Discovery and Exploitation of New Biases in RC4

Pouyan Sepehrdad, Serge Vaudenay and Martin Vuagnoux

EPFL
CH–1015 Lausanne, Switzerland
`http://lasecwww.epfl.ch`

Abstract. In this paper, we present several weaknesses in the stream cipher RC4. First, we present a technique to automatically reveal linear correlations in the PRGA of RC4. With this method, 48 new exploitable correlations have been discovered. Then we bind these new biases in the PRGA with known KSA weaknesses to provide practical key recovery attacks. Henceforth, we apply a similar technique on RC4 as a black box, i.e. the secret key words as input and the keystream words as output. Our objective is to exhaustively find linear correlations between these elements. Thanks to this technique, 9 new exploitable correlations have been revealed. Finally, we exploit these weaknesses on RC4 to some practical examples, such as the WEP protocol. We show that these correlations lead to a key recovery attack on WEP with only 9 800 encrypted packets (less than 20 seconds), instead of 24 200 for the best previous attack.

1 Introduction

RC4 is a stream cipher designed by Ronald Rivest in 1987. It had initially been a trade secret until the algorithm was anonymously posted to the Cypherpunks mailing list in September 1994. Nowadays, RC4 is still widely used: it is the default cipher of the SSL/TLS protocol and a cryptographic primitive of the WEP (Wired Equivalent Privacy) and WPA (Wi-Fi Protected Access) protocols. Its popularity probably comes from its simplicity and the low computational cost of the encryption and decryption operations. Due to its straightforwardness, RC4 sparked extensive research, revealing weaknesses in case of misuse. The most famous example is the attack on the WEP protocol used in IEEE 802.11.

WEP was designed to provide confidentiality on wireless communications by using RC4. In order to simplify the key set up, WEP uses fixed keys. In wireless communications, packets may be easily lost. Because of the lack of transport control at the link level, IEEE 802.11 designers chose to encrypt each packet independently. However, RC4 is a stream cipher: the same secret key must never be used twice. To prevent any key repetition, WEP concatenates an initialization vector (IV) to the key, where the IV is a 24-bit value which is publicly disclosed in the header of the protocol. This particular use of RC4 is subject to many weaknesses. However, RC4 is not generally used with an IV and almost all the attacks concerning WEP cannot be applied to the plain RC4. Thus, RC4 is still believed to be secure, even if many weaknesses have been explored.

1.1 Description of RC4

The stream cipher RC4 consists of two algorithms: the Key Scheduling Algorithm (KSA) and the Pseudo Random Generator Algorithm (PRGA). The KSA generates an initial state from a random key K of ℓ words of n bits as described in Algorithm 1. It starts with an array $\{0,1,\ldots,N-1\}$, where $N = 2^n$ and swaps N pairs, depending on the value of the secret key K. At the end, we obtain the initial state S_{N-1}.

Algorithm 1 RC4 Key Scheduling Algorithm (KSA)

for $i = 0$ to $N-1$ **do**
    $S[i] \leftarrow i$
end for
$j \leftarrow 0$
for $i = 0$ to $N-1$ **do**
    $j \leftarrow j + S[i] + K[i \bmod \ell] \bmod N$
    swap($S[i]$,$S[j]$)
end for

Algorithm 2 RC4 Pseudo Random Generator Algorithm (PRGA)

$i \leftarrow 0$
$j \leftarrow 0$
loop
    $i \leftarrow i + 1 \bmod N$
    $j \leftarrow j + S[i] \bmod N$
    swap($S[i]$,$S[j]$)
    keystream word $z_i = S[S[i] + S[j] \bmod N]$
end loop

Once the initial state S_{N-1} is created, it is used by the second algorithm of RC4, the PRGA. Its role is to generate a keystream of words of n bits, which will be XORed with the plaintext to obtain the ciphertext. Thus, RC4 computes the loop of the PRGA each time a new keystream word z_i is needed, according to Algorithm 2. Note that each time a word of the keystream is generated the internal state of RC4 is updated.

Notation. In this paper, we define all the operators such as addition, subtraction and multiplication in the group $\mathbf{Z}/N\mathbf{Z}$. Thus, $x+y$ should be read as $(x+y) \bmod N$. The indices of the table respect the C-style programming reference. This means that the first entry of the table has the index 0. Let $S_i[k]$ denote the value of the array S at index k, after round i in KSA. S_{-1} denotes the array where $S[i] = i$, where $i = 0,1,\ldots,N-1$. Let $S_i^{-1}[p]$ be the index of the value p in the array S after round i in KSA. For example, $S_i^{-1}[S_i[k]] = k$ and $S_i[S_i^{-1}[p]] = p$. Let j_i (resp. j'_i) be the value of j during the round i

of KSA (resp. PRGA) where the rounds are indexed with respect to i. Thus, the KSA has rounds $0,1,\ldots,N-1$ and the PRGA has rounds $1,2,\ldots$. Let S'_i denote the array S after the i^{th} round of the PRGA (i.e. S'_1 is equal to S_{N-1} with $S_{N-1}[1]$ and $S_{N-1}[S_{N-1}[1]]$ swapped). We also denote $S_{N-1} = S'_0$. In this paper, RC4 is always used with $N = 256$. Thus, instead of words we may use bytes, which are equivalent. The keystream z_i is defined by

$$z_i = S'_i[S'_i[i] + S'_i[j'_i] \bmod N] = S'_i[S'_{i-1}[j'_i] + S'_{i-1}[i] \bmod N] \tag{1}$$

Let p be an integer and $\theta = e^{\frac{2i\pi}{p}}$. Unless otherwise mentioned, $\mathbf{G}$ denotes the $\mathbf{Z}_p$ group. The Discrete Fourier transform (DFT) of a function f over $\mathbf{G}^s$ is defined as

$$\hat{f}(c) = \sum_{x\in\mathbf{G}^s} f(x)\theta^{-c\bullet x}$$

where $\bullet$ is the dot product.

Previous Work. There are two approaches in the study of cryptanalysis of RC4: attacks based on the weaknesses of the KSA and attacks based on the weaknesses of the PRGA. Concerning the KSA, one of the first weakness published on RC4 was discovered by Roos [28] in 1995. This correlation binds the secret key bytes to the initial state S'_0. Roos [28] and Wagner [34] identified classes of weak keys which reveal the secret key if the first key bytes are known. This property has been largely exploited against WEP (see [5,9,2,15,16,32,30,29]). Another study of cryptanalysis of the KSA is the secret key recovery when the initial state S'_0 is known [25,1]. On the PRGA, The analysis has been largely motivated by distinguishing attacks [7,6,19,21] or initial state reconstruction from the keystream bytes [8,31,14,22] with complexity of 2^{241} for the best state recovery attack. Relevant studies of the PRGA reveal biases in the keystream output bytes in [20,27]. Mironov in [23] recommends that the first 512 initial keystream bytes must be discarded to avoid these weaknesses. Jenkins published in 1996 on his website [11] two biases in the PRGA of RC4. These biases have been generalized by Mantin in his Master Thesis [18] as *useful states*. Paul, Rathi and Maitra [26] discovered in 2008 a biased output index of the first keystream word generated by the PRGA. Ultimately, the last bias on the PRGA has been experimentally discovered by Maitra and Paul [17]. In practice, key recovery attacks on RC4 must bind KSA and PRGA weaknesses to correlate secret key words to keystream words. Some biases on the PRGA [13,26,17] have been successfully bound to the Roos correlation [28] to provide known plaintext attacks.

Our Contributions. Since, almost all known PRGA correlations have been experimentally found, we propose a method to exhaustively reveal new weaknesses in the PRGA. From 4 known biases in the PRGA, we have found 48 additional new exploitable correlations thanks to this technique. To provide key recovery attacks on RC4, we must bind KSA and PRGA weaknesses, we show that some of these new correlations can be bound to KSA vulnerabilities and lead to new key recovery attacks. Subsequently, we present another technique which does not consider the KSA and the PRGA, but only RC4 as a black box with the secret key words as input and the keystream words as output. Similar to the previous technique, we exhaustively search for correlations to rediscover the 3 known biases. Thanks to this technique, we have discovered 9 additional exploitable correlations between the secret key and the keystream words. Then, we exploit the known and new weaknesses against plain RC4 (with 48 first keystream words known by the attacker) and we obtain a key recovery attack with a complexity of $2^{122.06}$ instead of 2^{128} for RC4 with $N = 256$ and a key length of 16 bytes.

We also show that some of these correlations can be applied to WEP, decreasing the number of required encrypted packets to 9 800. The best previous key recovery attacks on WEP needed 24 200 encrypted packets [32,29] for the same success probability. This permits to recover a WEP key of 104 bits with a passive ciphertext only attack in less than 20 seconds in practice. This new attack is the best key recovery on WEP to our knowledge.

Structure of the Paper. In Section 2 we briefly explore known correlations in the PRGA of RC4. Section 3 details the technique used to visually represent correlations in the PRGA. Then, we describe the new biases discovered with our technique. In Section 4 we bind some of these new PRGA correlations with known KSA weaknesses to provide practical attacks on RC4. In Section 5 we detail another kind of exhaustive search where RC4 is a black box with the secret key words as input and the keystream words as output. Then, we present the new correlations found with this technique. Section 6 briefly describes a practical application of these biases to RC4 and RC4 with an IV such as used by WEP and WPA. Finally, we conclude.

2 Known Correlations in the PRGA of RC4

Jenkins Correlation. In 1996, Robert Jenkins described in his website [11] two biased correlations experimentally found on the PRGA of RC4. The first correlation considers the case where $S'_i[i] + S'_i[j'_i] = j'_i$. Thus, the i^{th} keystream byte given by Equation (1) becomes

$$z_i = S'_i[S'_i[i] + S'_i[j'_i]] = S'_i[j'_i] = j'_i - S'_i[i] \tag{2}$$

which holds with probability of $2/N$ instead of $1/N$ with $i = 1, 2, \ldots$. The second correlation appears when $S'_i[i] + S'_i[j'_i] = i$. In this case, the i^{th} byte of the keystream is given by

$$z_i = S'_i[S'_i[i] + S'_i[j'_i]] = S'_i[i] = i - S'_i[j'_i] \tag{3}$$

which holds with probability of $2/N$.

Mantin and Shamir Correlation. Mantin and Shamir [20] discovered a biased distribution for the second keystream word.

Theorem 1. *Assume an initial state S'_0 is chosen randomly. The probability that the second keystream word z_2 of RC4 is 0 is approximately $2/N$ instead of $1/N$.*

Paul, Rathi and Maitra Correlation. In 2008, Paul, Rathi and Maitra [26] described a biased correlation between the three first words of the secret key and the first keystream word z_1 of RC4.

Theorem 2. *Assume that the initial state S'_0 is chosen uniformly at random from the set of all possible permutations of the set $\{0,1,\ldots,N-1\}$. Then the probability distribution of the output index $S'_1[1]+S'_1[S'_0[1]] = S'^{-1}_1[z_1]$ that selects the first byte of the keystream output is given by*

$$P\left(S'_1[1]+S'_1[j'_i]=x\right) = \begin{cases} \frac{1}{N} & \text{for odd } x \\ \frac{1}{N} - \frac{2}{N(N-1)} & \text{for even } x \neq 2 \\ \frac{2}{N} - \frac{1}{N(N-1)} & \text{for even } x = 2 \end{cases}$$

3 Visual Representation of Correlations in the PRGA

In general, the methods used to find correlations in RC4 are either opportunistic or not given. Papers tend to describe the characteristics of the biases without revealing the techniques used to discover them. We propose to describe some simple but efficient techniques to highlight weaknesses in the PRGA through exhaustive search on a subset of elements.

We define a set of linear equations which contains all the known biased correlations of the PRGA described in the previous section. Our objective is to highlight linear correlations between the internal values of a round of the PRGA and the keystream word generated by this round i.e. the subset of elements $\{i, j'_i, S'_i[i], S'_i[j'_i], z_i\}$. The correlations previously discovered by Jenkins, Mantin and Shamir and Paul, Rathi and Maitra must be rediscovered with this method. Surprisingly, some new biases are found as well. We define the linear equations as

$$(a_0 \cdot i + a_1 \cdot j'_i + a_2 \cdot S'_i[i] + a_3 \cdot S'_i[j'_i] + a_4 \cdot z_i) \bmod N = b \tag{4}$$

where the a_i's are elements of $\mathbf{Z}/256\mathbf{Z}$ and b is a fixed value in $\mathbf{Z}/256\mathbf{Z}$. This defines 2^{48} linear equations. To reduce this number, we decompose these equations into 256 subgroups. Each of them corresponds to a specific round (i.e. i is fixed). Thus, both $a_0 \cdot i$ and b can be merged into one value and Equation (4) becomes

$$(c_0 \cdot j'_i + c_1 \cdot S'_i[i] + c_2 \cdot S'_i[j'_i] + c_3 \cdot z_i) \bmod N = C \tag{5}$$

where $C = (b - a_0 \cdot i) \bmod 256$ and c_i's are elements of $\mathbf{Z}/256\mathbf{Z}$. Since the number of linear equations is still too large, we limit the coefficients set of the c_i's to $\{-1,0,1\}$. Indeed, this set is enough to include all the previously known biased correlation in the PRGA. We obtain 256 graphs of 81 linear equations. We compute 256 first rounds of the PRGA with 10^9 randomly chosen RC4 secret keys of 16 bytes and we verify all the linear equations described by Equation 5. For every equation, a counter is incremented when it holds. Subsequently, we represent these counters as a graph to visually illustrate potential biases. Human brains are very efficient to visually detect anomalies in uniform

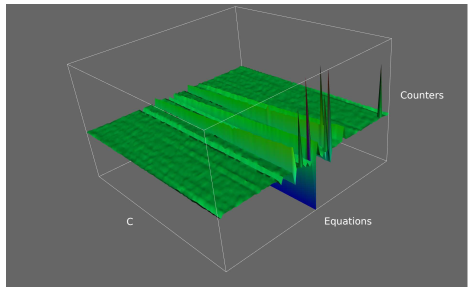

Fig. 1: 3D representation of the equations of the second round of the PRGA according to C and the counters of the equations. The objective of this graph is to visually detect biased.

distributions (see Figure 1). Below we give the biased correlations found for the 256 first keystream bytes (from z_1 to z_{256}) generated by the PRGA. Every coefficient c_i has been replaced by the corresponding element to provide an easier reading of the table (i.e. j_i' must be read as c_0, $S_i'[i]$ as c_1, etc.). Correlations with z_i (i.e. $c_3 \neq 0$) are called New_XXX and biases without z_i (i.e. $c_3 = 0$) are named New_noz_XXX.

In Figure 2, we confirm the presence of known biases such as the Jenkins correlations. More interestingly, new biases in the PRGA appear. Some of them have a probability of success which depends on the value of i.

Some rounds of the PRGA provide additional biased correlations. In Figure 3, we give extra biases which appear in round 1. Figure 4 depicts the additional correlations in the second round of the PRGA. Finally, Figure 5 describes further biased correlations in rounds 0 mod 16 of the PRGA. The probability of the biased correlations New_001 and New_002 depends on the round of the PRGA and the value C. In Figure 6 we show the success probability of New_001 according to C and the rounds 1, 16, 32, 64, 128, 192 and 256 of the PRGA. Similarly, we compute the evolution of the success probability of the biased correlations New_002 in Figure 8. Figure 7 gives the 3D representation of the same bias. Figure 9 depicts the same bias for all the first 256 rounds of the PRGA using a 3D representation.

3.1 Spectral Approach to Derive New Biases

Another systematic method to derive linear relations is to use Fourier transform of the type f of the distribution that some state bits of RC4 is following. We can use exactly the same approach to derive a linear relation between the main key bits and the key stream. Deploying this method over $\mathbf{Z}_{256}^{s}$, we can derive some *good* linear relations. We call a linear relation good if the probability of Equation (4) occurring is much higher than expected ($\gg \frac{1}{N}$). We use the 4-tuple defined above as an example. In fact, we can use exactly the same method to derive biases for linear relations in Section 5.2 between the secret key and keystream. To deal with this problem, we assume $\mathbf{G} = \mathbf{Z}_{256}$ and we query RC4 for $\mathcal{N}$ vectors $\mathcal{V}_t \in (j_i', S_i'[i], S_i'[j_i'], z_i)_t \in \mathbf{G}^4$ for $1 \leq t \leq \mathcal{N}$ and we define a

j'_i	$S'_i[i]$	$S'_i[j'_i]$	z_i	C	Probability	Remark
1	-1	0	-1	0	$2/N$	Jenkins Equation (2)
0	0	1	1	i	$2/N$	Jenkins Equation (3)
0	1	1	-1	0	$1.9/N$	New_000
0	1	1	-1	1	$0.89/N$	New_001
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	1	1	-1	255	$1.25/N$	New_001
0	1	1	1	0	$0.95/N$	New_002
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	1	1	1	255	$0.95/N$	New_002
1	1	0	0	0	$0.95/N$	New_noz_000
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	0	0	255	$0.95/N$	New_noz_000
1	1	-1	0	i	$2/N$	New_noz_001
1	-1	1	0	i	$2/N$	New_noz_002
1	-1	0	0	1	$0.9/N$	New_noz_003
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	-1	0	0	255	$1.25/N$	New_noz_003
1	-1	0	0	0	$1.9/N$	New_noz_004
0	0	1	0	$i+1$	$1.36/N$	New_noz_005
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	0	1	0	255	$0.9/N$	New_noz_005
0	0	1	0	i	$2.34/N$	New_noz_006

Fig. 2: correlations experimentally observed for rounds $1,2,3,\ldots,256$ of the PRGA. Note that probability of biases New_000, New_noz_005 and New_noz_006 decrease according to i. The probabilities given in this table correspond to round 3 (i.e. $i=3$). New_noz_004 and New_noz_006 are not biased when $i=1$.

j'_i	$S'_i[i]$	$S'_i[j'_i]$	z_i	C	Probability	Remark
0	1	0	-1	0	$0.95/N$	New_003
0	1	1	0	2	$1.95/N$	Paul, Rathi and Maitra
1	1	0	0	2	$1.94/N$	New_noz_014

Fig. 3: Additional biased correlations experimentally observed using Equation (5) in the first round of the PRGA (i.e. $i=1$).

function f as follows.

$$f(x) = \text{counter}(x) = \frac{1}{\mathcal{N}} \sum_{t=1}^{\mathcal{N}} \mathbf{1}_{\mathcal{V}_t = x}$$

j'_i	$S'_i[i]$	$S'_i[j'_i]$	z_i	C	Probability	Remark
0	0	0	1	0	$2/N$	Mantin and Shamir [20]
1	-1	1	-1	0	$2/N$	New_004
1	1	0	-1	even	$1.0183/N$	New_005
1	1	0	-1	odd	$1.0316/N$	New_006
1	0	1	0	6	$2.37/N$	New_noz_007
1	0	-1	0	255	$0.75/N$	New_noz_008
1	-1	1	0	0	$2/N$	New_noz_009
0	-1	1	0	0	$0.95/N$	New_noz_010

Fig. 4: Additional biased correlations experimentally observed using Equation (5) in the second round of the PRGA ($i = 2$). Note that the probability of success of correlations New_005 and New_006 decreases according to the value C.

j'_i	$S'_i[i]$	$S'_i[j'_i]$	z_i	C	Probability	Remark
0	0	0	1	-i	$1.0411/N$	New_007
0	0	1	-1	i	$1.0500/N$	New_008
0	0	1	1	-i	$1.0338/N$	New_009
0	1	1	0	-i	$1.1107/N$	New_noz_011
0	1	0	0	-i	$1.1276/N$	New_noz_012
0	1	-1	0	-i	$1.1067/N$	New_noz_013

Fig. 5: Additional biased correlations experimentally observed using Equation (5) in rounds 0 mod 16 of the PRGA. Note that the probability of success of these correlations decreases according to the value i and become barely exploitable when $i > 48$. Probabilities given in this table come from round 16.

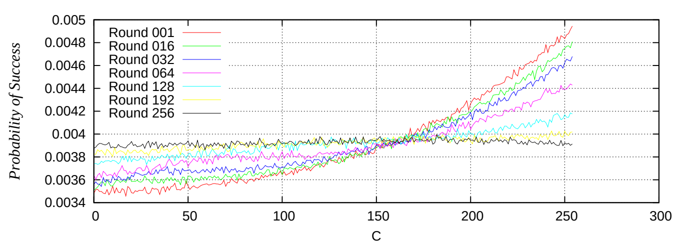

Fig. 6: Probability of success of biased correlation New_001, according the value C for some rounds of the PRGA.

where $x \in \mathbf{G}^4$. In fact, f is the *type* of the distribution the 4-tuples are following. Deploying DFT on f, we have

$$\hat{f}(c) = \sum_x \theta^{-c.x} f(x) = \sum_C \sum_{x:c.x=C} \theta^{-C} f(x) = \sum_C \theta^{-C}.\ \Pr[c\ .\ v = C]$$

Then, we follow this approach: we compute $|\hat{f}(c)|^2$ for c's such that this norm is high. Filtering those c's yields "good" c's.

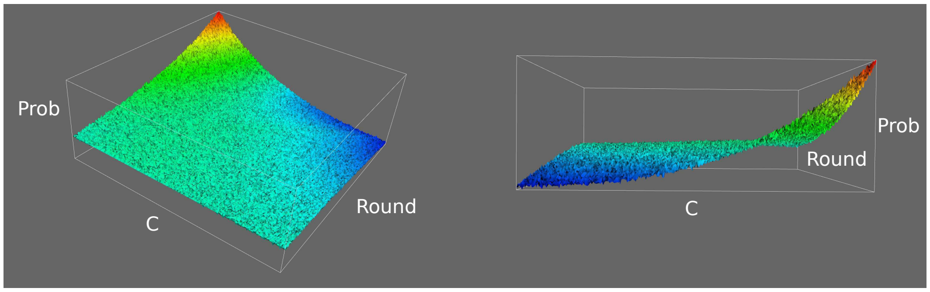

Fig. 7: Probability of biased correlation New_001, according the value C for the first 256 rounds of the PRGA, using a 3D representation.

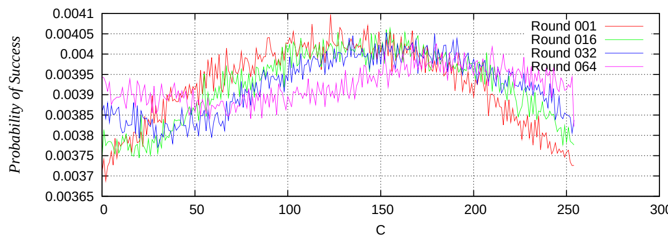

Fig. 8: Probability of success of biased correlation New_002, according the value C for some rounds of the PRGA.

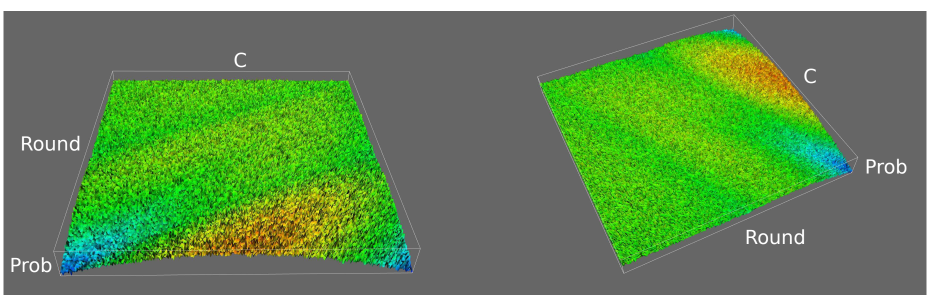

Fig. 9: Probability of biased correlation New_002, according the value C for the first 256 rounds of the PRGA, using a 3D representation.

We construct the table $|\hat{f}(c)|^2$ of all linear masks and filter out c's that leads to small value for $|\hat{f}(c)|^2$. This remains us some c's. Then, we can exhaustively search in the c's left and find the good $c's$. This method is much faster than exhaustive search. Assume we consider the linear relation between the elements of the vector $(j'_i, S'_i[j'_i], z_i)$ instead of $(j'_i, S'_i[i], S'_i[j'_i], z_i)$. This already gives us all biases quite fast. This method can be easily generalized to the case when $S'_i[i]$ is also involved and it is dramatically faster than exhaustive search.

The complexity of this method for the triplet case is 3.2^{32}, while for exhaustive search the complexity reaches to 2^{47} if $N = 10^7$, which is a reasonable number of samples we found out experimentally. This will gain us all the biases for the triplet over $\mathbf{Z}_{256}^3$. Using this method, we found out some new biases. We only list those which are not artifact of known biases and which can be bound with biases of KSA. They are listed in Fig 10. Any time the coefficient of $S'_i[j'_i]$ is one in the table, we can use that equation to bind it like what would be explained in the next section for binding New_008 and New_009 biases. This method removes the restriction on coefficients to be only in the set $\{-1, 0, 1\}$. Using these technique, we can recover all biases in $\mathbf{Z}_{256}^4$ in a reasonable time.

j'_i	$S'_i[i]$	$S'_i[j'_i]$	z_i	C	i	Probability	Remark
0	0	1	1	240	16	$1.04/N$	New_010
0	0	1	50	224	16	$1.04/N$	New_011
0	0	1	68	192	16	$1.05/N$	New_012
0	0	1	98	224	16	$1.04/N$	New_013
0	0	1	148	192	16	$1.05/N$	New_014
0	0	1	162	224	16	$1.05/N$	New_015
0	0	1	186	96	16	$1.03/N$	New_016
0	0	1	187	80	16	$1.04/N$	New_017
0	0	1	251	80	16	$1.04/N$	New_018
0	0	2	19	208	16	$1.04/N$	New_019
0	0	2	127	16	16	$1.04/N$	New_020
0	0	2	147	208	16	$1.04/N$	New_021
0	0	2	255	16	16	$1.04/N$	New_022
0	0	4	59	80	16	$1.04/N$	New_023
0	0	4	123	80	16	$1.04/N$	New_024
0	0	8	19	208	16	$1.04/N$	New_025
0	0	8	55	144	16	$1.03/N$	New_026
0	0	8	81	240	16	$1.03/N$	New_027
0	0	8	215	144	16	$1.03/N$	New_028
0	0	8	241	48	16	$1.03/N$	New_029
0	0	8	243	208	16	$1.04/N$	New_030
0	0	32	39	144	16	$1.04/N$	New_031
0	0	32	191	16	16	$1.04/N$	New_032

Fig. 10: correlations in PRGA derived using DFT. If coefficient of $S'_i[j'_i]$ is one, that equation can be bound with KSA biases.

After investigation, it seems that all the listed biases are artifact of a new conditional bias which is

$$\Pr[S'_{16}[j'_{16}] = 0 | z_{16} = -16] = 0.038488$$

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

4 Binding PRGA and KSA Weaknesses

Since we have new PRGA biased correlations, we have to bind them with KSA weaknesses to provide key recovery attacks.

4.1 Known Binding between KSA and PRGA Weaknesses

Already known bindings between KSA and PRGA have been exploited. In 2006, Klein [12,13] demonstrated that the Jenkins correlation of the PRGA and some weaknesses in the KSA can be combined.

$$
\begin{aligned}
&S'_i[j'_i] \stackrel{P_j}{=} i - z_i && \text{from Equation (3) with } P_j = 2/N && (6)\\
&S'_i[j'_i] = S'_{i-1}[i] && \text{step 6 of the PRGA} && (7)\\
&S'_{i-1}[i] \stackrel{P'}{=} S_i[i] && P' = ((N-1)/N)^{N-2} && (8)\\
&S_i[i] = S_{i-1}[j_i] && \text{KSA} && (9)\\
&j_i = S_{i-1}[i] + j_{i-1} + K[i] && \text{step 6 of the KSA} && (10)
\end{aligned}
$$

From (6) with respectively (7), (8), (9) and (10) we have

$$K[i] \stackrel{P_{\mathsf{Klein}}}{=} S_{i-1}^{-1}[i - z_i \bmod N] - S_{i-1}[i] - j_{i-1} \bmod N \tag{11}$$

which holds with probability

$$P_{\mathsf{Klein}} = \frac{2}{N} \cdot \left(\frac{N-1}{N}\right)^{N-2} + \frac{N-2}{N(N-1)} \cdot \left(1 - \left(\frac{N-1}{N}\right)^{N-2}\right) \approx \frac{1.36}{N} \tag{12}$$

In 2007, Vaudenay and Vuagnoux [32] improved this attack by using the Roos correlation and the repetition of the secret key modulo ℓ. Thus, the sum of the secret key bytes can be recovered with

$$\sum_{y=0}^{i} K[y] \stackrel{P_{\mathsf{KleinImproved}}}{=} i - z_i - \frac{i \cdot (i+1)}{2} \bmod N \tag{13}$$

with success probability of $P_{\mathsf{KleinImproved}}(i)$ defined by

$$
\begin{aligned}
&P_{\mathsf{C}}(i) = \left(\tfrac{N-1}{N}\right)^i \cdot \textstyle\prod_{k=1}^{i} \left(\tfrac{N-k}{N}\right) \cdot \left(\tfrac{N-1}{N}\right)^{N-2}\\
&P_{\mathsf{KleinImproved}}(i) = \tfrac{2}{N} \cdot P_{\mathsf{C}}(i) + \tfrac{N-2}{N(N-1)} \cdot (1 - P_{\mathsf{C}}(i))
\end{aligned} \tag{14}
$$

for any $i - z_i \mod N \neq \{0, 1, \ldots, i-1\}$.

4.2 Binding New PRGA Bias where $S'_i[j'_i]$ is Involved

Interestingly, from the Jenkins correlation described by $S'_i[j'_i] = i - z_i$ and Equation (13) we have

$$S'_i[j'_i] \overset{P_{\mathsf{KleinImproved}}}{=} \frac{i\cdot(i+1)}{2} + \sum_{y=0}^{i} K[y] \bmod N \tag{15}$$

for the same success probability. Hence, every new biased equation containing $S'_i[j'_i]$ and public values such as i and z_i can be exploited as key recovery attack.

New_009 ($-i - S_i[j'_i] = z_i$). This biased correlation concerns rounds 0 mod 16 of the PRGA. The success probability is $1.0338/N$ for round 16. Using the same technique described above, we can exploit this bias to recover the secret key sum

$$\sum_{y=0}^{i} K[y] \overset{P_0}{=} -i - z_i - \frac{i\cdot(i+1)}{2} \bmod N$$

with a success probability equal to

$$P_0(i) = \frac{1.0338}{N}\cdot P_{\mathsf{C}}(i) + \frac{N-1.0338}{N(N-1)}\cdot(1-P_{\mathsf{C}}(i))$$

for any $-i - z_i \mod N \neq \{0,1,\ldots,i-1\}$ and $i = 16$.

New_008 ($-i + S_i[j'_i] = z_i$). Similarly, we can exploit this biased equation using the same technique

$$\sum_{y=0}^{i} K[y] \overset{P_1}{=} -i + z_i - \frac{i\cdot(i+1)}{2} \bmod N$$

with a success probability equal to

$$P_1(i) = \frac{1.05}{N}\cdot P_{\mathsf{C}}(i) + \frac{N-1.05}{N(N-1)}\cdot(1-P_{\mathsf{C}}(i))$$

for any $-i + z_i \mod N \neq \{0,1,\ldots,i-1\}$ and $i = 16$.

4.3 Exploiting Additional Biased Linear Correlations in the PRGA

The binding between KSA weakness and PRGA biases presented above is an example of a practical application of the correlations discovered in the previous section. We did not find a way to exploit biases where $S'_i[i]$ or j'_i are involved. However, these correlations could be exploited in future work.

5 RC4 as a Black Box

In Section 4 we have seen that secret key words and keystream words may be correlated if weaknesses in the KSA and PRGA can be bound. However, from an attacker's point of view there is no reason to determine weaknesses in the KSA or the PRGA. The objective is to find correlations between known values and the secret key words. Moreover, the biased correlations previously found concern only elements inside a round of the PRGA. Correlations between elements from different rounds cannot be highlighted.

In this section, we present another way to attack RC4. We consider RC4 as a black box. The objective is to discover linear correlations between the input (the secret key words) and the output (the keystream words), since we consider known keystream attacks. First, we study known RC4 correlations between the keystream words and the secret key words. Then, we propose a method to highlight new biases. Finally, we list new discovered biases in RC4.

5.1 Maitra and Paul Correlation

In 2008, Maitra and Paul [17] discovered a new experimental observation on RC4 which holds with probability of $\approx 1.10/N$ for $i = 0$.

$$z_{i+1} \overset{P_{\mathsf{Maitra}}}{=} \frac{i \cdot (i+1)}{2} + \sum_{y=0}^{i} K[y] \tag{16}$$

Maitra and Paul did not find any practical application for this bias in protocols using or based on RC4. However, from Equation 15 we can rewrite Equation (16) as

$$z_{i+1} \overset{P_{\mathsf{Maitra}}}{=} S'_i[j'_i]$$

This bias has not been found with our previous technique, since these elements are not in the same round of the PRGA. However, with the black box technique we are able to rediscover this bias and new ones.

5.2 Discovering New Linear Correlations in RC4

We define a linear equation containing input and output elements.

$$(a_0 \cdot K[0] + \ldots + a_{\ell-1} \cdot K[\ell-1] + a_\ell \cdot z_1 + \ldots + a_{N+\ell-1} \cdot z_N) \bmod N = b \tag{17}$$

This kind of exhaustive search is identical to those presented in Section 3. First, we consider the subset of all a_i's defined by $A = \{-1, 0, 1\}$ and $b \in \mathbf{Z}/N\mathbf{Z}$. The number of equation is $N \cdot 3^{\ell+N} = 2^{439.11}$ for $N = 256$ and $\ell = 16$, which is obviously too large for an exhaustive search.

Based on the pseudo T-function's behavior of the KSA (i.e. $S_{N-1}[i]$ depends only on $K[i-1], K[i-2], \ldots, K[0]$ with a non negligible probability) and the Roos correlation, we can reduce the size of the equation by considering only the first ℓ keystream words. Equation (17) becomes

$$(a_0 \cdot K[0] + \ldots + a_{\ell-1} \cdot K[\ell-1] + a_\ell \cdot z_1 + \ldots + a_{2\ell-1} \cdot z_\ell) \bmod N = b \tag{18}$$

Thus, we obtain a number of equations equals to $N \cdot 3^{2\ell} = 2^{58.7}$ which is still too large for an exhaustive search. Thus, we reduced the secret keys length to $\ell = 5$ bytes to obtain $2^{23.8496}$ equations. Indeed, based on the pseudo T-function behavior's of the KSA, we suppose that the correlations found with a RC4 key length of 5 bytes can be generalized to RC4 with a secret key of 16 bytes. Then, the supposed biased correlation are tested experimentally. After a few computation, we remarked that the constant value represented by b can be reduced to the subset generated by $i \cdot (i+1)/2$ with $i = 0, 1, 2 \ldots, 22$, since only the Roos correlation seems to be exploited in the KSA. Thus, the number of equations decreases to $23 \cdot 3^{10} = 2^{20.373}$. Figure 11 gives the correlations found in RC4 with a secret key of 5 bytes which are experimentally confirmed on RC4 with a key length of 16 bytes.

Equation	Probability	Remarks
$z_1 + K[0] + K[1] = 0$	1.35779/N	Klein Improved
$z_1 - K[0] = 0$	1.11784/N	Maitra and Paul
$z_2 = 0$	2.01825/N	Mantin and Shamir
$z_2 + K[0] + K[1] + K[2] = -1$	1.36095/N	Klein Improved
$z_1 - K[0] - K[1] = 1$	1.04237/N	New_bb_000
$z_1 - K[0] + K[1] = -1$	1.04969/N	New_bb_001
$z_3 + K[0] + K[1] + K[2] + K[3] = -3$	1.35362/N	Klein Improved
$z_3 - K[0] + K[3] = -3$	1.04620/N	New_bb_002
$z_1 - K[0] - K[1] - K[2] = 3$	1.33474/N	Roos/Paul et al.
$z_2 - K[0] - K[1] - K[2] = 3$	0.64300/N	New_bb_003
$z_3 - K[0] - K[1] - K[2] = 3$	1.13555/N	Maitra and Paul
$z_2 + K[1] + K[2] = -3$	1.36897/N	New_bb_004
$z_2 - K[1] - K[2] = 3$	1.36733/N	New_bb_005
$z_1 - K[2] = 3$	1.14193/N	New_bb_006
$z_1 + K[0] + K[1] - K[2] = 3$	1.14116/N	New_bb_007
$z_4 - K[0] + K[4] = 4$	1.04463/N	New_bb_008
$z_4 + K[0] + K[1] + K[2] + K[3] + K[4] = -6$	1.35275/N	Klein Improved
$z_4 - K[0] - K[1] - K[2] - K[3] = 10$	1.11432/N	Maitra and Paul

Fig. 11: Biased correlations experimentally observed with the black box technique with $\ell = 5$ in Equation 18. Note that these biases are exploitable in RC4 with a secret key of 16 bytes as well.

For every index i corresponding to the index of a key byte, let d_i the number of biased equations we have for $\bar{K}[i] = K[0] + \cdots + K[i]$. Let $p_{i,j}$ be the probability of the j_{th} equation for this byte. The list of biases we use is depicted in Fig 12.

If the key has size ℓ, indices $i = k.\ell - 1$ correspond to the same byte, so we can merge the associated list of biases. Similarly, if $\bar{K}[\ell - 1]$ is known, indices which are equal modulo ℓ correspond to the same byte, so we can merge these lists as well. Let $p'_{i,j}$ be the table of merged lists and d'_i be the length of list $p'_{i,j}$.

Equation	Probability	Remarks
$\bar{K}[0] - z_1 = 0$	1.10873/N	Maitra and Paul
$\bar{K}[1] + z_1 = 0$	1.36467/N	Klein Improved
$\bar{K}[1] - z_1 = 255$	1.04237/N	New_bb_000
$\bar{K}[2] - z_2 = 253$	0.64300/N	New_bb_003
$\bar{K}[2] + z_2 = 255$	1.36036/N	Klein Improved
$\bar{K}[2] - z_3 = 253$	1.12742/N	Maitra and Paul
$\vdots$	$\vdots$	$\vdots$
$\bar{K}[14] + z_{14} = 165$	1.22758/N	Klein Improved
$\bar{K}[14] - z_{15} = 151$	1.06444/N	Maitra and Paul
$\bar{K}[15] + z_{15} = 151$	1.21317/N	Klein Improved
$\bar{K}[15] - z_{16} = 136$	1.07519/N	Maitra and Paul
$\bar{K}[15] + \bar{K}[0] - z_{16} = 104$	1.01838/N	New_008
$\bar{K}[15] + \bar{K}[0] + z_{16} = 104$	1.01242/N	New_009
$\bar{K}[15] + \bar{K}[0] + z_{16} = 136$	1.19880/N	Klein Improved
$\bar{K}[15] + \bar{K}[0] - z_{17} = 136$	1.05983/N	Maitra and Paul
$\vdots$	$\vdots$	$\vdots$
$\bar{K}[15] + \bar{K}[14] + z_{30} = 77$	1.04582/N	Klein Improved
$\bar{K}[15] + \bar{K}[14] - z_{31} = 47$	1.02118/N	Maitra and Paul
$2\bar{K}[15] + z_{31} = 47$	1.03963/N	Klein Improved
$2\bar{K}[15] - z_{32} = 16$	1.03833/N	Maitra and Paul
$2\bar{K}[15] + \bar{K}[0] - z_{32} = 208$	1.009/N	New_008
$2\bar{K}[15] + \bar{K}[0] + z_{32} = 208$	1.0062/N	New_009
$2\bar{K}[15] + \bar{K}[0] + z_{32} = 16$	1.03403/N	Klein Improved
$\vdots$	$\vdots$	$\vdots$
$2\bar{K}[15] + \bar{K}[14] + z_{46} = 57$	1.00027/N	Klein Improved
$3\bar{K}[15] + z_{47} = 199$	0.99951/N	Klein Improved

Fig. 12: Useful biases in key recovery attack on RC4 for $\ell = 16$.

6 Key Recovery Attacks

6.1 Theoretical Key Recovery Attack on Plain RC4

We consider RC4 with $N = 256$ and a secret key length $\ell = 16$ bytes. Thus, the exhaustive search has complexity of 2^{128}. Using all the exploitable biased correlations presented in this paper and the previously known biases in RC4 and considering that they are independent, we are able to recover the RC4 secret key words from the 48 first keystream words (known keystream attack) with a complexity of $2^{122.06}$. In fact, the probability that all key bytes are expressed by at least one bias is

$$p \approx \prod_{i=0}^{\ell-1} \left(1 - \prod_{j=1}^{d_i'} (1 - p_{i,j}') \right)$$

and the number of combination of biases is $k = \prod_{i=0}^{\ell-1} d'_i$. So the average complexity is $\frac{k}{2}$ with success probability p. The average complexity by iterating is $\frac{k}{p}$. With our table, we compute $p \approx 2^{-87.90}$ and $k \approx 2^{38.09}$. Thus, the complexity of attacking plain RC4 would be $2^{125.99}$.

Note that p is optimized this way, but not $\frac{k}{p}$. By taking only the largest bias for each byte, we obtain $\frac{k}{p} = 2^{122.06}$ with $k = 1$.

6.2 Practical Key Recovery Attack on WEP

To provide a practical use of these attacks, we tried to deploy them on RC4 with an IV such as used by the protocol WEP or WPA. For some people, attacking WEP is like beating a dead horse, since it has been already badly broken [5,9,2,15,16,32,30,29]. First, the new biases presented in this paper are related to the stream cipher RC4. WEP is an example of a practical exploitation of these biases. Moreover, the cryptanalysis of WEP is one of the most applied cryptographic attack in practice. Indeed, tools such as *Aircrack* [4] are massively downloaded to provide a good example of weaknesses in cryptography.

In the case of WEP and WPA, the first 3 bytes are known and the key is a repetition of $\ell = 16$ bytes. So, we can remove the $P_{0,j}, P_{1,j}, P_{2,j}$ lists, redo the merge operation, then merge the lists for $i' = 3, 4, \ldots, 15$. In practice, this gives $k \approx 2^{31.77}$ biases and results in $p = 2^{-70.57}$. So, we have a complexity of $\frac{k}{p} \approx 2^{102.34}$ to recover the full key K with a single packet.

We picked the same key recovery algorithm described in [32], but we added known and new correlation presented in this paper. We also include all conditional biases from the Korek attacks [15,16,3] with the improvement described in [32,33]. The complexity of the key recovery attacks on WEP depends on the number of encrypted packets captured. For every captured packet, we sort the list of potential secret keys given by the key recovery attacks and we test the first 10^6 keys according to this list. See [32] for more details. The previous best key recovery attack described in [32] and better implemented in [29] needs 24 200 packets to recover a secret key of 104 bits with a success probability $> 1/2$. With the new key recovery described in this paper, we are able to recover the secret key with an average of 9 800 encrypted packets for the same success probability. This complexity of 9 800 packets was measured experimentally by running the attacks on 10^6 random secret keys. This represents the best key recovery attack on WEP to our knowledge.

6.3 Theoretical Key Recovery Attacks on WPA

In order to correct the weaknesses on WEP discovered before 2004, the Wi-Fi Alliance proposed a WEP improved protocol called WPA [10]. It has been established that WPA must be hardware compatible with existing WEP capable devices to be deployed as a software patch. Basically, WPA is a WEP wrapper which contains anti-replay protections and a key management scheme to avoid key reuse. In 2004, Moen, Raddum and Hole [24] discovered that the recovery of at least two RC4 packet keys in WPA leads to a full recovery of the temporal key and the message integrity check key. The complexity of this attack is defined by the exhaustive search of two 104-bit long keys, i.e. 2^{104}.

Almost all known and new key recovery attacks on WEP can be applied to WPA. Only the Fluhrer, Mantin and Shamir attack [5] is filtered. Indeed, WPA encryption is similar to WEP, using RC4 with an IV. But, WPA uses a different secret key for every encrypted packet. Once from the same segment of 2^{16} consequetive packets, two keys are successfully recovered, the Moen, Raddum and Hole attack can be applied. However, the attack in Section 6.2 has a success probability p too low to recover two keys. The actual application to attacking WPA is left for future work.

7 Conclusion

In this paper, we have seen some techniques to exhaustively highlight linear correlations in RC4. First, we have considered only the elements inside a round of the PRGA. Then, we have generalized this method to the whole RC4 as a black box with the secret key words as input and the keystream words as output. These techniques led to the discovery of 57 new correlations in RC4. Some of them can be directly applied to existing key recovery attacks on RC4, WEP and WPA. For example, a WEP secret key of 128 bits (104 unknown bits) can be recovered in less than 20 seconds, the time to eavesdrop at least 9 800 encrypted packets. This is the best attack on WEP to our knowledge.

However, the main interest of this paper is the application of an automated discovery of weaknesses in ciphers. Similar to fuzzing techniques used to highlight security vulnerabilities in computer systems, these methods, although relatively simple, reveal an impressive number of new weaknesses in a intensively analyzed stream cipher such as RC4. This may suggest a new kind of automated tools for cryptanalysts. Indeed, weaknesses in network protocol or computer systems are largely found by automated tools such as fuzzers, negative testers or black box analyzers. With the results presented in this paper, it may be interesting to adapt these tools for cryptanalysis.

References

1. Eli Biham and Yaniv Carmeli. Efficient Reconstruction of RC4 Keys from Internal States. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 270–288. Springer, 2008.
2. Andrea Bittau. Additional Weak IV Classes for the FMS Attack, 2003. `http://www.cs.ucl.ac.uk/staff/a.bittau/sorwep.txt`.
3. Rafik Chaabouni. *Breaking WEP Faster with Statistical Analysis*. Ecole Polytechnique Fédérale de Lausanne, LASEC, Semester Project, 2006.
4. Christophe Devine and Thomas Otreppe. Aircrack. `http://www.aircrack-ng.org/`.
5. Scott Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In Serge Vaudenay and Amr M. Youssef, editors, *Selected Areas in Cryptography*, volume 2259 of *Lecture Notes in Computer Science*, pages 1–24. Springer, 2001.
6. Scott R. Fluhrer and David A. McGrew. Statistical Analysis of the Alleged RC4 Keystream Generator. In Bruce Schneier, editor, *FSE*, volume 1978 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 2000.

7. Jovan Dj. Golic. Linear Statistical Weakness of Alleged RC4 Keystream Generator. In *EUROCRYPT*, pages 226–238, 1997.
8. Jovan Dj. Golic. Iterative Probabilistic Cryptanalysis of RC4 Keystream Generator. In Ed Dawson, Andrew Clark, and Colin Boyd, editors, *ACISP*, volume 1841 of *Lecture Notes in Computer Science*, pages 220–233. Springer, 2000.
9. David Hulton. Practical Exploitation of RC4 Weaknesses in WEP Environments, 2001. `http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt`.
10. IEEE. ANSI/IEEE standard 802.11i: Amendment 6 Wireless LAN Medium Access Control (MAC) and Physical Layer (phy) Specifications, Draft 3, 2003.
11. Robert Jenkins. ISAAC and RC4. `http://burtleburtle.net/bob/rand/isaac.html`.
12. Andreas Klein. Attacks on the RC4 Stream Cipher. Personal Andreas Klein website, 2006. `http://cage.ugent.be/~klein/RC4/RC4-en.ps`.
13. Andreas Klein. Attacks on the RC4 Stream Cipher. *Des. Codes Cryptography*, 48(3):269–286, 2008.
14. Lars R. Knudsen, Willi Meier, Bart Preneel, Vincent Rijmen, and Sven Verdoolaege. Analysis Methods for (Alleged) RC4. In Kazuo Ohta and Dingyi Pei, editors, *ASIACRYPT*, volume 1514 of *Lecture Notes in Computer Science*, pages 327–341. Springer, 1998.
15. KoreK. Need Security Pointers, 2004. `http://www.netstumbler.org/showthread.php?postid=89036#post89036x`.
16. KoreK. Next Generation of WEP Attacks?, 2004. `http://www.netstumbler.org/showpost.php?p=93942&postcount=35`.
17. Subhamoy Maitra and Goutam Paul. New Form of Permutation Bias and Secret Key Leakage in Keystream Bytes of RC4. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 253–269. Springer, 2008.
18. Itsik Mantin. Analysis of the Stream Cipher RC4. `http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html`.
19. Itsik Mantin. Predicting and Distinguishing Attacks on RC4 Keystream Generator. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 491–506. Springer, 2005.
20. Itsik Mantin and Adi Shamir. A Practical Attack on Broadcast RC4. In Mitsuru Matsui, editor, *FSE*, volume 2355 of *Lecture Notes in Computer Science*, pages 152–164. Springer, 2001.
21. Alexander Maximov. Two Linear Distinguishing Attacks on VMPC and RC4A and Weakness of RC4 Family of Stream Ciphers. In Henri Gilbert and Helena Handschuh, editors, *FSE*, volume 3557 of *Lecture Notes in Computer Science*, pages 342–358. Springer, 2005.
22. Alexander Maximov and Dmitry Khovratovich. New State Recovery Attack on RC4. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 297–316. Springer, 2008.
23. Ilya Mironov. (Not So) Random Shuffles of RC4. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 304–319. Springer, 2002.
24. Vebjørn Moen, Håvard Raddum, and Kjell Jørgen Hole. Weaknesses in the Temporal Key Hash of WPA. *Mobile Computing and Communications Review*, 8(2):76–83, 2004.
25. Goutam Paul and Subhamoy Maitra. Permutation After RC4 Key Scheduling Reveals the Secret Key. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *Selected Areas in Cryptography*, volume 4876 of *Lecture Notes in Computer Science*, pages 360–377. Springer, 2007.
26. Goutam Paul, Siddheshwar Rathi, and Subhamoy Maitra. On Non-negligible Bias of the First Output Bytes of RC4 towards the First Three Bytes of the Secret Key. In *WCC'07 - International Workshop on Coding and Cryptography*, pages 285–294, 2007.

27. Souradyuti Paul and Bart Preneel. A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher. In Bimal K. Roy and Willi Meier, editors, *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 2004.
28. Andrew Roos. A Class of Weak Keys in RC4 Stream Cipher (sci.crypt), 1995. `http://groups.google.com/group/sci.crypt.research/msg/078aa9249d76eacc?dmode=source`.
29. Erik Tews and Martin Beck. Practical attacks against WEP and WPA. In David A. Basin, Srdjan Capkun, and Wenke Lee, editors, *WISEC*, pages 79–86. ACM, 2009.
30. Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 Bit WEP in Less Than 60 Seconds. In Sehun Kim, Moti Yung, and Hyung-Woo Lee, editors, *WISA*, volume 4867 of *Lecture Notes in Computer Science*, pages 188–202. Springer, 2007.
31. Violeta Tomasevic, Slobodan Bojanic, and Octavio Nieto-Taladriz. Finding an internal state of RC4 stream cipher. *Inf. Sci.*, 177(7):1715–1727, 2007.
32. Serge Vaudenay and Martin Vuagnoux. Passive-Only Key Recovery Attacks on RC4. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *Selected Areas in Cryptography*, volume 4876 of *Lecture Notes in Computer Science*, pages 344–359. Springer, 2007.
33. Martin Vuagnoux. *Computer Aided Cryptanalysis from Ciphers to Side channels*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne — EPFL, 2010.
34. David Wagner. Weak Keys in RC4 (sci.crypt), 1995. `http://www.cs.berkeley.edu/~daw/my-posts/my-rc4-weak-keys`.