New Form of Permutation Bias and Secret Key Leakage in Keystream Bytes of RC4

Subhamoy Maitra[1] and Goutam Paul[2]

[1] Applied Statistics Unit, Indian Statistical Institute,
Kolkata 700 108, India.
subho@isical.ac.in

[2] Department of Computer Science and Engineering,
Jadavpur University, Kolkata 700 032, India.
goutam_paul@cse.jdvu.ac.in

Abstract. Consider the permutation S in RC4. Roos pointed out in 1995 that after the Key Scheduling Algorithm (KSA) of RC4, each of the initial bytes of the permutation, i.e., $S[y]$ for small values of y, is biased towards some linear combination of the secret key bytes. In this paper, for the first time we show that the bias can be observed in $S[S[y]]$ too. Based on this new form of permutation bias after the KSA and other related results, a complete framework is presented to show that many keystream output bytes of RC4 are significantly biased towards several linear combinations of the secret key bytes. The results do not assume any condition on the secret key. We find new biases in the initial as well as in the 256-th and 257-th keystream output bytes. For the first time biases at such later stages are discovered without any knowledge of the secret key bytes. We also identify that these biases propagate further, once the information for the index j is revealed.

Keywords: Bias, Cryptanalysis, Keystream, Key Leakage, RC4, Stream Cipher.

1 Introduction

RC4 is one of the most well known stream ciphers. It has very simple implementation and is used in a number of commercial products till date. Being one of the popular stream ciphers, it has also been subjected to many cryptanalytic attempts for more than a decade. Though lots of weaknesses have already been explored in RC4 [1–8, 10–13, 15–17, 19–21], it could not be thoroughly cracked yet and proper use of this stream cipher is still believed to be quite secure. This motivates the analysis of RC4.

The Key Scheduling Algorithm (KSA) and the Pseudo Random Generation Algorithm (PRGA) of RC4 are presented below. The data structure contains an array S of size N (typically, 256), which contains a permutation of the integers $\{0, \ldots, N-1\}$, two indices i, j and the secret key array K. Given a secret key k of l bytes (typically 5 to 16), the array K of size N is such that $K[y] = k[y \bmod l]$ for any y, $0 \le y \le N-1$.

Algorithm KSA

Initialization:

For $i = 0, \ldots, N-1$

$S[i] = i$;

$j = 0$;

Scrambling:

For $i = 0, \ldots, N-1$

$j = (j + S[i] + K[i])$;

Swap$(S[i], S[j])$;

Algorithm PRGA

Initialization:

$i = j = 0$;

Output Keystream Generation Loop:

$i = i + 1$;

$j = j + S[i]$;

Swap$(S[i], S[j])$;

$t = S[i] + S[j]$;

Output $z = S[t]$;

Apart from some minor details, the KSA and the PRGA are almost the same. In the KSA, the update of the index j depends on the secret key, whereas the key is not used in the PRGA. One may consider the PRGA as the KSA with all zero key. All additions in both the KSA and the PRGA are additions modulo N.

Initial empirical works based on the weaknesses of the RC4 KSA were explored in [17, 21] and several classes of weak keys had been identified. In [17], experimental evidences of the bias of the initial permutation bytes after the KSA towards the secret key have been reported. It was also observed in [17] that the first keystream output byte of RC4 leaks information about the secret key when the first two secret key bytes add to 0 mod 256. A more general theoretical study has been performed in [11, 12] which includes the observations of [17]. These biases do propagate to the keystream output bytes as observed in [5, 11]. In [5], the Glimpse theorem [4] is used to show the propagation of biases in the initial keystream output bytes. On the other hand, a bias in the choice of index has been exploited in [11] to show a bias in the first keystream output byte.

More than a decade ago (1995), Roos [17] pointed out that the initial bytes $S[y]$ of the permutation after the KSA are biased towards some function f_y (see Section 1.1 for the definition of f_y) of the secret key. Since then several works [2, 9–12, 14] have considered biases of $S[y]$ either with functions of the secret key bytes or with absolute values and discussed applications of these biases. However, no research has so far been published to study how the bytes $S[S[y]]$ are related to the secret key for different values of y. Here we solve this problem, identifying substantial biases in this direction. It is interesting to note that as the KSA proceeds, the probabilities $P(S[y] = f_y)$ decrease monotonically, whereas the probabilities $P(S[S[y]] = f_y)$ first increases monotonically till the middle of the KSA and then decreases monotonically until the end of the KSA.

Using these results and other related techniques, we find new biases in the keystream output bytes towards the secret key. A complete framework is presented towards the leakage of information about the secret key in the keystream output bytes, that not only reveals new biases at a later stage (256, 257-th bytes), but also points out that the biases propagate further, once the information regarding j is known.

The works [2, 7] also explain how secret key information is leaked in the keystream output bytes. In [2], it is considered that the first few bytes of the secret key is known and based on that the next byte of the secret key is predicted.

The attack is based on how secret key information is leaked in the first keystream output byte of the PRGA. In [7], the same idea of [2] has been exploited with the Glimpse theorem [4] to find the information leakage about the secret key at the 257-th byte of the PRGA. Note that, our result is better than that of [7], as in [7] the bias is observed only when certain conditions on the secret key and IV hold. However, the biases we note at 256, 257-th bytes do not assume any such condition on the secret key.

1.1 Notations, Contributions and Outline

Let S_r be the permutation, i_r and j_r be the values of the indices i and j after r many rounds of the RC4 KSA, $1 \leq r \leq N$. Hence S_N is the permutation after the complete key scheduling. By S_0, we denote the initial identity permutation. During round r of the KSA, $i_r = r - 1$, $1 \leq r \leq N$, and hence the permutation S_r after round r can also be denoted by S_{i_r+1}.

Let S_r^G be the permutation, i_r^G and j_r^G be the values of the indices i and j, and z_r be the keystream output byte after r many rounds of the PRGA, $r \geq 1$. Clearly, $i_r^G = r \bmod N$. We also denote S_N by S_0^G as this is the permutation before the PRGA starts.

Further, let

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

for $y \geq 0$. Note that all the additions and subtractions related to the key bytes, the permutation bytes and the indices are modulo N.

Our contribution can be summarized as follows.

- In Section 2, we present the results related to biased association of $S_N[S_N[y]]$ towards the linear combination f_y of the secret key bytes.
- In Section 3, we present a framework for identifying biases in RC4 keystream bytes towards several linear combinations of the secret key bytes.
  - In Section 3.1, we show that $P(z_N = N - f_0)$ is not a random association. This indicates bias at z_{256}.
  - In Section 3.2, we use the bias of $S_N[S_N[1]]$ (from Section 2) to prove that $P(z_{N+1} = N + 1 - f_1)$ is not a random association. This indicates bias at z_{257}.
  - In Section 3.3, we observe new biases in the initial keystream bytes apart from the known ones [5]. It is shown that for $3 \leq r \leq 32$, $P(z_r = f_{r-1})$ are not random associations.
  - These results are taken together in Section 3.4 to present cryptanalytic applications.
- In Section 4, considering that the values of index j are leaked at some points during the PRGA, we show that biases of the keystream output bytes towards the secret key are observed at a much later stage.

2 Bias of $S[S[y]]$ to Secret Key

We start this section discussing how $P(S_r[S_r[1]] = f_1)$ varies with round $r, 1 \leq r \leq N$, during the KSA of RC4. Once again, note that $f_1 = (K[0] + K[1] + 1) \bmod N$. To motivate, we like to refer to Figure 1 that demonstrates the nature of the curve with an experimentation using 10 million randomly chosen secret keys. The probability $P(S_r[S_r[1]] = f_1)$ increases till around $r = \frac{N}{2}$ where it gets the maximum value around 0.185 and then it decreases to 0.136 at $r = N$. Note that this nature is not similar to the nature of $P(S_r[1] = f_1)$ that decreases continuously as r increases during the KSA.

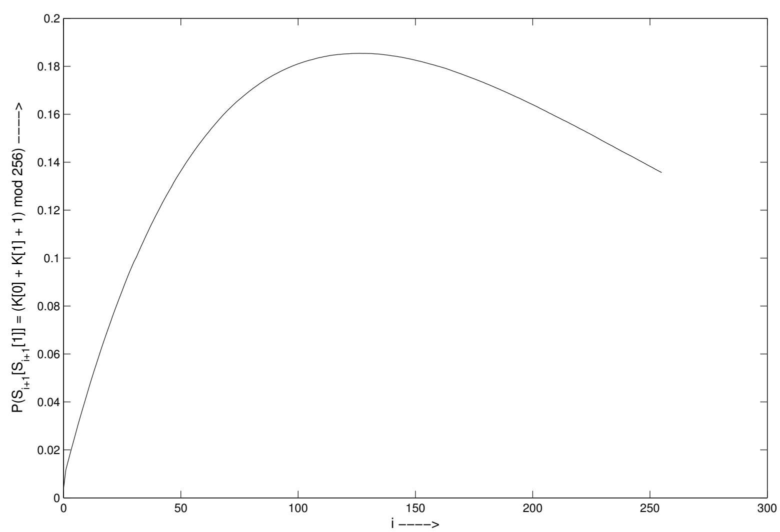

Fig. 1. $P(S_{i+1}[S_{i+1}[1]] = f_1)$ versus i $(r = i + 1)$ during RC4 KSA.

Towards the theoretical results, let us first present the base case for $r = 2$, i.e., after round 2 of the RC4 KSA.

Lemma 1. *$P(S_2[S_2[1]] = K[0] + K[1] + 1) = \frac{3}{N} - \frac{4}{N^2} + \frac{2}{N^3}$. Further, $P(S_2[S_2[1]] = K[0] + K[1] + 1 \wedge S_2[1] \leq 1) \approx \frac{2}{N}$.*

Proof. The proof is based on three cases.

1. Let $K[0] \neq 0, K[1] = N - 1$. The probability of this event is $\frac{N-1}{N^2}$. Now $S_2[1] = S_1[K[0] + K[1] + 1] = S_1[K[0]] = S_0[0] = 0$. So, $S_2[S_2[1]] = S_2[0] = S_1[0] = K[0] = K[0]+K[1]+1$. Note that $S_2[0] = S_1[0]$, as $K[0]+K[1]+1 \neq 0$. Moreover, in this case, $S_2[1] \leq 1$.

2. Let $K[0]+K[1]=0, K[0]\neq 1$, i.e., $K[1]\neq N-1$. The probability of this event is $\frac{N-1}{N^2}$. Now $S_2[1]=S_1[K[0]+K[1]+1]=S_1[1]=S_0[1]=1$. Note that $S_1[1]=S_0[1]$, as $K[0]\neq 1$. So, $S_2[S_2[1]]=S_2[1]=1=K[0]+K[1]+1$. Also, in this case, $S_2[1]\leq 1$.
3. $S_2[S_2[1]]$ could be $K[0]+K[1]+1$ by random association except the two previous cases.
   Out of that, $S_2[1]\leq 1$ will happen in $\frac{2}{N}$ proportion of cases.

Thus $P(S_2[S_2[1]]=K[0]+K[1]+1)=\frac{2(N-1)}{N^2}+(1-\frac{2(N-1)}{N^2})\frac{1}{N}=\frac{3}{N}-\frac{4}{N^2}+\frac{2}{N^3}$. Further $P(S_2[S_2[1]]=K[0]+K[1]+1\wedge S_2[1]\leq 1)=\frac{2(N-1)}{N^2}+\frac{2}{N}(1-\frac{2(N-1)}{N^2})\frac{1}{N}=\frac{2}{N}-\frac{4(N-1)}{N^4}\approx\frac{2}{N}$. □

Lemma 1 shows that after the second round ($i=1, r=2$), the event $(S_2[S_2[1]]=K[0]+K[1]+1)$ is not a random association.

Lemma 2. *Let* $p_r=P(S_r[S_r[1]]=K[0]+K[1]+1\wedge S_r[1]\leq r-1)$ *for* $r\geq 2$. *Then for* $r\geq 3$, $p_r=(\frac{N-2}{N})p_{r-1}+\frac{1}{N}\cdot(\frac{N-2}{N})\cdot(\frac{N-1}{N})^{2(r-2)}$.

Proof. After the $(r-1)$-th round is over, the permutation is S_{r-1}. In this case, $p_{r-1}=P(S_{r-1}[S_{r-1}[1]]=K[0]+K[1]+1\wedge S_{r-1}[1]\leq r-2)$. The event $\big((S_r[S_r[1]]=K[0]+K[1]+1)\wedge(S_r[1]\leq r-1)\big)$ can occur in two mutually exclusive and exhaustive ways: $\big((S_r[S_r[1]]=K[0]+K[1]+1)\wedge(S_r[1]\leq r-2)\big)$ and $\big((S_r[S_r[1]]=K[0]+K[1]+1)\wedge(S_r[1]=r-1)\big)$. We compute the contribution of each separately.

In the r-th round, $i=r-1$ and hence does not touch the indices $0,\ldots,r-2$. Thus, the event $\big((S_r[S_r[1]]=K[0]+K[1]+1)\wedge(S_r[1]\leq r-2)\big)$ occurs if we already had $\big((S_{r-1}[S_{r-1}[1]]=K[0]+K[1]+1)\wedge(S_{r-1}[1]\leq r-2)\big)$ and $j_r\notin\{1,r-1\}$. Thus, the contribution of this part is $p_{r-1}(\frac{N-2}{N})$.

The event $\big((S_r[S_r[1]]=K[0]+K[1]+1)\wedge(S_r[1]=r-1)\big)$ occurs if after the $(r-1)$-th round, $S_{r-1}[r-1]=r-1$, $S_{r-1}[1]=K[0]+K[1]+1$ and $j_r=1$ causing a swap involving the indices 1 and $r-1$.

1. We have $S_{r-1}[r-1]=r-1$ if the location $r-1$ is not touched during the rounds $i=0,\ldots,r-2$. This happens with a probability at least $(\frac{N-1}{N})^{r-1}$.
2. The event $S_{r-1}[1]=K[0]+K[1]+1$ may happen as follows. In the first round (when $i=0$), $j_1\notin\{1,K[0]+K[1]+1\}$ so that $S_1[1]=1$ and $S_1[K[0]+K[1]+1]=K[0]+K[1]+1$ with probability $(\frac{N-2}{N})$. After this, in the second round (when $i=1$), we will have $j_2=j_1+S_1[1]+K[1]=K[0]+K[1]+1$, and so after the swap, $S_2[1]=K[0]+K[1]+1$. Now, $K[0]+K[1]+1$ remains in location 1 from the end of round 2 till the end of round $(r-1)$ (when $i=r-2$) with probability $(\frac{N-1}{N})^{r-3}$. Thus, $P(S_{r-1}[1]=K[0]+K[1]+1)=(\frac{N-2}{N})\cdot(\frac{N-1}{N})^{r-3}$.
3. In the r-th round (when $i=r-1$), j_r becomes 1 with probability $\frac{1}{N}$.

Thus, $P\big((S_r[S_r[1]]=K[0]+K[1]+1)\wedge(S_r[1]=r-1)\big)=(\frac{N-1}{N})^{r-1}\cdot(\frac{N-2}{N})\cdot(\frac{N-1}{N})^{r-3}\cdot\frac{1}{N}=\frac{1}{N}\cdot(\frac{N-2}{N})\cdot(\frac{N-1}{N})^{2(r-2)}$.

Adding the above two contributions, we get $p_r=(\frac{N-2}{N})p_{r-1}+\frac{1}{N}\cdot(\frac{N-2}{N})\cdot(\frac{N-1}{N})^{2(r-2)}$. □

The recurrence in Lemma 2 along with the base case in Lemma 1 completely specify the probabilities p_r for all $r \in$[2,...,N].

Theorem 1. *After the complete KSA,*
$P(S_N[S_N[1]] = K[0] + K[1] + 1) \approx (\frac{N-1}{N})^{2(N-1)}$.

Proof. Using the approximation $\frac{N-2}{N} \approx (\frac{N-1}{N})^2$, the recurrence in Lemma 2 can be rewritten as $p_r = ap_{r-1} + a^{r-1}b$, where $a = (\frac{N-1}{N})^2$ and $b = \frac{1}{N}$. The solution of this recurrence is given by $p_r = a^{r-2}p_2 + (r-2)a^{r-1}b$, $r \geq 2$. Substituting the values of p_2 (from Lemma 1), a and b, we get $p_r = \frac{2}{N}(\frac{N-1}{N})^{2(r-2)} + (\frac{r-2}{N})(\frac{N-1}{N})^{2(r-1)}$. Substituting $r = N$ and noting the fact that $P\big((S_N[S_N[1]] = K[0] + K[1] + 1) \wedge (S_N[1] \leq N-1)\big) = P(S_N[S_N[1]] = K[0] + K[1] + 1)$, we get $P(S_N[S_N[1]] = K[0] + K[1] + 1) = \frac{2}{N}(\frac{N-1}{N})^{2(N-2)} + (\frac{N-2}{N})(\frac{N-1}{N})^{2(N-1)}$. Note that the second term (≈ 0.1348 for $N = 256$) dominates over the negligibly small first term (≈ 0.0011 for $N = 256$), and so $P(S_N[S_N[1]] = K[0] + K[1] + 1) \approx (\frac{N-1}{N})^{2(N-1)}$ (replacing $\frac{N-2}{N} = 1 - \frac{2}{N}$ by 1 in the second term). $\square$

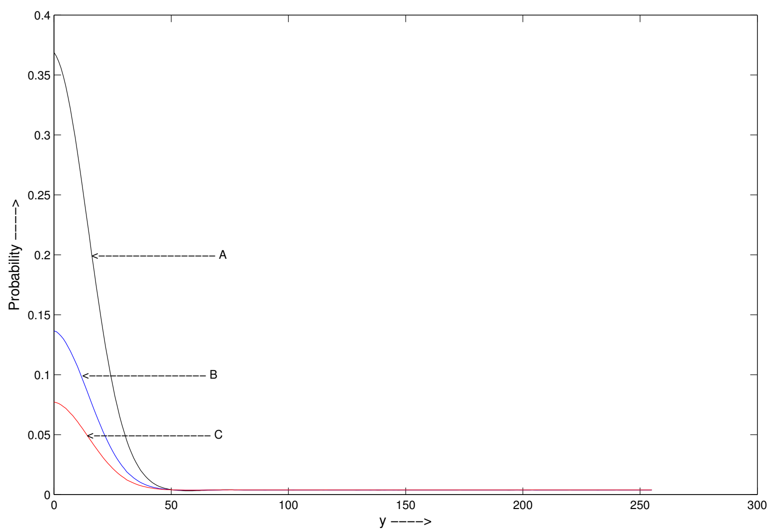

Fig. 2. A: $P(S_N[y] = f_y)$, B: $P(S_N[S_N[y]] = f_y)$, C: $P(S_N[S_N[S_N[y]]] = f_y)$ versus y ($0 \leq y \leq 255$).

Now we like to present a more detailed observation. In [17, 12], the association between $S_N[y]$ and f_y is shown. As we have observed the non-random association between $S_N[S_N[1]]$ and f_1, it is important to study what is the association between $S_N[S_N[y]]$ and f_y, and moving further, the association between

$S_N[S_N[S_N[y]]]$ and f_y, for $0 \le y \le N-1$ and so on. Our experimental observations show that these associations are not random (i.e., much more than $\frac{1}{N}$) for initial values of y. The experimental observations (over 10 million runs of randomly chosen keys) are presented in Figure 2.

The theoretical analysis of the biases of $S_r[S_r[y]]$ towards f_y for small values of y is presented in Appendix A. The results involved in the process are tedious and we need to approximate certain quantities to get the following closed form formula.

Theorem 2. *After the complete KSA,*
$P(S_N[S_N[y]] = f_y) \approx \frac{y}{N} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2(N-2)} + \frac{1}{N} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}-y+2(N-1)} + (\frac{N-y-1}{N}) \cdot (\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2N-3}$, $0 \le y \le 31$.

Extending similar techniques, the association between $S_N[S_N \ldots [S_N[y]] \ldots]$ and f_y can be studied in general. Though the general results are combinatorially interesting, it is not immediate how they will be applicable to find further weaknesses in the RC4 PRGA. In terms of cryptanalytic point of view, we use the non-random association of $S_N[S_N[1]]$ relating f_1 (Theorem 1) to obtain the bias at the 257-th keystream output byte in Section 3.2.

3 New Biases in RC4 Keystream

We will first build the framework and then present new biases in Sections 3.1, 3.2 and 3.3, which were not known earlier.

Let us consider the existing result that relates each permutation byte after the KSA with certain linear combination of the secret key bytes.

Proposition 1. *[12, Theorem 1] Consider that the index j takes its values uniformly at random during the KSA rounds. Then,* $P(S_r[y] = f_y) \approx (\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{[\frac{y(y+1)}{2}+r]} + \frac{1}{N}$, $0 \le y \le r-1$, $1 \le r \le N$.

Substituting $r = N$ in the statement of the above Proposition, we get the following.

Corollary 1. *The bias of the final permutation after the KSA towards the secret key is given by* $P(S_N[y] = f_y) = (\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{[\frac{y(y+1)}{2}+N]} + \frac{1}{N}$, $0 \le y \le N-1$.

As explained in [12], the above result indicates significant biases for small values of y (more precisely, for $0 \le y \le 47$), that is supported by the experimental result presented in [17].

The Glimpse Main Theorem [4, 7] states that after the r-th round of the PRGA, $r \ge 1$, $P(S_r^G[j_r^G] = r - z_r) = P(S_r^G[i_r^G] = j_r^G - z_r) = \frac{2}{N}$. We rewrite the first relation between $S_r^G[j_r^G]$ and $r - z_r$ as the following proposition.

Proposition 2. $P(z_r = r - S_{r-1}^G[i_r^G]) = \frac{2}{N}$, $r \ge 1$.

Proof. $S_r^G[j_r^G]$ is assigned the value of $S_{r-1}^G[i_r^G]$. As the Glimpse Main Theorem gives $P(z_r = r - S_r^G[j_r^G]) = \frac{2}{N}$ for $r \geq 1$, we get $P(z_r = r - S_{r-1}^G[i_r^G]) = \frac{2}{N}$ for $r \geq 1$. □

The idea of writing the Glimpse Main Theorem in the form of Proposition 2 is due to the fact that relating "z_r to $S_{r-1}^G[i_r^G]$" will ultimately relate "z_r to the secret key bytes", as the permutations in the initial rounds of the PRGA are related to the secret key.

Now we start with our results. The following lemma shows how the permutation bytes at rounds t and $r-1$ of the PRGA, for $t+2 \leq r$, are related.

Lemma 3. *Let $P(S_t^G[i_r^G] = X) = q_{t,r}$, for some X. Then, for $t+2 \leq r \leq t+N$, $P(S_{r-1}^G[i_r^G] = X) = q_{t,r} \cdot \left[(\frac{N-1}{N})^{r-t-1} - \frac{1}{N}\right] + \frac{1}{N}$.*

Proof. We consider two separate cases.

1. $S_t^G[i_r^G] = X$ and during the next $(r-t-1)$ rounds of the PRGA, the index i_r^G is not touched by any of the $r-t-1$ many j values (since $t+2 \leq r \leq t+N$, the index i_r^G is not touched by any of the $r-t-1$ many i values anyway). The first event occurs with probability $q_{t,r}$ and the second event occurs with probability $(\frac{N-1}{N})^{r-t-1}$. Thus the contribution of this case is $q_{t,r} \cdot (\frac{N-1}{N})^{r-t-1}$.
2. $S_t^G[i_r^G] \neq X$ and still $S_{r-1}^G[i_r^G]$ equals X by random association. The contribution of this case is $(1-q_{t,r}) \cdot \frac{1}{N}$.

Thus, adding the above two contributions, we get $P(S_{r-1}^G[i_r^G] = X) = q_{t,r} \cdot (\frac{N-1}{N})^{r-t-1} + (1-q_{t,r}) \cdot \frac{1}{N} = q_{t,r} \cdot \left[(\frac{N-1}{N})^{r-t-1} - \frac{1}{N}\right] + \frac{1}{N}$. □

Remark 1. The above result holds for $t+2 \leq r \leq t+N$, and not for $r = t+1$. If we take $r = t+1$, then $S_{r-1}^G = S_t^G$, which is our starting point, i.e., $P(S_{r-1}^G[i_r^G] = X) = P(S_t^G[i_r^G] = X) = q_{t,r}$.

The following is an immediate consequence of Lemma 3.

Corollary 2. *For $2 \leq r \leq N-1$, $P(S_{r-1}^G[r] = f_r) = \left[(\frac{N-r}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+N]} + \frac{1}{N}\right] \cdot \left[(\frac{N-1}{N})^{r-1} - \frac{1}{N}\right] + \frac{1}{N}$.*

Proof. For $2 \leq r \leq N-1$, we have $i_r^G = r$. Taking $X = f_r$ and $t = 0$ in Lemma 3, we have $q_{0,r} = P(S_0^G[r] = f_r) = P(S_N[r] = f_r) = (\frac{N-r}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+N]} + \frac{1}{N}$ (by Corollary 1), and hence $P(S_{r-1}^G[r] = f_r) = \left[(\frac{N-r}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+N]} + \frac{1}{N}\right] \cdot \left[(\frac{N-1}{N})^{r-1} - \frac{1}{N}\right] + \frac{1}{N}$. □

Next, we present the bias of each keystream output byte to a combination of the secret key bytes in the following lemma.

Lemma 4. *Let $P(S_{r-1}^G[i_r^G] = f_{i_r^G}) = w_r$, for $r \geq 1$. Then $P(z_r = r - f_{i_r^G}) = \frac{1}{N} \cdot (1 + w_r)$, $r \geq 1$.*

Proof. We consider two separate cases in which the event $(z_r = r - f_{i_r^G})$ can occur.

1. $S_{r-1}^G[i_r^G] = f_{i_r^G}$ and $z_r = r - S_{r-1}^G[i_r^G]$. The contribution of this case is $P(S_{r-1}^G[i_r^G] = f_{i_r^G}) \cdot P(z_r = r - S_{r-1}^G[i_r^G]) = w_r \cdot \frac{2}{N}$ (by Proposition 2).
2. $S_{r-1}^G[i_r^G] \neq f_{i_r^G}$, and still $z_r = r - f_{i_r^G}$ due to random association. So the contribution of this case is $P(S_{r-1}^G[i_r^G] \neq f_{i_r^G}) \cdot \frac{1}{N} = (1 - w_r) \cdot \frac{1}{N}$.

Adding the above two contributions, we get the total probability as $w_r \cdot \frac{2}{N} + (1 - w_r) \cdot \frac{1}{N} = \frac{1}{N} \cdot (1 + w_r)$. □

Some results for biases in initial keystream bytes has earlier been pointed out in [5] that has later been discussed in [19] too. We detail these biases giving explicit formula under our theoretical framework.

Theorem 3.
(1) $P(z_1 = 1 - f_1) = \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{N+2} + \frac{1}{N}\right)$.
(2) *For* $2 \leq r \leq N - 1$,
$P(z_r = r - f_r) = \frac{1}{N} \cdot \left(1 + \left[(\frac{N-r}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+N]} + \frac{1}{N}\right] \cdot \left[(\frac{N-1}{N})^{r-1} - \frac{1}{N}\right] + \frac{1}{N}\right)$.

Proof. First, we prove part (1). In the first round, i.e., when $r = 1$, we have $i_r^G = 1$ and $f_{i_r^G} = f_1$, and so $w_1 = P(S_0^G[1] = f_1) = P(S_N[1] = f_1) = (\frac{N-1}{N}) \cdot (\frac{N-1}{N})^{[\frac{1(1+1)}{2}+N]} + \frac{1}{N} = (\frac{N-1}{N})^{N+2} + \frac{1}{N}$ (by Corollary 1). Now, using Lemma 4, we get $P(z_1 = 1 - f_1) = \frac{1}{N} \cdot (1 + w_1) = \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{N+2} + \frac{1}{N}\right)$.

Next, we prove part (2). From Corollary 2, $w_r = P(S_{r-1}^G[r] = f_r) = \left[(\frac{N-r}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+N]} + \frac{1}{N}\right] \cdot \left[(\frac{N-1}{N})^{r-1} - \frac{1}{N}\right] + \frac{1}{N}$, $2 \leq r \leq N-1$. Now, using Lemma 4, we get $P(z_r = r - f_r) = \frac{1}{N} \cdot (1 + w_r) = \frac{1}{N} \cdot \left(1 + \left[(\frac{N-r}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+N]} + \frac{1}{N}\right] \cdot \left[(\frac{N-1}{N})^{r-1} - \frac{1}{N}\right] + \frac{1}{N}\right)$. □

Note that Lemma 3 or Corollary 2 is not used in proving part (1) of the above theorem. It is proved directly from Corollary 1. In fact, Lemma 3 can not be used in part (1), as here we have $r = t + 1$ with $t = 0$ (see Remark 1).

To have a clear understanding of the quantity of the biases, Table 1 lists the numerical values of the probabilities according to the formula given in Theorem 3. Note that the random association is $\frac{1}{N}$, which is 0.0039 for $N = 256$.

Close to the round 48, the biases tend to disappear. This is indicated by the convergence of the values to the probability $\frac{1}{256} = 0.0039$.

One may check that $P(z_1 = 1 - f_1) = \frac{1}{N}(1 + 0.36)$ and that decreases to $P(z_{32} = 32 - f_{32}) = \frac{1}{N}(1 + 0.05)$, but still then it is 5% more than the random association.

3.1 Bias in the 256-th keystream output byte

Interestingly, the biases again reappear after rounds 256 and 257. First we present the bias for the 256-th keystream byte.

r	$P(z_r = r - f_r)$							
1-8	0.0053	0.0053	0.0053	0.0053	0.0052	0.0052	0.0052	0.0051
9-16	0.0051	0.0050	0.0050	0.0049	0.0048	0.0048	0.0047	0.0047
17-24	0.0046	0.0046	0.0045	0.0045	0.0044	0.0044	0.0043	0.0043
25-32	0.0043	0.0042	0.0042	0.0042	0.0041	0.0041	0.0041	0.0041
33-40	0.0041	0.0040	0.0040	0.0040	0.0040	0.0040	0.0040	0.0040
41-48	0.0040	0.0040	0.0040	0.0040	0.0040	0.0039	0.0039	0.0039

Table 1. The probabilities computed following Theorem 3.

Theorem 4. $P(z_N = N - f_0) = \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{2N-1} + \frac{1}{N^2} \cdot (\frac{N-1}{N})^{N-1} - \frac{1}{N^2} + \frac{1}{N}\right)$.

Proof. During the N-th round of the PRGA, $i^G_N = N \bmod N = 0$. Taking $X = f_0$, $t = 0$ and $r = N$ in Lemma 3, we have $q_{0,N} = P(S^G_0[0] = f_0) = P(S_N[0] = f_0) = (\frac{N-1}{N})^N + \frac{1}{N}$ (by Corollary 1), and hence $w_N = P(S^G_{N-1}[0] = f_0) = \left[(\frac{N-1}{N})^N + \frac{1}{N}\right] \cdot \left[(\frac{N-1}{N})^{N-1} - \frac{1}{N}\right] + \frac{1}{N} = (\frac{N-1}{N})^{2N-1} + \frac{1}{N^2} \cdot (\frac{N-1}{N})^{N-1} - \frac{1}{N^2} + \frac{1}{N}$. Thus, by Lemma 4, the bias is given by $P(z_N = N - f_0) = \frac{1}{N} \cdot (1 + w_N) = \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{2N-1} + \frac{1}{N^2} \cdot (\frac{N-1}{N})^{N-1} - \frac{1}{N^2} + \frac{1}{N}\right)$. □

For $N = 256$, $w_N = w_{256} = 0.1392$ and the bias turns out to be 0.0045 (i.e., $\frac{1}{256}(1 + 0.1392)$). Experimental results confirm this bias.

3.2 Bias in the 257-th keystream output byte

We will now show that the bias in the 257-th keystream output byte follows from Theorem 1, i.e., $P(S_N[S_N[1]] = K[0] + K[1] + 1) \approx (\frac{N-1}{N})^{2(N-1)}$.

Theorem 5. $P(z_{N+1} = N + 1 - f_1)$
$= \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{3(N-1)} - \frac{1}{N} \cdot (\frac{N-1}{N})^{2(N-1)} + \frac{1}{N}\right)$.

Proof. During the $(N+1)$-th round, we have, $i^G_{N+1} = (N+1) \bmod N = 1$. Taking $X = f_1$, $t = 1$ and $r = N + 1$ in Lemma 3, we have $q_{1,N+1} = P(S^G_1[1] = f_1) = P(S_N[S_N[1]] = f_1) = (\frac{N-1}{N})^{2(N-1)}$, and hence $w_{N+1} = P(S^G_N[1] = f_1) = (\frac{N-1}{N})^{2(N-1)} \cdot \left[(\frac{N-1}{N})^{N-1} - \frac{1}{N}\right] + \frac{1}{N} = (\frac{N-1}{N})^{3(N-1)} - \frac{1}{N} \cdot (\frac{N-1}{N})^{2(N-1)} + \frac{1}{N}$. Now, using Lemma 4, we get $P(z_{N+1} = N + 1 - f_1) = \frac{1}{N} \cdot (1 + w_{N+1}) = \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{3(N-1)} - \frac{1}{N} \cdot (\frac{N-1}{N})^{2(N-1)} + \frac{1}{N}\right)$. □

For $N = 256$, $w_{N+1} = w_{257} = 0.0535$ and $P(z_{257} = 257 - f_1) = \frac{1}{N} \cdot (1 + 0.0535) = 0.0041$ which also conforms to experimental observation.

3.3 More biases in initial bytes of RC4 keystream

The biases of z_r with $r - f_r$ for the initial keystream output bytes have been pointed out in Theorem 3. Interestingly, experimental observation reveals bias of z_r with f_{r-1} too. The results are presented in Table 2 which is experimented over hundred million (10^8) randomly chosen keys of 16 bytes. For proper random association, $P(z_r = f_{r-1})$ should have been $\frac{1}{256}$, i.e., 0.0039.

r	$P(z_r = f_{r-1})$							
1-8	0.0043	0.0039	0.0044	0.0044	0.0044	0.0044	0.0043	0.0043
9-16	0.0043	0.0043	0.0043	0.0042	0.0042	0.0042	0.0042	0.0042
17-24	0.0041	0.0041	0.0041	0.0041	0.0041	0.0040	0.0040	0.0040
25-32	0.0040	0.0040	0.0040	0.0040	0.0040	0.0040	0.0040	0.0040
33-40	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039
41-48	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039

Table 2. Additional bias of the keystream bytes towards the secret key.

Following our experimental observation, the explanation of the fact $P(z_3 = f_2) > \frac{1}{256}$ was pointed out in [18]. We present the idea of [18] in this paragraph. Assume that after the third round of the KSA, $S_3[2]$ takes the value f_2, and is hit by j later in the KSA. Then f_2 is swapped with $S_k[k]$ and consider that $S_k[k]$ has remained k so far. Further, suppose that $S_N[3] = 0$ holds. Thus, $S_N[2] = k$, $S_N[k] = f_2$ and $S_N[3] = 0$ at the end of the KSA. In the second round of the PRGA, $S_1^G[2] = k$ is swapped with a more or less random location $S_1^G[l]$. Therefore, $S_2^G[l] = k$ and $j_2^G = l$. In the next round, $i = 3$ and points to $S_2^G[3] = 0$. So j does not change and hence $j_3^G = l = j_2^G$. Thus, $S_2^G[l] = k$ is swapped with $S_2^G[3] = 0$, and one gets $S_3^G[l] = 0$ and $S_3^G[3] = k$. The output z_3 is now $S_3^G[S_3^G[i] + S_3^G[j_3^G]] = S_3^G[k + 0] = S_3^G[k] = f_2$.

Along the same line of arguments given in [18], we here provide a detailed proof considering the event $z_r = f_{r-1}$ for $r > 2$ in general and explicitly derive a formula for $P(z_r = f_{r-1})$. The proof depends on $P(S_N[r] = 0)$ for different r values. The explicit formula for the probabilities $P(S_N[u] = v)$ was derived for the first time in [9] and the problem was addressed again in [10, 14].

Proposition 3. *[14, Theorem 1, Item 2] For $0 \le v \le N-1$, $v \le u \le N-1$, $P(S_N[u] = v) = \frac{1}{N} \cdot (\frac{N-1}{N})^{N-1-u} + \frac{1}{N} \cdot (\frac{N-1}{N})^{v+1} - \frac{1}{N} \cdot (\frac{N-1}{N})^{N+v-u}$.*

Theorem 6. *For $3 \le r \le N$, $P(z_r = f_{r-1}) = (\frac{N-1}{N}) \cdot (\frac{N-r}{N}) \cdot \left((\frac{N-r+1}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r-1)}{2}+r]} + \frac{1}{N}\right) \cdot (\frac{N-2}{N})^{N-r+1} \cdot (\frac{N-3}{N})^{r-2} \cdot \gamma_r + \frac{1}{N}$,*
where $\gamma_r = \frac{1}{N} \cdot (\frac{N-1}{N})^{N-1-r} + \frac{1}{N} \cdot (\frac{N-1}{N}) - \frac{1}{N} \cdot (\frac{N-1}{N})^{N-r}$.

Proof. Substituting $y = r-1$ in Proposition 1, we have $P(S_r[r-1] = f_{r-1}) = \alpha_r$, where $\alpha_r \approx (\frac{N-r+1}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r-1)}{2}+r]} + \frac{1}{N}$, $1 \le r \le N$. After round r, suppose that the index $r-1$ is touched for the first time by j_{t+1} in round $t+1$ of the KSA and due to the swap the value f_{r-1} is moved to the index t, $r \le t \le N-1$ and also prior to this swap the value at the index t was t itself, which now comes

to the index $r-1$. This means that from round $r+1$ to round t (both inclusive), the pseudo-random index j has not taken the values $r-1$ and t. So, after round $t+1$, $P\big((S_{t+1}[r-1]=t) \wedge (S_{t+1}[t]=f_{r-1})\big)$
$= P\big((S_t[r-1]=f_{r-1}) \wedge (S_t[t]=t) \wedge (j_{t+1}=r-1)\big)$
$= \alpha_r \cdot (\frac{N-2}{N})^{t-r} \cdot \frac{1}{N}$.
From round $t+1$ until the end of the KSA, f_{r-1} remains in index t and t remains in index $r-1$ with probability $(\frac{N-2}{N})^{N-t}$. Thus,
$P\big((S_N[r-1]=t) \wedge (S_N[t]=f_{r-1})\big)$
$= \alpha_r \cdot (\frac{N-2}{N})^{t-r} \cdot \frac{1}{N} \cdot (\frac{N-2}{N})^{N-t}$
$= \alpha_r \cdot (\frac{N-2}{N})^{N-r} \cdot \frac{1}{N} = \beta_r$ (say). Also, from Proposition 3, we have $P(S_N[r] = 0) = \gamma_r$, where $\gamma_r = \frac{1}{N} \cdot (\frac{N-1}{N})^{N-1-r} + \frac{1}{N} \cdot (\frac{N-1}{N}) - \frac{1}{N} \cdot (\frac{N-1}{N})^{N-r}$.

Suppose the indices $r-1$, t and r are not touched by the pseudo-random index j in the first $r-2$ rounds of the PRGA. This happens with probability $(\frac{N-3}{N})^{r-2}$. In round $r-1$ of the PRGA, due to the swap, the value t at index $r-1$ moves to the index j^G_{r-1} with probability 1, and $j^G_{r-1} \notin \{t, r\}$ with probability $(\frac{N-2}{N})$. Further, if $S^G_{r-1}[r]$ remains 0, then in round r of the PRGA, $j^G_r = j^G_{r-1}$ and $z_r = S^G_r\big[S^G_r[r] + S^G_r[j^G_r]\big] = S^G_r\big[S^G_{r-1}[r] + S^G_{r-1}[j^G_{r-1}]\big] = S^G_r[0+t] = S^G_r[t] = f_{r-1}$ with probability $\beta_r \cdot \gamma_r \cdot (\frac{N-3}{N})^{r-2} \cdot (\frac{N-2}{N}) = \delta_r$ (say). Since, t can values $r, r+1, r+2, \ldots, N-1$, the total probability is $\delta_r \cdot (N-r)$. Substituting the values of $\alpha_r, \beta_r, \gamma_r, \delta_r$, we get the probability that the event $(z_r = f_{r-1})$ occurs in the above path is $p = (\frac{N-r}{N}) \cdot \Big((\frac{N-r+1}{N}) \cdot (\frac{N-1}{N})^{[\frac{r(r-1)}{2}+r]} + \frac{1}{N}\Big) \cdot (\frac{N-2}{N})^{N-r+1} \cdot (\frac{N-3}{N})^{r-2} \cdot \gamma_r$.

If the above path is not followed, still there is $(1-p) \cdot \frac{1}{N}$ probability of occurrence of the event due to random association. Adding these two probabilities, we get the result. □

The theoretically computed values of the probabilities according to the above theorem match with the estimated values provided in Table 2. It will be interesting to justify the bias at $r = 1$ and the absence of the bias at $r = 2$ as observed experimentally in Table 2. These two cases are not covered in Theorem 6.

3.4 Cryptanalytic applications

Here we accumulate the results explained above. Consider the first keystream output byte z_1 of the PRGA. We find the theoretical result that $P(z_1 = 1 - f_1) = 0.0053$ (see Theorem 3 and Table 1) and the experimental observation that $P(z_1 = f_0) = 0.0043$ (see Table 2). Further, from [11], we have the result that $P(z_1 = f_2) = 0.0053$. Taking them together, one can check that the $P(z_1 = f_0 \vee z_1 = 1 - f_1 \vee z_1 = f_2) = 1 - (1 - 0.0043) \cdot (1 - 0.0053) \cdot (1 - 0.0053) = 0.0148$. (The independence assumption in calculating the probability is supported by detailed experimentation with 100 different runs, each run presenting the average probability considering 10 million randomly chosen secret keys of 16 bytes.) Our result indicates that out of randomly chosen 10000 secret keys, in 148 cases on an average, z_1 reveals f_0 or $1 - f_1$ or f_2, i.e., $K[0]$ or $1 - (K[0] + K[1] + 1)$ or $(K[0] + K[1] + K[2] + 3)$. If, however, one tries a random association, considering that z_1 will be among three randomly chosen values v_1, v_2, v_3 from $[0, \ldots, 255]$, then $P(z_1 = v_1 \vee z_1 = v_2 \vee z_1 = v_3) = 1 - (1 - \frac{1}{256})^3 = 0.0117$. Thus one can guess z_1 with an additional advantage of $\frac{0.0148-0.0117}{0.0117} \cdot 100\% = 27\%$ over the random guess.

Looking at z_2, we have $P(z_2 = 2 - f_2) = 0.0053$ (see Theorem 3 and Table 1), which provides an advantage of $\frac{0.0053-0.0039}{0.0039} \cdot 100\% = 36\%$.

Similarly, referring to Theorem 3 and Theorem 6 (and also Table 1 and Table 2), significant biases can be observed in $P(z_r = f_{r-1} \vee z_r = r - f_r)$ for $r = 3$ to 32 over random association.

Now consider the following scenario with the events $E_1, \ldots, E_{32}$, where $E_1 : (z_1 = f_0 \vee z_1 = 1 - f_1 \vee z_1 = f_2)$, $E_2 : (z_2 = 2 - f_2)$, and $E_r : (z_r = f_{r-1} \vee z_r = r - f_r)$ for $3 \leq r \leq 32$. Observing the first 32 keystream output bytes $z_1, \ldots, z_{32}$, one may try to guess the secret key assuming that 3 or more of the events $E_1, \ldots, E_{32}$ occur. We experimented with 10 million randomly chosen secret keys of length 16 bytes. We found that 3 or more of the events occur in 0.0028 proportion of cases, which is true for 0.0020 proportion of cases for random association. This demonstrates a substantial advantage (40%) over random guess.

4 Further Biases when j is Known during PRGA

In all the currently known biases as well as in all the new biases discussed in this paper so far, it is assumed that the value of the pseudo-random index j is unknown. In this section, we are going to show that the biases in the permutation at some stage of the PRGA propagates to the keystream output bytes at a later stage, if the value of the index j at the earlier stage is known.

Suppose that we know the value j_t^G of j after the round t in the PRGA. With high probability, the value V at the index j_t^G will remain there until j_t^G is touched by the deterministic index i for the first time after a few more rounds depending on what was the position of i at the t-th stage. This immediately leaks V in keystream output byte. More importantly, if the value V is biased to the secret key bytes, then that information will be leaked too.

Formally, let $P(S_t^G[j_t^G] = V) = \eta_t$ for some V. j_t^G will be touched by i in round r, where $r = j_t^G$ or $N + j_t^G$ depending on whether $j_t^G > t$ or $j_t^G \leq t$ respectively. By Lemma 3, we would have $P(S_{r-1}^G[j_t^G] = V) = \eta_t \cdot \left[(\frac{N-1}{N})^{r-t-1} - \frac{1}{N}\right] + \frac{1}{N}$. Now, Lemma 4 immediately gives
$P(z_r = r - V) = \frac{1}{N} \cdot \left(1 + \eta_t \cdot \left[(\frac{N-1}{N})^{r-t-1} - \frac{1}{N}\right] + \frac{1}{N}\right)$.

For some special V's, the form of η_t may be known. In that case, it will be advantageous to probe the values of j at particular rounds. For example, according to Corollary 2, after the $(t-1)$-th round of the PRGA, $S_{t-1}^G[t]$ is biased to the linear combination f_t of the secret key bytes with probability $\eta_t = \left[(\frac{N-t}{N}) \cdot (\frac{N-1}{N})^{[\frac{t(t+1)}{2}+N]} + \frac{1}{N}\right] \cdot \left[(\frac{N-1}{N})^{t-1} - \frac{1}{N}\right] + \frac{1}{N}$. Now, at round t, f_t would move to the index j_t due to the swap, and hence $S_t^G[j_t]$ will be biased to f_t with the same probability. So, the knowledge of j_t will leak information about f_t in round j_t^G or $N + j_t^G$ according as $j_t^G > t$ or $j_t^G \leq t$ respectively.

If we know the values of j at multiple stages of the PRGA (it may be possible to read some values of j through side-channel attacks), then the biases propagate further down the keystream output bytes. The following example illustrates how the biases propagate down the keystream output bytes when single as well as multiple j^G values are known.

Example 1. Suppose we know the value of j_5^G as 18. With probability η_5, $S_4^G[5]$ would have remained f_5 which would move to index 18 due to the swap in round 5, i.e., $S_5^G[18] = f_5$. With approximately $\eta_5 \cdot \left[(\frac{N-1}{N})^{18-5-1} - \frac{1}{N} \right] + \frac{1}{N}$ probability, f_5 would remain in index 18 till the end of the round 18-1 = 17. So, we immediately get a bias of z_{18} with $18 - f_5$.

Moreover, in round 18, f_5 would move from index 18 to j_{18}^G. So, if the value of j_{18}^G is also known, say $j_{18}^G = 3$, then we have $S_{18}^G[3] = f_5$. We can apply the same line of arguments for round $256 + 3 = 259$ to get a bias of z_{259} with $259 - f_5$.

Experiments with 1 billion random keys demonstrate that in this case the bias of z_{18} towards $18 - f_5$ is 0.0052 and the bias of z_{259} towards $259 - f_5$ is 0.0044. These conform to the theoretical values and show that the knowledge of j during the PRGA helps in finding non-random association (away from $\frac{1}{256} = 0.0039$) between the keystream output bytes and the secret key.

5 Conclusion

In this paper, we present several new observations on weaknesses of RC4. It is shown that biases towards the secret key exists at the permutation bytes $S[S[y]]$ for different y values. To our knowledge, this is the first attempt to formally analyze the biases of $S[S[y]]$ and its implications towards the security of RC4. Moreover, a framework is built to analyze biases of the keystream output bytes towards different linear combinations of the secret key bytes. Subsequently, theoretical results are proved to show that RC4 keystream output bytes at the indices 1 to 32 leak significant information about the secret key bytes. We also discovered and proved new biases towards the secret key at the 256-th and the 257-th keystream output bytes. Moreover, we show that if one knows the value of j during some rounds of the PRGA, then the biases propagate further down the keystream.

Acknowledgment: The authors like to thank Mr. Snehasis Mukherjee, Indian Statistical Institute, Kolkata for his support in the preparation of the graphs.

References

1. S. R. Fluhrer and D. A. McGrew. Statistical Analysis of the Alleged RC4 Keystream Generator. FSE 2000, pages 19-30, vol. 1978, Lecture Notes in Computer Science, Springer-Verlag.
2. S. R. Fluhrer, I. Mantin and A. Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. Selected Areas in Cryptography 2001, pages 1-24, vol. 2259, Lecture Notes in Computer Science, Springer-Verlag.

3. J. Golic. Linear statistical weakness of alleged RC4 keystream generator. EUROCRYPT 1997, pages 226-238, vol. 1233, Lecture Notes in Computer Science, Springer-Verlag.
4. R. J. Jenkins. ISAAC and RC4. 1996. Available at `http://burtleburtle.net/bob/rand/isaac.html`.
5. A. Klein. Attacks on the RC4 stream cipher. February 27, 2006. Available at `http://cage.ugent.be/~ klein/RC4/`, [last accessed on June 27, 2007].
6. I. Mantin and A. Shamir. A Practical Attack on Broadcast RC4. FSE 2001, pages 152-164, vol. 2355, Lecture Notes in Computer Science, Springer-Verlag.
7. I. Mantin. A Practical Attack on the Fixed RC4 in the WEP Mode. ASIACRYPT 2005, pages 395-411, vol. 3788, Lecture Notes in Computer Science, Springer-Verlag.
8. I. Mantin. Predicting and Distinguishing Attacks on RC4 Keystream Generator. EUROCRYPT 2005, pages 491-506, vol. 3494, Lecture Notes in Computer Science, Springer-Verlag.
9. I. Mantin. Analysis of the stream cipher RC4. Master's Thesis, The Weizmann Institute of Science, Israel, 2001.
10. I. Mironov. (Not So) Random Shuffles of RC4. CRYPTO 2002, pages 304-319, vol. 2442, Lecture Notes in Computer Science, Springer-Verlag.
11. 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. Proceedings of the International Workshop on Coding and Cryptography 2007, pages 285-294.
12. G. Paul and S. Maitra. RC4 State Information at Any Stage Reveals the Secret Key. IACR Eprint Server, eprint.iacr.org, number 2007/208, June 1, 2007. This is an extended version of [13].
13. G. Paul and S. Maitra. Permutation after RC4 Key Scheduling Reveals the Secret Key. In *Selected Areas in Cryptography, 14th International Workshop, SAC 2007*, pages 360–377, volume 4876, Lecture Notes in Computer Science, Springer Verlag, 2007.
14. G. Paul, S. Maitra and R. Srivastava. On Non-Randomness of the Permutation after RC4 Key Scheduling. In *Applied Algebra, Algebraic Algorithms, and Error Correcting Codes (AAECC-17)*, pages 100–109, volume 4851, Lecture Notes in Computer Science, Springer Verlag, 2007.
15. S. Paul and B. Preneel. Analysis of Non-fortuitous Predictive States of the RC4 Keystream Generator. INDOCRYPT 2003, pages 52-67, vol. 2904, Lecture Notes in Computer Science, Springer-Verlag.
16. S. Paul and B. Preneel. A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher. FSE 2004, pages 245-259, vol. 3017, Lecture Notes in Computer Science, Springer-Verlag.
17. 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://marcel.wanda.ch/Archive/WeakKeys`.
18. E. Tews. Email Communications, September, 2007.
19. E. Tews, R. P. Weinmann and A. Pyshkin. Breaking 104 bit WEP in less than 60 seconds. IACR Eprint Server, eprint.iacr.org, number 2007/120, April 1, 2007.
20. S. Vaudenay and M. Vuagnoux. Passive-only key recovery attacks on RC4. In *14th Annual Workshop on Selected Areas in Cryptography*, SAC 2007, August 16-17, Ottawa, Canada.
21. D. Wagner. My RC4 weak keys. Post in sci.crypt, message-id `447o1l$cbj@cnn.Princeton.EDU`, 26 September, 1995. Available at `http://www.cs.berkeley.edu/∼daw/my-posts/my-rc4-weak-keys`.

Appendix A

Lemma 5. $P\big((S_{y+1}[S_{y+1}[y]] = f_y) \wedge (S_{y+1}[y] \leq y)\big) \approx \big(\frac{1}{N} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}}\big) \cdot \big(y(\frac{N-2}{N})^{y-1} + (\frac{N-1}{N})^{y}\big)$, $0 \leq y \leq 31$.

Proof. $S_{y+1}[y] \leq y$ means that it can take $y+1$ many values $0, 1, \ldots, y$. Suppose $S_{y+1}[y] = x$, $0 \leq x \leq y-1$. Then $S_{y+1}[x]$ can equal f_y in the following way.

1. From round 1 (when $i = 0$) through x (when $i = x-1$), j does not touch the indices x and f_y. Thus, after round x, $S_x[x] = x$ and $S_x[f_y] = f_y$. This happens with probability $(\frac{N-2}{N})^x$.
2. In round $x+1$ (when $i = x$), j_{x+1} becomes equal to f_y, and after the swap, $S_{x+1}[x] = f_y$ and $S_{x+1}[f_y] = x$. The probability of this event is $P(j_{x+1} = f_y) = \frac{1}{N}$.
3. From round $x+2$ (when $i = x+1$) through y (when $i = y-1$), again j does not touch the indices x and f_y. This, after round y, $S_y[x] = f_y$ and $S_y[f_y] = x$. This occurs with probability $(\frac{N-2}{N})^{y-x-1}$.
4. In round $y+1$ (when $i = y$), j_{y+1} becomes equal to f_y, and after the swap, $S_{y+1}[y] = S_y[f_y] = x$ and $S_{y+1}[S_{y+1}[y]] = S_{y+1}[x] = S_y[x] = f_y$. This happens with probability $P(j_{y+1} = f_y)$ which is approximately equal to $(\frac{N-1}{N})^{\frac{y(y+1)}{2}}$ for small values of y as in the proof of [12, Lemma 1]. We consider $0 \leq y \leq 31$ for good approximation.

Considering the above events to be independent, we have
$P\big((S_{y+1}[S_{y+1}[y]] = f_y) \wedge (S_{y+1}[y] = x)\big)$
$= (\frac{N-2}{N})^x \cdot \frac{1}{N} \cdot (\frac{N-2}{N})^{y-x-1} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}} = (\frac{1}{N}) \cdot (\frac{N-2}{N})^{y-1} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}}$.
Summing for all x in $[0, \ldots, y-1]$, we get $P\big((S_{y+1}[S_{y+1}[y]] = f_y) \wedge (S_{y+1}[y] \leq y-1)\big) = (\frac{y}{N}) \cdot (\frac{N-2}{N})^{y-1} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}}$.

If $S_{y+1}[y] = y$, then $S_{y+1}[S_{y+1}[y]]$ can equal f_y in the following ways: (a) f_y has to be equal to y; this happens with probability $\frac{1}{N}$, (b) index y is not touched by j in any of the first y rounds; this happens with probability $(\frac{N-1}{N})^y$, and (c) in the $(y+1)$-th round, $j_{y+1} = f_y$ so that there is no swap; this happens with probability $(\frac{N-1}{N})^{\frac{y(y+1)}{2}}$. Hence, $P\big((S_{y+1}[S_{y+1}[y]] = f_y) \wedge (S_{y+1}[y] = y)\big) = (\frac{1}{N}) \cdot (\frac{N-1}{N})^y \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}}$.

Adding the above two contributions (one for $0 \leq S_{y+1}[y] \leq y-1$ and the other for $S_{y+1}[y] = y$), we get $P\big((S_{y+1}[S_{y+1}[y]] = f_y) \wedge (S_{y+1}[y] \leq y)\big) = \big(\frac{1}{N} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}}\big) \cdot \big(y(\frac{N-2}{N})^{y-1} + (\frac{N-1}{N})^y\big)$. □

Lemma 6. *Let* $p_r(y) = P\big((S_r[S_r[y]] = f_y) \wedge (S_r[y] \leq r-1)\big)$, $0 \leq y \leq N-1$, $1 \leq r \leq N$. *Then* $p_r(y) = (\frac{N-2}{N})p_{r-1}(y) + \frac{1}{N} \cdot (\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2r-3}$, $0 \leq y \leq 31$, $y+2 \leq r \leq N$.

Proof. Then event $\big((S_r[S_r[y]] = f_y) \wedge (S_r[y] \leq r-1)\big)$, where $r \geq y+2$, can occur in two mutually exclusive and exhaustive ways: $\big((S_r[S_r[y]] = f_y) \wedge (S_r[y] \leq r-2)\big)$

and $\big((S_r[S_r[y]] = f_y) \wedge (S_r[y] = r-1)\big)$. We compute the contribution of each separately.

In the r-th round, $i = r-1$ and hence does not touch the indices $0, \ldots, r-2$. Hence the event $\big((S_r[S_r[y]] = f_y) \wedge (S_r[y] \leq r-2)\big)$ occurs if we already had $\big((S_{r-1}[S_{r-1}[y]] = f_y) \wedge (S_{r-1}[y] \leq r-2)\big)$ and $j_r \notin \{y, S_{r-1}[y]\}$. Thus, the contribution of this part is $p_{r-1}(y) \cdot (\frac{N-2}{N})$.

The event $\big((S_r[S_r[y]] = f_y) \wedge (S_r[y] = r-1)\big)$ occurs if after the $(r-1)$-th round, $S_{r-1}[r-1] = r-1$, $S_{r-1}[y] = f_y$ and in the r-th round (i.e., when $i = r-1$), $j_r = y$ causing a swap involving the indices y and $r-1$.

1. We have $S_{r-1}[r-1] = r-1$ if the location $r-1$ is not touched during the rounds $i = 0, \ldots, r-2$. This happens with probability $(\frac{N-1}{N})^{r-1}$.
2. The event $S_{r-1}[y] = f_y$ happens with a probability which is approximately equal to $(\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{[\frac{y(y+1)}{2}+r-2]}$ for small values of y as in the proof of [12, Theorem 1]. We consider $0 \leq y \leq 31$ for good approximation.
3. In the r-th round (when $i = r-1$), j_r becomes y with probability $\frac{1}{N}$.

Thus, $P\big((S_r[S_r[y]] = f_y) \wedge (S_r[y] = r-1)\big) = (\frac{N-1}{N})^{r-1} \cdot (\frac{N-y}{N})(\frac{N-1}{N})^{[\frac{y(y+1)}{2}+r-2]} \cdot \frac{1}{N} = \frac{1}{N} \cdot (\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2r-3}$.

Adding the above two contributions, we get

$p_r(y) = (\frac{N-2}{N})p_{r-1}(y) + \frac{1}{N} \cdot (\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2r-3}$. □

The recurrence in Lemma 6 and the base case in Lemma 5 completely specify the probabilities $p_r(y)$ for all y in $[0, \ldots, 31]$ and r in $[y+1, \ldots, N]$.

Theorem 2 (Section 2): *After the complete KSA,*
$P(S_N[S_N[y]] = f_y) \approx \frac{y}{N} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2(N-2)} + \frac{1}{N} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}-y+2(N-1)} + (\frac{N-y-1}{N}) \cdot (\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2N-3}$, $0 \leq y \leq 31$.

Proof. Using the approximation $\frac{N-2}{N} \approx (\frac{N-1}{N})^2$, the recurrence in Lemma 6 can be rewritten as $p_r(y) = (\frac{N-1}{N})^2 p_{r-1}(y) + \frac{1}{N}(\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2r-3}$, i.e., $p_r(y) = a p_{r-1}(y) + a^{r-1} b$, where $a = (\frac{N-1}{N})^2$ and $b = \frac{1}{N}(\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}-1}$. The solution of this recurrence is $p_r(y) = a^{r-y-1} p_{y+1}(y) + (r-y-1)a^{r-1}b$, $r \geq y+1$. Substituting the values of $p_{y+1}(y)$ (from Lemma 5), a and b, we get $p_r(y) = \frac{y}{N} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2(r-2)} + \frac{1}{N} \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}-y+2(r-1)} + (\frac{r-y-1}{N}) \cdot (\frac{N-y}{N}) \cdot (\frac{N-1}{N})^{\frac{y(y+1)}{2}+2r-3}$, $y+1 \leq r \leq N$, for initial values of y ($0 \leq y \leq 31$). Substituting $r = N$ and noting the fact that $P\big((S_N[S_N[y]] = f_y) \wedge (S_N[y] \leq N-1)\big) = P(S_N[S_N[y]] = f_y)$, we get the result. □

Even after the approximation, our theoretical formula matches closely with the experimental results for $0 \leq y \leq 31$.