Many Keystream Bytes of RC4 Leak Secret Key Information

Subhamoy Maitra*, Goutam Paul†

Abstract

In this paper, we show that RC4 keystream leaks secret key information in the first 32 bytes and also in the 256-th and 257-th bytes. For the first time these many keystream output bytes are found to be significantly biased towards several linear combinations of the secret key bytes, without assuming any condition on the secret key.

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

1 Introduction

RC4 is one of the most well known stream ciphers. It has very simple implementation and it is being 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 [2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17], it could not be thoroughly cracked yet and proper use of this stream cipher is still believed to be quite secure.

The Key Scheduling Algorithm (KSA) and the Pseudo Random Generation Algorithm (PRGA) of RC4 are presented below. The data structure contains an array of size N (in practice 256 which is followed in this paper) with each location having an integer in the range of $[0, \ldots, N-1]$, 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, $0 \leq y \leq N-1$.

*Applied Statistics Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India, Email: subho@isical.ac.in

†Department of Computer Science and Engineering, Jadavpur University, Kolkata 700 032, India, Email: goutam_paul@cse.jdvu.ac.in

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 same. In KSA, the updation of the index j depends on the secret key, whereas the key is not used in PRGA. One may consider the PRGA as the KSA with all zero key.

There are two broad approaches in the study of cryptanalysis of RC4: attacks based on the weaknesses of the KSA and those based on the weaknesses of the PRGA. Distinguishing attacks are the main motivation for PRGA-based approach [2, 4, 8, 9, 10, 14, 15]. Important results in this approach include bias in the keystream output bytes. Initial works on distinguishing the RC4 keystream from random stream has been done in [4, 2]. A bias in the second output byte being zero has been proved in [8] and a bias in the equality of the first two output bytes has been shown in [15]. In [10], it has been shown that getting strings of pattern $ABGAB$ (A, B are bytes and G is a string of bytes of small length, say ≤ 16) are more probable in RC4 keystream than in random stream. In [11], RC4 has been analyzed using the theory of random shuffles and it has been recommended that initial 512 bytes of the keystream output should be discarded in order to be safe. Recently, differential attacks on RC4 have been discussed in [1, Section 5].

Initial empirical works based on the weaknesses of the RC4 KSA were done in [16, 17] and several classes of weak keys had been identified. In [16], experimental evidences have been reported that the first keystream output byte of RC4 leaks information about secret key when the first two secret key bytes add to 0 mod 256. Recently, a more general theoretical study has been performed in [12] which includes the observations of [16]. The work [12] shows how the bias of the "third permutation byte" (after the KSA) towards the "first three secret key bytes" propagates to the first keystream output byte (in the PRGA). This is proved for any secret key, i.e., there is no condition on the secret key bytes. The exact result depicts that the first keystream output byte is three more than the sum of first three secret key bytes with a probability $(1 + 0.37)\frac{1}{N}$. Very recently [13], it has been identified that if the permutation after the KSA, or the permutation at any stage of PRGA with relevant information about the indices i, j are known, then the secret key bytes can be recovered efficiently.

The works [3, 9] also explain how secret key information is leaked in the keystream output bytes. In [3], it is considered that the first few bytes of the secret key is known (this is practical as in one mode of WEP the IV bytes and the secret key bytes are concatenated to get the secret key of RC4) 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 PRGA. In [9], the same idea of [3] has been exploited with the Glimpse theorem [5] to find the information leakage about the secret key at the 257-th byte of the PRGA. Later, the work [6] improves [3].

1.1 Our Contribution

Let S_r be the permutation after r many rounds of the KSA, $r \geq 1$. So S_0 is the initial permutation and S_N is the permutation after the complete key scheduling. Let S_r^G be the permutation, i_r and j_r 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, $i_r = r \bmod N$. 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 are modulo N.

Our contribution can be summarized as follows.

- We theoretically prove that $P(z_r = r - f_r) \approx \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+(N+r-2)]} \cdot (\frac{N-r}{N})\right)$, for initial values of $r \geq 1$ and this bias is significant till $r = 32$.
- Using similar arguments, we show that $P(z_N = N - f_0) \approx \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{2(N-1)}\right)$, and $P(z_{N+1} = N + 1 - f_1) \approx \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{3(N-1)}\right)$. These indicate biases at z_{256} and z_{257}.
- We find additional experimental biases in the initial keystream bytes showing $P(z_r = f_{r-1}) \geq \frac{1}{256} \cdot (1 + 0.05)$, for $r = 1$ to 21, except $r = 2$.
- Finally, we accumulate these results to present how one can guess the secret key from the keystream output bytes with non-negligible biases than random guess.

The works presented in [16, 3, 9, 6] assume certain conditions on the secret key bytes of RC4 in order to mount their attacks. We do not consider any such requirements here. Further, our work is much more general than [12], as in [12] the bias in the first keystream output byte towards the secret key is reported, whereas here we show that there exist significant biases in many of the keystream output bytes (bytes 1 to 32 and also 256, 257).

2 New Biases in RC4 Keystream

In this section we present new biases of the RC4 keystream bytes towards the secret key. We first present the theoretical results and then continue with experimental studies followed by cryptanalytic applications of theoretical as well as experimental biases.

2.1 Theoretical Results

In [12], a bias of the first output byte of RC4 towards the first three bytes of the secret key is theoretically proved. In this section, we prove many more additional biases of the keystream output bytes of RC4 towards the secret key. First let us refer to some existing related results.

Proposition 1 *[13] Consider that the index j takes its values uniformly at random during the KSA rounds. Then, $P(S_N[y] = f_y) \approx (\frac{N-1}{N})^{[\frac{y(y+1)}{2}+(N-1)]} \cdot (\frac{N-y}{N})$, for initial values of y.*

As explained in [13], the above result corresponds to the experimental result presented in [16] for small values of y, and more precisely for $0 \leq y \leq 47$.

The Glimpse Main Theorem [5, 9] states that after the r-th round of the PRGA, $r \geq 1$, $P(S_r^G[j_r] = r - z_r) \approx P(S_r^G[i_r] = j_r^G - z_r) \approx \frac{2}{N}$. We rewrite the first relation between $S_r^G[j_r]$ and $r - z_r$ as the following proposition.

Proposition 2 *$P(z_r = r - S_{r-1}^G[i_r]) \approx \frac{2}{N}$ for $r \geq 1$.*

Proof: One may note that $S_r^G[j_r]$ is assigned the value of $S_{r-1}^G[i_r]$. As the Glimpse Main Theorem gives $P(z_r = r - S_r^G[j_r]) \approx \frac{2}{N}$ for $r \geq 1$, we get $P(z_r = r - S_{r-1}^G[i_r]) \approx \frac{2}{N}$ 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 $S_{r-1}^G[i_r]$" will ultimately relate "z_r with secret key bytes" as "initial values of the permutations in initial rounds of PRGA" are related to "secret key".

The following lemma shows that the permutation bytes $S_{r-1}^G[i_r]$ (used in Proposition 2 above) are biased to the secret key.

Lemma 1 *Suppose after the completion of the KSA, $P(S_N[y] = f_y) = p_y$, $0 \leq y \leq N-1$. Then, $P(S_{r-1}^G[i_r] = f_{i_r}) = p_{i_r} \cdot (\frac{N-1}{N})^{r-1} + (1 - p_{i_r}) \cdot \frac{1}{N}$ for $r \geq 1$.*

Proof: During the first $(r-1)$ rounds of PRGA, the value $S_{r-1}^G[i_r]$ will not be touched by the index i. Thus, the value $S_{r-1}^G[i_r]$ remains the same as $S_N[i_r]$ if the index i_r is not touched by any of the previous $r-1$ many j values, the probability of which is $(\frac{N-1}{N})^{r-1}$. Thus the contribution towards $P(S_{r-1}^G[i_r] = f_{i_r})$ will be $p_{i_r} \cdot (\frac{N-1}{N})^{r-1}$ as $P(S_N[i_r] = f_{i_r}) = p_{i_r}$.

The other part of the contribution will come when $S_N[i_r] \neq f_{i_r}$, but $S_r^G[j_r] = f_{i_r}$ by random association. This will happen with probability $(1 - p_{i_r}) \cdot \frac{1}{N}$.

Thus, $P(S_{r-1}^G[i_r] = f_{i_r}) = p_{i_r} \cdot (\frac{N-1}{N})^{r-1} + (1 - p_{i_r}) \cdot \frac{1}{N}$. ■

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

Lemma 2 *Let $P(S_{r-1}^G[i_r] = f_{i_r}) = w_r$, for $r \geq 1$. Then $P(z_r = r - f_{i_r}) \approx \frac{1}{N} \cdot (1 + w_r)$.*

Proof: $P(z_r = r - f_{i_r})$
$= P(z_r = r - f_{i_r} \wedge S_{r-1}^G[i_r] = f_{i_r}) + P(z_r = r - f_{i_r} \wedge S_{r-1}^G[i_r] \neq f_{i_r})$
$= P(z_r = r - S_{r-1}^G[i_r] \wedge S_{r-1}^G[i_r] = f_{i_r}) + P(z_r = r - f_{i_r} \wedge S_{r-1}^G[i_r] \neq f_{i_r})$

$\approx P(z_r = r - S^G_{r-1}[i_r]) \cdot P(S^G_{r-1}[i_r] = f_{i_r})$
$\quad + P(z_r = r - f_{i_r}) \cdot P(S^G_{r-1}[i_r] \neq f_{i_r})$ (assuming independence)
$\approx \frac{2}{N} \cdot w_r + \frac{1}{N} \cdot (1 - w_r)$ ($\frac{2}{N}$ from Proposition 2 and $\frac{1}{N}$ considering random association)
$= \frac{1}{N} \cdot (1 + w_r)$. ■

We will now explicitly compute the bias of z_r towards the secret key bytes for the first few rounds.

Theorem 1 *$P(z_r = r - f_r) \approx \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+(N+r-2)]} \cdot (\frac{N-r}{N})\right)$, for initial values of $r \geq 1$.*

Proof: From Lemma 1, we have $w_r \approx p_{i_r} \cdot (\frac{N-1}{N})^{r-1}$ (neglecting the term $(1 - p_{i_r}) \cdot \frac{1}{N}$) for any $r \geq 1$. For small values of r, we have $i_r = r$ and so $p_{i_r} = p_r$ and $f_{i_r} = f_r$. Again, for small values of $r \geq 0$, Proposition 1 gives $p_r \approx (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+(N-1)]} \cdot (\frac{N-r}{N})$, and hence $w_r \approx (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+(N-1)]} \cdot (\frac{N-r}{N}) \cdot (\frac{N-1}{N})^{r-1} = (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+(N+r-2)]} \cdot (\frac{N-r}{N})$. Now, using Lemma 2, we get $P(z_r = r - f_r) \approx \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{[\frac{r(r+1)}{2}+(N+r-2)]} \cdot (\frac{N-r}{N})\right)$. ■

Table 1 lists the biases according to the formula given in Theorem 1.

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

Table 1: The probabilities computed following Theorem 1.

We also run experiments with ten million different secret keys of length 16 bytes each and get the values of the biases as given in Table 2. The values in Table 1 and Table 2 are very close justifying that our theoretical approximation in Theorem 1 is very tight.

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

Table 2: Experimentally observed biases corresponding to Theorem 1.

From round 32 onwards, the biases tend to disappear. This is indicated by the convergence of the theoretically computed as well as the experimentally observed values to the

probability $\frac{1}{256} = 0.0039$ beyond round 32. One may check that $P(z_1 = 1 - f_1) \approx \frac{1}{N}(1+0.36)$ and that decreases to $P(z_{32} = 32 - f_{32}) \approx \frac{1}{N}(1 + 0.05)$, but still then it is 5% more than the random association.

Interestingly, the bias again reappears after round 256 and round 257. First we present a bias for the 256-th keystream byte.

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

Proof: During the N-th round, we have, $i_N = N \bmod N = 0$. Further, $S^G_{N-1}[i_N]$, i.e., $S^G_{N-1}[0]$ will be the same as $S_N[0]$ if the index 0 is not touched by any of the previous $N-1$ many j values, the probability of which is $(\frac{N-1}{N})^{N-1}$. Since $P(S_N[0] = f_0) = p_0 \approx (\frac{N-1}{N})^{N-1}$, therefore $w_N = P(S^G_{N-1}[i_N] = f_0) \approx p_0 \cdot (\frac{N-1}{N})^{N-1} = (\frac{N-1}{N})^{2(N-1)}$. Then from Lemma 2, the bias is given by $P(z_N = N - f_0) \approx \frac{1}{N} \cdot (1 + w_N) = \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{2(N-1)}\right)$. ■

For $N = 256$, $w_N \approx 0.136$ and the bias turns out to be 0.0044 (i.e., $\frac{1}{256}(1 + 0.13)$) approximately. Experimental results confirm these theoretical values.

Using similar arguments, we can show that there exists a very small bias in $(N+1)$-th output byte also. During the $(N+1)$-th round, we have, $i_{N+1} = (N+1) \bmod N = 1$. Moreover, $S^G_N[i_{N+1}]$, i.e., $S^G_N[1]$ will be the same as $S^G_1[1]$ if the index 1 is not touched by any of the previous $N-1$ many j values, the probability of which is $(\frac{N-1}{N})^{N-1}$. Since, $S^G_1[1] = S_N[S_N[1]]$, and we observed that $P(S_N[S_N[1]] = f_1) \approx (\frac{N-1}{N})^{2(N-1)}$, we have $w_{N+1} = P(S^G_N[i_{N+1}] = f_1) \approx (\frac{N-1}{N})^{3(N-1)}$. Now, using Lemma 2, we get $P(z_{N+1} = N + 1 - f_1) \approx \frac{1}{N} \cdot (1 + w_{N+1}) \approx \frac{1}{N} \cdot \left(1 + (\frac{N-1}{N})^{3(N-1)}\right)$. For $N = 256$, $w_{N+1} \approx 0.05$ and $P(z_{257} = 257 - f_1) \approx \frac{1}{N} \cdot (1 + 0.05) = 0.0041$ which also conforms to experimental observation.

2.2 Further Biases in RC4 Keystream: Experimental Results

Additionally, we observe some other significant biases that we could not prove theoretically so far. One may easily simulate the experiments to check our claims.

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 3: Additional bias of the keystream bytes towards the secret key.

The results are presented in Table 3 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. However, this is not true for RC4 keystream generation and experimental results show that $P(z_r = f_{r-1}) \geq \frac{1}{256}(1 + 0.05)$ for $1 \leq r \leq 21$ except $r = 2$. We are currently in the process of proving these biases theoretically.

2.3 Cryptanalytic Applications

Here we accumulate the theoretical and experimental results of the previous two sections. Consider the first keystream output byte z_1 of PRGA. We find the theoretical result that $P(z_1 = 1 - f_1) \approx 0.0053$ (see Theorem 1 and Table 1, 2) and the experimental observation that $P(z_1 = f_0) \approx 0.0043$ (see Table 3). Further, from [12], we have the result that $P(z_1 = f_2) \approx 0.0053$. Taking them together, one can check that the $P(z_1 = f_0 \vee z_1 = 1 - f_1 \vee z_1 = f_2) \approx 1 - (1 - 0.0043) \cdot (1 - 0.0053) \cdot (1 - 0.0053) \approx 0.0148$. Our result indicates that out of randomly chosen 10000 secret keys, in approximately 148 case, 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 $\alpha_1, \alpha_2, \alpha_3$ from $[0, \ldots, 255]$, then $P(z_1 = \alpha_1 \vee z_1 = \alpha_2 \vee z_1 = \alpha_3) = 1 - (1 - \frac{1}{256})^3 \approx 0.0117$. Thus one can guess z_1 with an additional advantage of $\frac{0.0148 - 0.0117}{0.0117} \cdot 100\% \approx 27\%$ over the random guess.

Now consider the keystream output byte z_2. We have $P(z_2 = 2 - f_2) = 0.0053$ (see Theorem 1 and Table 1, 2), which provides an advantage of $\frac{0.0053 - 0.0039}{0.0039} \cdot 100\% \approx 36\%$.

Similarly, referring to Table 1 and Table 3, 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.

References

[1] E. Biham and O. Dunkelman. Differential Cryptanalysis in Stream Ciphers. IACR Eprint Server, eprint.iacr.org, number 2007/218, June 6, 2007.

[2] 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.

[3] 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.

[4] J. Golic. Linear statistical weakness of alleged RC4 keystream generator. EUROCRYPT 1997, pages 226-238, vol. 1233, Lecture Notes in Computer Science, Springer-Verlag.

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

[6] 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].

[7] LAN/MAN Standard Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999 edition. IEEE standard 802.11, 1999.

[8] 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.

[9] 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.

[10] I. Mantin. Predicting and Distinguishing Attacks on RC4 Keystream Generator. EUROCRYPT 2005, pages 491-506, vol. 3494, Lecture Notes in Computer Science, Springer-Verlag.

[11] I. Mironov. (Not So) Random Shuffles of RC4. CRYPTO 2002, pages 304-319, vol. 2442, Lecture Notes in Computer Science, Springer-Verlag.

[12] 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.

[13] G. Paul and S. Maitra. Permutation after RC4 Key Scheduling Reveals the Secret Key. Accepted in SAC 2007. An extended version is available as "RC4 State Information at Any Stage Reveals the Secret Key" in IACR Eprint Server, eprint.iacr.org, number 2007/208, June 1, 2007.

[14] 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.

[15] 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.

[16] 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`.

[17] 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.