

Proof. The event $S_{r-1}[r] = r$ may occur in the following two ways.

1. $S_0[r] = r$, and index r is not touched by any i or j during first $(r-1)$ PRGA rounds: The first event occurs with probability $p_{0,r}$. For the second one, note that index r is not touched by $i = 1, \ldots, r-1$ values, and the probability that none of j touches it either is approximately $(\frac{N-1}{N})^{r-1}$. Thus the contribution of this case is approximately $p_{0,r} \cdot (\frac{N-1}{N})^{r-1}$.
2. $S_0[r] \neq r$, and still $S_{r-1}[r]$ equals r by random association: The probability of the first event is $(1 - p_{0,r})$ and given this event, the second one is likely to occur only due to random association, thus with probability $\approx \frac{1}{N}$. Hence, the contribution of this case is approximately $(1 - p_{0,r}) \cdot \frac{1}{N}$.

Adding the two contributions calculated above, we get the result. □

Remark 2. RC4 PRGA starts with $j_0 = 0$. For $r = 1$, we have $j_1 = j_0 + S_0[1] = S_0[1]$ which, due to Proposition 1, is not uniformly distributed. For $r = 2$, we have $j_2 = j_1 + S_1[2] = S_0[1] + S_1[2]$, whose probability distribution is more close to the uniform random distribution than that in case of j_1. In round 3, another pseudo-random byte $S_2[3]$ would be added to form j_3. From round 3 onwards, j can safely be assumed to be uniform over $\mathbb{Z}_N$. Experimental observations also confirm this. A detailed discussion on the randomness of j is presented in Section 4. In Item 1 of the proof of Lemma 1, the product

$$\Pr(j_1 \neq r) \cdot \Pr(j_2 \neq r) \cdots \Pr(j_{r-1} \neq r) = \Pr(j_1 \neq r) \cdot \Pr(j_2 \neq r) \cdot \left(\frac{N-1}{N} \right)^{r-3}$$

is approximated as $(\frac{N-1}{N})^{r-1}$, but one may always try the exact forms for the probabilities $\Pr(j_1 \neq r)$ and $\Pr(j_2 \neq r)$ to obtain further accuracy.

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

Theorem 1. *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 $\left[\left(\frac{N-1}{N}\right)^r + \left(\frac{N-1}{N}\right)^{N-r-1} - \left(\frac{N-1}{N}\right)^{N-1}\right] \cdot \left[\left(\frac{N-1}{N}\right)^{r-2} - \frac{1}{N-1}\right]$.

Proof. We prove the result by decomposing the event $(z_r = 0)$ into two mutually exclusive and exhaustive cases[1], as follows.

$$\Pr(z_r = 0) = \Pr\left(z_r = 0 \ \&\ S_{r-1}[r] = r\right) + \Pr\left(z_r = 0 \ \&\ S_{r-1}[r] \neq r\right) \quad (1)$$

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

$$\begin{aligned} z_r &= S_r[S_r[i_r] + S_r[j_r]] = S_r[S_r[r] + S_{r-1}[i_r]] \\ &= S_r[S_r[r] + S_{r-1}[i_r]] = S_r[S_r[r] + S_{r-1}[r]]. \end{aligned}$$

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

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

$$\begin{aligned} &\Pr\left(z_r = 0 \ \&\ S_{r-1}[r] = r\right) \\ &= \Pr(S_r[S_r[r] + r] = 0 \ \&\ S_{r-1}[r] = r) \\ &= \sum_{x=0}^{N-1} \Pr\left(S_r[x + r] = 0 \ \&\ S_r[r] = x \ \&\ S_{r-1}[r] = r\right) \\ &= \sum_{x=0}^{N-1} \Pr\left(S_r[x + r] = 0 \ \&\ S_r[r] = x\right) \cdot \Pr\left(S_{r-1}[r] = r\right) \end{aligned} \quad (2)$$

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 $\Pr(S_r[x + r] = 0 \ \&\ S_r[r] = x) = \Pr(S_r[x + r] = 0) \cdot \Pr(S_r[r] = x \mid S_r[x + r] = 0)$ 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 Lemma 1. However,

[1] In the pre-proceedings version, we had considered the same cases, and had obtained the same expressions for $\Pr(z_r = 0)$ and c_r. However, the proof for Theorem 1 used Jenkin's bias [1] (Glimpse) in an intermediate step as a crude approximation. In this version, we present a rigorous analysis which does not require to use Jenkin's bias.

$\Pr(S_0[x+r] = 0) = \frac{1}{N}$ by Proposition 1, 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 4.

According to the discussion above, we obtain

$$\Pr\left(S_r[x+r] = 0 \;\&\; S_r[r] = x\right) = \begin{cases} \frac{1}{N} \cdot 1 = \frac{1}{N} & \text{if } x = 0, \\ \frac{1}{N} \cdot \frac{1}{N-1} = \frac{1}{N(N-1)} & \text{if } x \neq 0. \end{cases} \tag{3}$$

Substituting these probability values in Equation (2), we get

$$\begin{aligned} &\Pr\left(z_r = 0 \;\&\; S_{r-1}[r] = r\right) \\ &= \Pr\left(S_{r-1}[r] = r\right) \left[\sum_{x=0}^{N-1} \Pr\left(S_r[x+r] = 0 \;\&\; S_r[r] = x\right)\right] \\ &= p_{r-1,r} \cdot \left[\frac{1}{N} + \sum_{x=1}^{N-1} \frac{1}{N(N-1)}\right] \\ &= p_{r-1,r} \cdot \left[\frac{1}{N} + (N-1) \cdot \frac{1}{N(N-1)}\right] = p_{r-1,r} \cdot \frac{2}{N}. \end{aligned} \tag{4}$$

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

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

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 can write

$$\begin{aligned}
&\Pr\left(z_r = 0 \;\&\; S_{r-1}[r] \neq r\right) \\
&= \sum_{y \neq r} \sum_{x \neq r-y} \Pr\left(S_r[x+y] = 0 \;\&\; S_r[r] = x \;\&\; S_{r-1}[r] = y\right) \\
&= \sum_{y \neq r} \sum_{x \neq r-y} \Pr\left(S_r[x+y] = 0 \;\&\; S_r[r] = x\right) \cdot \Pr\left(S_{r-1}[r] = y\right), \qquad (5)
\end{aligned}$$

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 (3), we obtain

$$\Pr\left(S_r[x+y] = 0 \;\&\; S_r[r] = x\right) = \begin{cases} 0 & \text{if } x = 0, \\ \frac{1}{N(N-1)} & \text{if } x \neq 0. \end{cases} \qquad (6)$$

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 (5), we get

$$\begin{aligned}
&\Pr\left(z_r = 0 \;\&\; S_{r-1}[r] \neq r\right) \\
&= \sum_{y \neq r} \Pr\left(S_{r-1}[r] = y\right) \left[\sum_{x \neq r-y} \Pr\left(S_r[x+y] = 0 \;\&\; S_r[r] = x\right) \right] \\
&= \sum_{y \neq r} \Pr\left(S_{r-1}[r] = y\right) \left[0 + \sum_{\substack{x \neq r-y \\ x \neq 0}} \frac{1}{N(N-1)} \right] \\
&= \sum_{y \neq r} \Pr\left(S_{r-1}[r] = y\right) \left[(N-2) \cdot \frac{1}{N(N-1)} \right] \\
&= \frac{N-2}{N(N-1)} \sum_{y \neq r} \Pr\left(S_{r-1}[r] = y\right) \\
&= \frac{N-2}{N(N-1)} \cdot \left(1 - \Pr\left(S_{r-1}[r] = r\right)\right) = \frac{N-2}{N(N-1)} \cdot \left(1 - p_{r-1,r}\right) \qquad (7)
\end{aligned}$$

Calculation for $\Pr(z_r = 0)$**:** Combining the probabilities from Equation (4) and Equation (7) in the final expression of Equation (1), we obtain the following.

$$\begin{aligned}
\Pr(z_r = 0) &= p_{r-1,r} \cdot \frac{2}{N} + \frac{N-2}{N(N-1)} \cdot \left(1 - p_{r-1,r}\right) \\
&= \frac{p_{r-1,r}}{N-1} + \frac{N-2}{N(N-1)} \;=\; \frac{1}{N} + \frac{1}{N-1} \cdot \left(p_{r-1,r} - \frac{1}{N}\right) \qquad (8)
\end{aligned}$$

Now, substituting the value of $p_{r-1,r}$ from Lemma 1 in Equation (8), we obtain

$$\Pr(z_r = 0) \approx \frac{1}{N} + \frac{1}{N-1} \cdot p_{0,r} \cdot \left[\left(\frac{N-1}{N} \right)^{r-1} - \frac{1}{N} \right]. \tag{9}$$

Further, we can use Proposition 1 to get the value of $p_{0,r}$ as

$$p_{0,r} = \Pr(S_0[r] = r) = \frac{1}{N} \left[\left(\frac{N-1}{N} \right)^{r} + \left(1 - \left(\frac{N-1}{N} \right)^{r} \right) \left(\frac{N-1}{N} \right)^{N-r-1} \right].$$

Substituting this expression for $p_{0,r}$ in Equation (9), we obtain the desired result $\Pr(z_r = 0) \approx \frac{1}{N} + \frac{c_r}{N^2}$ with the claimed value of c_r. □

In Theorem 1, 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. 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 1 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 1. *For $3 \leq r \leq 255$, the probability that the r-th RC4 keystream byte is equal to 0 is bounded as follows*

$$\frac{1}{N} + \frac{0.98490994}{N^2} \;\geq\; \Pr(z_r = 0) \;\geq\; \frac{1}{N} + \frac{0.36757467}{N^2}.$$

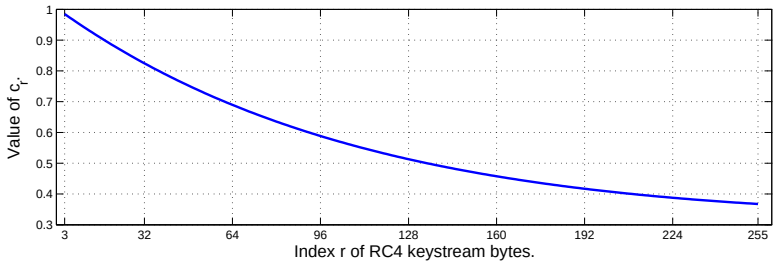

Fig. 1. Value of c_r versus r during RC4 PRGA ($3 \leq r \leq 255$).

Proof. We calculated all values of c_r (as in Theorem 1) for the range $3 \leq r \leq 255$, and checked that c_r is a decreasing function in r where $3 \leq r \leq 255$ (one may refer to the plot in Fig. 1 in this regard). Therefore we obtain

$$\max_{3 \leq r \leq 255} c_r = c_3 = 0.98490994 \quad \text{and} \quad \min_{3 \leq r \leq 255} c_r = c_{255} = 0.36757467.$$

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

Fig. 2 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.

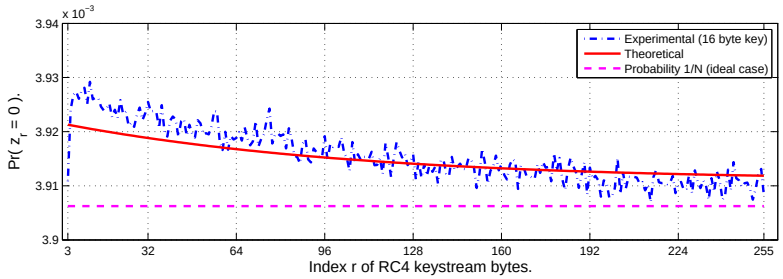

Fig. 2. $\Pr(z_r = 0)$ versus r during RC4 PRGA ($3 \leq r \leq 255$).

One may observe in Fig. 2 that the theoretical curve does not exactly coincide with the mean line of the experimental plot. This is algebraically expressed by the approximation in Theorem 1. The approximation arises due to the ideal randomness assumptions in the proof of Lemma 1, which do not hold in practice.

2.1 A Class of New Distinguishers

Theorem 1 immediately gives a class of distinguishers. In [5, Theorem 2], it is proved that if an event e happens with probabilities p and $p(1+\epsilon)$ in distributions X and Y respectively, then for p and ϵ with small magnitude, $O\left(p^{-1}\epsilon^{-2}\right)$ samples suffice to distinguish X from Y with a constant probability of success.

In our setting, let X and Y denote the distributions corresponding to *random stream* and *RC4 keystream* respectively, and e_r denote the event $(z_r = 0)$ for $r = 3$ to 255. From the formulation as in Equation (10), we can write $p = \frac{1}{N}$ and $\epsilon = \frac{c_r}{N}$. Thus, to distinguish RC4 keystream from random stream, based on the event $(z_r = 0)$, one would need number of samples of the order of

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

We can combine the effect of all these distinguishers by counting the number of zeros in the initial keystream of RC4, according to Theorem 2, as follows.

Theorem 2. *The expected number of 0's in RC4 keystream rounds 3 to 255 is approximately 0.9904610515.*

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 1. 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 1, we get $E(C) \approx 0.9904610515$. Hence the result. □

For a random stream of bytes, this expectation is $E(C) = \frac{253}{256} = 0.98828125$. Thus, the expectation for RC4 is approximately 0.22% higher than that for the random case. The inequality of this expectation in RC4 keystream compared to that in a random stream of bytes may also be used to design a distinguisher.

2.2 A Critical Analysis of the Event $(z_r = 0)$ Given $j_r =$ or $\neq 0$

Recall the expression for $\Pr(z_r = 0)$ from Theorem 1:

$$\Pr(z_r = 0) = \frac{1}{N} + \frac{1}{N-1} \cdot \left(p_{r-1,r} - \frac{1}{N} \right) \approx \frac{1}{N} + \frac{c_r}{N^2}. \tag{10}$$

In the expression for $p_{r-1,r}$, as in Lemma 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 $p_{r-1,r}$, and in turn in $\Pr(z_r = 0)$. Further, for any $r \geq 1$, we can write

$$\begin{aligned} \Pr(z_r = 0) = \Pr(j_r = 0) \cdot \Pr(z_r = 0 \mid j_r = 0) \\ + \Pr(j_r \neq 0) \cdot \Pr(z_r = 0 \mid j_r \neq 0). \end{aligned} \tag{11}$$

One may note that MS-Claim 2 of Mantin and Shamir [5] 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 (11), 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 1). However, extensive experiments have confirmed that $\Pr(z_r = 0 \mid j_r = 0) \approx \frac{1}{N}$, thereby refuting MS-Claim 2 of Mantin and Shamir.

2.3 Guessing State Information using the Bias in z_r

Mantin and Shamir [5] 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 $\Pr(S_0[2] = 0) = \frac{1}{N}$. This probability is exact and can be derived by substituting $u = 2, v = 0$ in Proposition 1. Hence, on every occasion we obtain $z_2 = 0$ in the keystream, we can guess $S_0[2]$ with probability $\frac{1}{2}$, and this is significantly more than a random guess with probability $\frac{1}{N}$.

In this section, we use the biases in bytes 3 to 255 (observed in Theorem 1) 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.

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

In the above expression, c_r is as in Theorem 1. One may write

$$p_{r-1,r} = \frac{1}{N} + \frac{c_r}{N} - \frac{c_r}{N^2},$$

using Equation (8) from the proof of Theorem 1, 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. 3, 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 $\frac{2}{N}$, as desired. The crisscross nature of the curves in Fig. 3 originates from a similar behavior observed in the curves of Fig. 2.

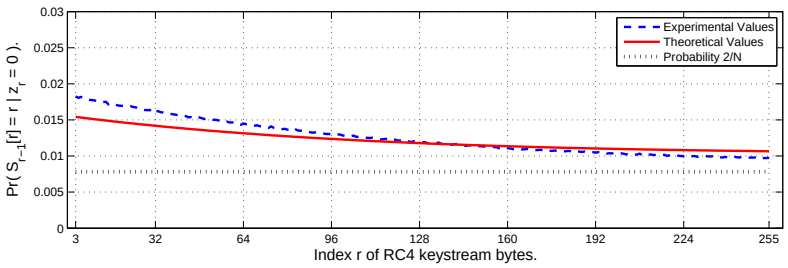

Fig. 3. $\Pr(S_{r-1}[r] = r \mid z_r = 0)$ versus r during RC4 PRGA ($3 \leq r \leq 255$).

3 Attacking the RC4 Broadcast Scheme

Let us now revisit the famous attack of Mantin and Shamir [5] 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 [5], 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 $\frac{1}{N^2}$ 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 [5, Theorem 3]) as follows.

Theorem 3. *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 1 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 [5] 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.

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.

4.1 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. Ac-

cording to Proposition 1, 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. 4.

4.2 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.

$$\begin{aligned} \Pr(j_2 = v) &= \Pr(S_0[1] + S_1[2] = v) \\ &= \sum_{w=0}^{N-1} \Pr(S_0[1] = w) \cdot \Pr((S_1[2] = v - w) \mid (S_0[1] = w)) \end{aligned} \quad (12)$$

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 (12).

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,

$$\Pr((S_1[2] = v - 2) \mid (S_0[1] = 2)) = \begin{cases} 1 & \text{if } v = 4, \\ 0 & \text{otherwise.} \end{cases}$$

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,

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

Let us substitute the results obtained from these cases to Equation (12) to obtain

$$\Pr(j_2 = v) = \begin{cases} \Pr(S_0[1] = 2) + \sum\limits_{\substack{w=0 \\ w \neq 2}}^{N-1} \Pr(S_0[1] = w) \Pr(S_0[2] = v - w), & \text{if } v = 4; \\ \sum\limits_{\substack{w=0 \\ w \neq 2}}^{N-1} \Pr(S_0[1] = w) \Pr(S_0[2] = v - w), & \text{if } v \neq 4. \end{cases} \quad (13)$$

Equation (13) 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 1. 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 (13)) match almost exactly with the experimental data plotted in Fig. 4. For the sake of clarity, we do not show the theoretical curve in Fig. 4.

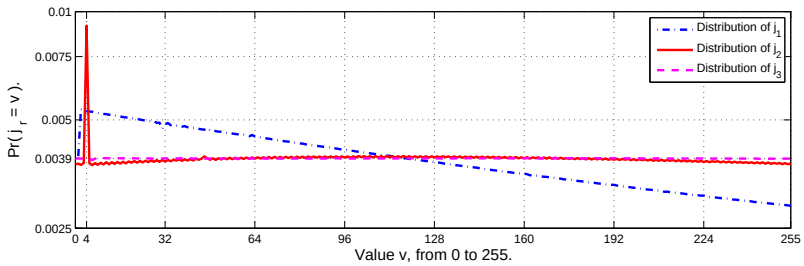

Fig. 4. Probability distribution of j_r for $1 \leq r \leq 3$.

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

$$\begin{aligned}
&\Pr\,(j_2 = 4) \\
&= \Pr(S_0[1] = 2) + \sum_{\substack{w=0 \\ w \neq 2}}^{N-1} \Pr(S_0[1] = w) \cdot \Pr(S_0[2] = 4 - w) \\
&= \frac{1}{N}\left[\left(\frac{N-1}{N}\right)^{N-2} + \left(\frac{N-1}{N}\right)^{2}\right] + \sum_{\substack{w=0 \\ w \neq 2}}^{N-1} \Pr(S_0[1] = w) \cdot \Pr(S_0[2] = 4 - w)
\end{aligned}$$

Following Proposition 1, the summation term in the above expression evaluates approximately to $\frac{0.965268}{N}$ 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. 4.

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 a bias in the probability of occurrence of this event, as follows.

Proposition 2. *After completion of the second round of RC4 PRGA, the state variable $S_2[2]$ equals the value $4 - z_2$ with probability*

$$\Pr\,(S_2[2] = 4 - z_2) \approx \frac{1}{N} + \frac{4/3}{N^2}.$$

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

$$\begin{aligned}\Pr(S_2[2] = 4 - z_2) &= \Pr(S_2[i_2] = 4 - S_2[S_1[j_2] + S_1[2]]) \\ &= \Pr(S_1[j_2] = 4 - S_2[S_1[j_2] + S_1[2]]) \\ &= \Pr(S_1[j_2] + S_2[S_1[j_2] + S_1[2]] = 4)\end{aligned}$$

Now, 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[2], as follows.

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

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

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

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

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

The probabilities in the above expression are verified through experimentation by running the RC4 algorithm 1 billion times, choosing a 16 byte key uniformly

[2] In the pre-proceedings version, we had considered the same cases, and had obtained the same expression for $\Pr(S_2[2] = 4 - z_2)$. However, the proof used Jenkin's bias [1] (Glimpse) in an intermediate step as a crude approximation. In this version, we present a rigorous analysis which does not require to use Jenkin's bias.

at random in each run. The probability for the first event turns out to be

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

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

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

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

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

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 situation is not possible, and all other cases deal with two distinct locations of the permutation S_1. Therefore, we obtain

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

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

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

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 $\frac{1}{N}$). For $N = 256$, the result matches with our experimental data generated from 1 billion runs of RC4 with randomly selected 16 byte keys.

4.3 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. 4, 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.

5 Conclusion

In this paper, we revisit the attack on broadcast RC4 introduced in FSE 2001 by Mantin and Shamir [5], and refute some claims made in that paper. Mantin and Shamir claimed that amongst the initial bytes of RC4 keystream, only the second one shows a bias to zero, and none of the other initial bytes has any bias (even weaker). Contrary to this claim, we prove that all the other initial keystream bytes (3 to 255 to be specific) *also* exhibit a bias to zero. It comes as a surprise to us that this observation has escaped the scrutiny of the RC4 research community for a long time.

The above biases can distinguish RC4 keystream reliably from a random stream of bytes. Further, these biases can also be exploited to mount an attack against broadcast RC4. In addition to the second plaintext byte recovery as in [5], our technique can retrieve the bytes 3 to 255 of the plaintext. The bias shown by these initial bytes also allow us to guess some state information from the RC4 keystream ($S_{r-1}[r]$ given $z_r = 0$ for $3 \leq r \leq 255$).

Further, we study the non-randomness of index j in RC4 PRGA that reveals a strong bias of j_2 towards 4. This bias in turn helps in guessing the state value $S_2[2]$ from the second keystream byte.

We would like to make a small note on a related observation. The probability calculation for event $(z_r = 0)$ in this paper was triggered by the observation that the event $(S_{r-1}[r] = r)$ is biased in the first place. There exist similar biases (though in a much weaker magnitude) in the event $(S_r[u] = v)$ for other values of u, v as well. These biases may in turn lead to corresponding biases in events $(z_r = k)$ for $k \neq 0$, but we do not study these in the scope of this paper.

Another observation that caught our attention during this work was the noticeable negative bias in $\Pr(z_1 = 0)$. Similar issues of non-random behavior in the first keystream byte z_1 has been reported earlier in [7, Section 6]. But neither [7] nor we could provide a satisfactory proof of this bias. We would like to pose this as an open problem to conclude our paper:

> *Open problem: Compute* $\Pr(z_1 = 0)$ *explicitly to support the observations made in [7] and the negative bias observed in the line of our work.*

Acknowledgment. The authors are thankful to the anonymous reviewers for their comments and suggestions that helped in improving technical and editorial details of the paper. The authors would also like to express their gratitude towards Dr. Mridul Nandi and Mr. Santanu Sarkar, who have helped improve the technical content of the paper through discussions regarding some of the probability computations.

References

1. R. J. Jenkins. ISAAC and RC4. 1996. Available at `http://burtleburtle.net/bob/rand/isaac.html`.

2. S. Maitra and G. Paul. New Form of Permutation Bias and Secret Key Leakage in Keystream Bytes of RC4. In proceedings of FSE 2008, Lecture Notes in Computer Science, Springer Verlag, Vol. 5086, pp. 253–269, 2008.
3. 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/Papers/Mantin1.zip`.
4. I. Mantin. Predicting and Distinguishing Attacks on RC4 Keystream Generator. In proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science, Springer-Verlag, Vol. 3494, pp. 491–506, 2005.
5. I. Mantin and A. Shamir. A Practical Attack on Broadcast RC4. In proceedings of FSE 2001, Lecture Notes in Computer Science, Springer-Verlag, Vol. 2355, pp. 152–164, 2001.
6. A. Maximov and D. Khovratovich. New State Recovering Attack on RC4. In proceedings of CRYPTO 2008, Lecture Notes in Computer Science, Springer, Vol. 5157, pp. 297–316, 2008.
7. I. Mironov. (Not So) Random Shuffles of RC4. In proceedings of CRYPTO 2002, Lecture Notes in Computer Science, Springer-Verlag, Vol. 2442, pp. 304–319, 2002.
8. P. Sepehrdad, S. Vaudenay and M. Vuagnoux. Discovery and Exploitation of New Biases in RC4. In proceedings of SAC 2010. Lecture Notes in Computer Science, Springer, Vol. 6544, pp. 74–91, 2011.
9. P. Sepehrdad, S. Vaudenay and M. Vuagnoux. Statistical Attack on RC4 Distinguishing WPA. Accepted at EUROCRYPT 2011.