Some New Weaknesses in the RC4 Stream Cipher

Jing Lv[1(✉)], Bin Zhang[1], and Dongdai Lin[2]

[1] Laboratory of Trusted Computing and Information Assurance, Institute of Software, Chinese Academy of Sciences, 100190 Beijing, China
[2] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{lvjing,zhangbin}@tca.iscas.ac.cn, ddlin@iie.ac.cn

Abstract. In FSE 2011, Maitra and Paul observed that there exists negative bias in the first byte of the RC4 keystream towards 0. In this paper, we give our theoretical proof of this bias. This bias immediately provide distinguisher for RC4, and ciphertext only attack on broadcast RC4. Additionally, we discover some new weaknesses of the keystream bytes even after the first N rounds of the PRGA, where N is the size of the RC4 permutation, generally, $N = 256$. The weaknesses in turn provide us with certain state information from the keystream bytes no matter how many initial bytes are thrown away.

Keywords: RC4 · Broadcast RC4 · Ciphertext only attack · Distinguishing attack · State recovery attack.

1 Introduction

RC4, designed by Ron Rivest in 1987, is the most widely deployed stream cipher in practical applications. Due to its simplicity and extremely fast software performance, RC4 has been integrated into TLS/SSL and WEP applications. RC4 takes an interesting design approach which is quite different from that of LFSR-based stream ciphers. This implies that many of the analysis methods known for such ciphers cannot be applied. The internal state of RC4 consists of a table of $N = 2^n$ n-bit words and two n-bit pointers, where n is a parameter (for the nominal version, $n = 8$). The table varies slowly in time under the control of itself. When $n = 8$, RC4 has a huge state of $(2^8)^2 * log_2 2^8!$, approximately 1,700 bits. It is thus impractical to guess even a small part of this state, or to use standard time/memory/data tradeoff attacks. In addition, the state evolves in a complex non-linear way, and thus it is difficult to combine partial information about states which are far away in round. Consequently, all the techniques developed to attack stream ciphers based on linear feedback shift registers seem to be inapplicable to RC4.

Y. Kim et al. (Eds.): WISA 2013, LNCS 8267, pp. 28–38, 2014.
DOI: 10.1007/978-3-319-05149-9_2,

The initial bytes(the first N outputs) of RC4 have been thoroughly analyzed in a large amount of papers. In FSE 2011, Matrai and Paul proved that the initial 3–255 bytes of the keystream are positive biased to 0 which are in accordance with the experiment. The experiment also showed that the first keystream byte is negative biased to 0, the proof of the bias was posed as an open problem, see [10]. In this paper, we provide a satisfied proof of this bias, this bias immediately provide distinguisher for RC4, and it can be used for plaintext recovery attack in the broadcast RC4.

In FSE 2013, Alfardan, Bernstein etc. reported their result of all the biases in the first N bytes of the RC4 keystream without theoretical proofs. However, the keystream bytes produced after round N of the PRGA haven't been adequately studied in the last decades, most of previous attacks will fail when the first N bytes of the keystream are dumped. In our paper, we give out some weaknesses exit in all rounds, which will in turn provide us with certain information form any keystream byte and improve the state recovery attack [1,7].

This paper will be organized as follows. In Sect. 2, we introduce the RC4 cipher and the notations we use throughout this paper. We give our theoretical proof of the open problem in [10] in Sect. 3, what's more, the corresponding distinguishing attack and ciphertext only attack were presented. Section 4 details some weaknesses exit in all PRGA rounds of RC4. Finally, we conclude in Sect. 5.

2 Description of RC4

RC4 runs in two phases, the key scheduling phase KSA and the output keystream generation phase PRGA. The description is as follows.

```
1  KSA
2  for i ← 0 to N − 1
3        do s[i] ← i
4  j ← 0
5  for i ← 0 to N − 1
6        do j ← j + s[i] + k[i mod l]
7           swap (s[i], s[j])
8  PRGA
9  i, j ← 0
10 while i ≥ 0
11       do i ← i + 1
12          j ← j + s[i]
13          swap (s[i], s[j])
14          output s[s[i] + s[j]]
```

The KSA swaps N pairs of the array $\{0, 1, 2, \ldots, N-1\}$, depending on the value of the secret key, where l is the word length of the secret key. At the end of KSA, we reach an initial state for PRGA phase, which generates keystream words of $\log_2 N$ bits. Note that the symbol $'+'$ denotes the addition modular N.

Notations. Let s_t, i_t, j_t, z_t denote the state, index i, index j and the keystream byte respectively, after $t(t \geqslant 0)$ rounds of PRGA have been performed. Specially, s_0 is the state just before the PRGA starts, i.e, right after the KSA ends. $'+'('-')$ denotes addition(substraction) modular by N when applying to the algorithm of RC4, where $N = 256$.

3 z_1 Is Negative Biased to 0

In this section, we give the theoretical proof of the negative bias. We state our result by the following theorem.

Theorem 1. *The probability that the first RC4 keystream byte is equal to* 0 *is* $Pr(z_1 = 0) \approx 0.003877$.

During the proof of our theorem, we shall require the following well known result in RC4 crytanalysis from the existing literature. This appears in [5], and we restate the result as follows.

Lemma 1. *At the end of KSA, for* $0 \leqslant u \leqslant N-1$, $0 \leqslant v \leqslant N-1$,

$$Pr(s_0[u] = v) = \begin{cases} \frac{1}{N}[(\frac{N-1}{N})^v + (1 - (\frac{N-1}{N})^v)(\frac{N-1}{N})^{N-u-1}] & if v \leqslant u; \\ \frac{1}{N}[(\frac{N-1}{N})^{N-u-1} + (\frac{N-1}{N})^v] & if v > u. \end{cases}$$

We denote the probability $Pr(s_0[u] = v)$ *as* $p_{u,v}$ *in the following part of the paper.*

We also need the following probability formula through our proof.

Lemma 2. *Let* A, B *be two events with* $Pr(B) \neq 0$, $\{C_i\}_{i=1}^n$ *be a sequence of events satisfied* $Pr(\cup C_i) = 1$ *and* $\cap C_i = \emptyset$. *Then* $\Pr(A|B) = \sum_i \Pr(A|C_i, B)\Pr(C_i|B)$.

Proof.

$$\begin{aligned} Pr(A|B) &= \sum_i Pr(A, C_i|B) = \sum_i \frac{Pr(A, C_i, B)}{Pr(B)} \\ &= \sum_i \frac{Pr(A|C_i, B)Pr(C_i, B)}{Pr(B)} = \sum_i Pr(A|C_i, B)Pr(C_i|B). \end{aligned}$$

Now we will prove Theorem 1 with the lemmas.

The proof of Theorem 1

We prove the result by decomposing the event $z_1 = 0$ into two mutually exclusive and exhaustive cases as follows.

$$\begin{aligned} &Pr(z_1 = 0) \\ &= Pr(z_1 = 0|s_0[1] = 1)Pr(s_0[1] = 1) + Pr(z_1 = 0|s_0[1] \neq 1)Pr(s_0[1] \neq 1) \\ &= p_{1,1}Pr(z_1 = 0|s_0[1] = 1) + (1 - p_{1,1})Pr(z_1 = 0|s_0[1] \neq 1) \end{aligned} \quad (1)$$

Now we consider the events $z_1 = 0|s_0[1] = 1$ and $z_1 = 0|s_0[1] \neq 1$ individually to calculate their probabilities. Note that $z_1 = s_1[s_1[1]+s_1[j_1]] = s_1[s_0[j_1]+s_0[1]] = s_1[s_0[s_0[1]] + s_0[1]]$.

Calculation of $Pr(z_1 = 0|s_0[1] = 1)$. In this case, $j_1 = s_0[1] = 1 = i_1$, and thus s_1 is the same permutation as s_0. Then we have the probability

$$\begin{aligned} Pr(z_1 = 0|s_0[1] = 1) &= Pr(s_1[s_0[s_0[1]] + s_0[1]] = 0|s_0[1] = 1) \\ &= Pr(s_1[2] = 0|s_0[1] = 1) = \frac{1}{N-1} \end{aligned} \tag{2}$$

In fact, we infer from Lemma 1 that $Pr(s_0[a] = 0)$ is uniformly distributed (v=0), with a trivial probability of $\frac{1}{N}$. Also, considering $s_0[2] \neq 1$ when $s_0[1] = 1$. It is reasonable to estimate $Pr(s_0[2] = 0|s_0[1] = 1)$ as $\frac{1}{N-1}$. Generally, we estimate $Pr(s_0[a] = x|s_0[b] = y)$ as $\frac{N}{N-1}p_{a,x}$ when $x \neq y$, $a \neq b$.

Calculation of $Pr(z_1 = 0|s_0[1] \neq 1)$. By Lemma 2,

$$\begin{aligned} &Pr(z_1 = 0|s_0[1] \neq 1) \\ &= Pr(z_1 = 0|s_0[s_0[1]] = 0, s_0[1] \neq 1)Pr(s_0[s_0[1]] = 0|s_0[1] \neq 1) \\ &+ Pr(z_1 = 0|s_0[s_0[1]] = 1 - s_0[1], s_0[1] \neq 1)Pr(s_0[s_0[1]] = 1 - s_0[1]|s_0[1] \neq 1) \\ &+ Pr(z_1 = 0|s_0[s_0[1]] \neq 0, 1 - s_0[1], s_0[1] \neq 1)Pr(s_0[s_0[1]] \neq 0, 1 - s_0[1]|s_0[1] \neq 1) \end{aligned}$$

Now we consider the three parts of the equation separately.

For $Pr(z_1 = 0|s_0[s_0[1]] = 0, s_0[1] \neq 1)$, since $s_0[1] \neq 1$, $0 = s_0[s_0[1]] \neq s_0[1]$, thus we get $z_1 = s_1[s_0[1] + s_0[s_0[1]]] = s_1[s_0[1]] = s_0[1] \neq 0$. Therefore, $Pr(z_1 = 0|s_0[s_0[1]] = 0, s_0[1] \neq 1) = 0$.

For $\Pr(z_1 = 0|s_0[s_0[1]] = 1 - s_0[1], s_0[1] \neq 1)$, since $s_0[1] \neq 1$, $s_0[s_0[1]] = 1 - s_0[1] \neq 0$, thus we get $z_1 = s_1[s_0[s_0[1]] + s_0[1]] = s_1[1] = s_0[s_0[1]] \neq 0$. Therefore $\Pr(z_1 = 0|s_0[s_0[1]] = 1 - s_0[1], s_0[1] \neq 1)$=0.

For $\Pr(z_1 = 0|s_0[s_0[1]] \neq 1 - s_0[1], s_0[1] \neq 1)$, since $s_0[s_0[1]] + s_0[1] \neq 1, s_0[1]$. $z_1 = s_1[s_0[s_0[1]] + s_0[1]] = s_0[s_0[s_0[1]] + s_0[1]]$. Thus we get

$$\begin{aligned} &Pr(z_1 = 0|s_0[s_0[1]] \neq 0, 1 - s_0[1], s_0[1] \neq 1) \\ &= Pr(s_0[s_0[s_0[1]] + s_0[1]] = 0|s_0[s_0[1]] \neq 0, 1 - s_0[1], s_0[1] \neq 1) \end{aligned}$$

the value of s_0 at $s_0[1]+s_0[s_0[1]]$ are independent of the value at $s_0[1]$, 1, because of the randomness of $s_0[1]$. Through a similar analysis with (2), we get

$$Pr(z_1 = 0|s_0[s_0[1]] \neq 0, 1 - s_0[1], s_0[1] \neq 1) = \frac{1}{N-1}$$

Combine these results, we get

$$\begin{aligned} &Pr(z_1 = 0|s_0[1] \neq 1) \\ &= \frac{1}{N-1}Pr(s_0[s_0[1]] \neq 0, 1 - s_0[1]|s_0[1] \neq 1) \\ &= \frac{1}{N-1}\sum_{x=0}^{N-1} \Pr(s_0[s_0[1]] \neq 0, 1 - s_0[1]|s_0[1] = x, s_0[1] \neq 1) \Pr(s_0[1] = x|s_0[1] \neq 1) \end{aligned}$$

$$= \frac{1}{N-1}\sum_{x\neq 1}\Pr(s_0[x]\neq 0, 1-x|s_0[1]=x)\Pr(s_0[1]=x|s_0[1]\neq 1)$$
$$= \frac{1}{N-1}\sum_{x\neq 1}[1-\Pr(s_0[x]=0|s_0[1]=x)-\Pr(s_0[x]=1-x|s_0[1]=x)]p_{1,x}*\frac{N}{N-1}$$
$$= \frac{N}{(N-1)^2}(\sum_{x\neq 1}p_{1,x}-\sum_{x=2}^{N-1}\frac{1}{N-1}p_{1,x}-\frac{N}{N-1}\sum_{x\neq 1}p_{x,1-x}p_{1,x}) \quad (3)$$

Combining the probabilities from Eqs. (2) and (3) into (1), we obtain the following

$$Pr(z_1=0)$$
$$= \frac{1}{N-1}p_{1,1}+\frac{N}{(N-1)^2}(\sum_{x\neq 1}p_{1,x}-\sum_{x=2}^{N-1}\frac{1}{N-1}p_{1,x}-\frac{N}{N-1}\sum_{x\neq 1}p_{x,1-x}p_{1,x})(1-p_{1,1})$$

Now, substituting the values of $p_{m,n}$ from Lemma 1, we obtain

$$Pr(z_1=0)\approx 0.003877. \quad (4)$$

We run the RC4 algorithm 1 billion times, each with a randomly generated 16 byte key, and obtain z_1. The probability of $z_1=0$ is 0.003896, which is slightly larger than our theoretical result, this may due to the approximation of the probability $Pr(s_0[a]=x|s_0[b]=y)$.

3.1 A New Distinguisher

Theorem 1 immediately give a new distinguisher. In [3], it is proved that if an event e happens with probabilities p and $p(1+q)$ in distributions X and Y respectively, then for p and q with small magnitude, $O(p^{-1}q^{-2})$ samples suffice to distinguish X from Y with a constant probability of success.

In our setting, let X and Y denote the distributions corresponding to random stream and RC4 keystream respectively, and e denotes the event $z_1=0$. From Eq. (4), we have $Pr(z_1=0)\approx\frac{1}{N}(1-0.007488)$, thus $p=\frac{1}{N}$, $q=0.007488$. Therefore, to distinguishing RC4 keystream from random stream, based on the event $z_1=0$, one would need number of samples of the order of $(\frac{1}{N})^{-1}*0.007488^{-2}\sim O(N^3)$. We list the distinguishers of the form $z_t=0$ in Table 1.

Table 1. Distinguishers of the form $z_t=0$.

Round number t	Data complexity	Reference
1	$O(N^3)$	Our
2	$O(N)$	[3]
3–255	$O(N^3)$	[10]

3.2 A Ciphertext-Only Attack on Broadcast RC4

A broadcast cipher is a multi-round protocol in which each general broadcasts the same message to all the other generals, where each copy is encrypted under a different key agreed in advance between any two generals. For example, many users send the same email message to multiple recipients(encrypted under different keys), and many groupware applications enable multiple users to synchronize their documents by broadcasting encrypted modification lists to all the other group members. By using RC4, the generals will succeed in reading coordinated decisions, however, an enemy will probably collects all the ciphertext and recover the first plaintext.

Theorem 2. *Let M be a plaintext, and let $C_1 \cdots C_k$ be the RC4 encryptions of M under k uniformly distributed keys. Then if $k = O(N^3)$, the first byte of M can be reliably extracted from $C_1 \cdots C_k$.*

Proof. Recall from Theorem 1 that $Pr(z_1 = 0) \approx 0.003877$. Thus, for each encryption key chosen during broadcast, the first plaintext byte M[1] has probability 0.003877 to be XOR-ed with 0.

Due to the bias of z_1 towards 0, 0.003877 fraction of the first ciphertext byte will have the same value as the first plaintext byte, with a lower probability. When $k = O(N^3)$, the attacker can identify the less frequent character in $C_1[1] \cdots C_k[1]$ with probability 0.003877 as M[1] with constant probability of success.

Experiment. We generate k 16 byte keys, and obtain k keystreams, these keystreams are used to encrypt the same message. When $k = 2^{27}$, the success probability is only 16 %, and it reaches 70 % when $k = 2^{30}$. The reason for higher data complexity is that the probabilities $Pr(z_1 = 253)$, $Pr(z_1 = 254)$, $Pr(z_1 = 255)$ are only slightly larger than the probability $Pr(z_1 = 0)$, which we can see from the experiment result of [2]. Thus we need more keystreams to distinguishing them.

In [3,10], there are plaintext recovery attack on $M[2]$ to $M[N-1]$, together with our recovery on $M[1]$, one can consist a plaintext recovery attack on the first N bytes of RC4. What's more, if we apply the biased sequence of the form ABSAB in [4] to recover the bytes after round N as well, a full plaintext recovery attack is possible.

4 Some New Weaknesses of RC4

When we take a closer look at the proofs of our bias on $z_1 = 0$ and other biases exit in the first N bytes mentioned in [3,6,10], we will find that most of the biases are due to the non-uniformly distributed s_0. That means the initialization of RC4 is weak. However, all these attacks become infeasible when the first N bytes of the keystream are dumped, in fact, when the round number t is large enough, the permutation s_t is uniformly distributed. In this section, we present two general weaknesses of RC4, these weaknesses exit no matter how many keystream bytes are dumped.

4.1 The Weakness about $z_t = 0$

We express the first weakness by the following theorem.

Theorem 3. *When the round number $t \geqslant 1$, if $s_{t-1}[t+1] = 0$, $j_{t-1} = 0$ and $s_{t-1}[t] \neq t+1$, then $z_{t+1} = 0$.*[1]

Proof. The proof comes from the execution process of the cipher. At the tth round, j_t is updated by $j_t = j_{t-1} + s_{t-1}[t] = s_{t-1}[t] \neq t+1$, together with $i_t = t \neq t+1$, we get $s_t[t+1] = s_{t-1}[t+1] = 0$. During the $(t+1)$th round, j_{t+1} is updated by $j_{t+1} = j_t + s_t[t+1] = j_t$, therefore we swap $s_t[t+1]$ and $s_t[j_t]$ to update the state s_{t+1}. From above, we obtain

$$\begin{aligned} z_{t+1} =& s_{t+1}[s_{t+1}[t+1] + s_{t+1}[j_t]] = s_{t+1}[s_t[j_t] + s_t[t+1]] \\ =& s_{t+1}[s_{t-1}[t]] = s_{t+1}[j_t] = s_t[t+1] = 0. \end{aligned}$$

As we know, the non-randomness of $z_{t+1} = 0$ will give new distinguishers. We denote E_{int} the event $s_{t-1}[t+1] = 0$, $j_{t-1} = 0$ and $s_{t-1}[t] \neq t+1$, then it follows immediately from Theorem 3 that $Pr(z_{t+1} = 0|E_{int}) = 1$. If we assume that when E_{int} does not occur, $z_{t+1} = 0$ happens with probability $\frac{1}{N}$, then the probability of $z_{t+1} = 0$ is computed as follows

$$\begin{aligned} Pr(z_{t+1} = 0) =& Pr(z_{t+1} = 0|E_{int})Pr(E_{int}) + Pr(z_{t+1} = 0|\overline{E_{int}})Pr(\overline{E_{int}}) \\ =& Pr(E_{int}) + \frac{1}{N}(1 - Pr(E_{int})) \end{aligned} \tag{5}$$

When t is large, s_t and j_t are expected to be uniformly distributed, thus $Pr(E_{int}) = \frac{1}{N^2}(1 - \frac{1}{N})$. We substitute this probability to (5) and get when t is large,

$$Pr(z_{t+1} = 0) = \frac{1}{N}(1 + \frac{1}{N}(1 - \frac{1}{N})^2) \tag{6}$$

Equation (6) implies a large bias. Unfortunately, experiment shows that when t is large, z_{t+1} is only a little positive biased towards 0. The bias is not so large as (6) claims, thus hard to detective. By carefully analyzing this situation one can show that though the event E_{int} is correctly computed, the probability $Pr(z_{t+1} = 0|E_{int})$ is slightly negative biased, i.e, smaller than $\frac{1}{N}$, thus cancels the positive bias. Therefore, we can still estimate the probability of $Pr(z_{t+1} = 0)$ by $\frac{1}{N}$. However, we can detect inner state from the keystream by this theorem. The event $z_{t+1} = 0$ is an external event in the keystream which we can obtain, while the event E_{int} an internal event of the inner state which is non-visible. By Theorem 3, $Pr(z_{t+1}|E_{int}) = 1$. We are more interested in the event $E_{int}|z_{t+1}$, since it means detecting the inner state from the known keystream.

Theorem 4. *When $z_{t+1} = 0$, the event $s_{t-1}[t+1] = 0$ and $j_{t-1} = 0$ happens with a probability larger than $\frac{1}{N} - \frac{1}{N^2}$, which is greatly larger than the random case of $\frac{1}{N^2}$.*

[1] All the operation $'+'$ and $'-'$ applying to the algorithm of RC4 are modular by N, and the notation $s_t[t_1]$ means $s_t[t_1 mod N]$.

Proof. We denote E_{int} as mentioned above. Applying Bayes formula we can derive the following.

$$\begin{aligned} &Pr(s_{t-1}[t+1]=0, j_{t-1}=0|z_{t+1}=0) \\ &\geqslant Pr(E_{int}|z_{t+1}) = \frac{Pr(z_{t+1}|E_{int})Pr(E_{int})}{Pr(z_{t+1})} \\ &= \frac{\frac{1}{N^2}(1-\frac{1}{N})}{\frac{1}{N}} = \frac{1}{N}(1-\frac{1}{N}) \end{aligned}$$

From Theorem 4, one can guess j_{t-1} and $s_{t-1}[t+1]$ for more than the probability of a random guess of $\frac{1}{N^2}$, every time we obtain $z_{t+1} = 0$ in the RC4 keystream.

Experiment. In Fig. 1, we plot the experiment values observed by running the RC4 algorithm 1 billion times each with a randomly selected 16 byte key, the initial $51 * N - 1$ rounds of keystream bytes are thrown away, we start from the $51 * N$th round. We can see from the figure that most of the probability are around $\frac{1}{N}$, all of them are much greater than $\frac{1}{N^2} \approx 0.000015$. But some of them are lower than $\frac{1}{N} - \frac{1}{N^2} \approx 0.00389099$, this may due to the probability of $Pr(z_{t+1} = 0)$ is slightly positive biased at some ts.

4.2 The Weakness about $z_t = z_{t+1}$

We will introduce another general weakness of RC4 in this subsection.

Theorem 5. *When $t \geqslant 1$ and $t \neq -2 (mod\ N)$, if $j_{t-1} = 0$, $s_{t-1}[t] = t+1$, then we have $z_t \neq z_{t+1}$.*

Proof. At the tth round, j_t is updated by $j_t = j_{t-1} + s_{t-1}[t] = t + 1$, together with $i_t = t$, we get $s_t[t+1] = s_{t-1}[t] = t+1$, $s_t[t] = s_{t-1}[t+1]$. And the output is

$$z_t = s_t[s_t[t+1] + s_t[t]] = s_t[s_t[t] + t + 1].$$

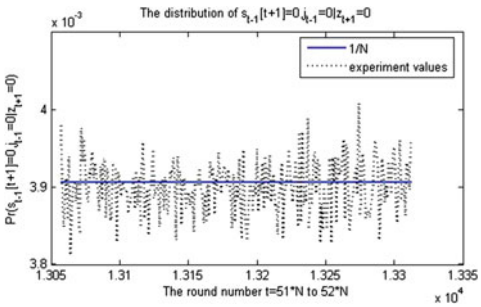

Fig. 1. The probability of $Pr(s_{t-1}[t+1] = 0, j_{t-1} = 0|z_{t+1} = 0)$ at $51 * N$ to $52 * N$ rounds

During the $(t+1)$th round, j_{t+1} is updated by $j_{t+1} = j_t + s_t[t+1] = 2t+2$, therefore we swap $s_t[t+1]$ and $s_t[2t+2]$, i.e, $s_{t+1}[t+1] = s_t[2t+2]$, $s_{t+1}[2t+2] = s_t[t+1] = t+1$. And the output is

$$z_{t+1} = s_{t+1}[s_{t+1}[t+1] + s_{t+1}[2t+2]] = s_{t+1}[s_t[2t+2] + t + 1].$$

We derive from $t \neq -2$ that $s_t[t] \neq s_t[2t+2]$, thus the indices of z_t and z_{t+1} are unequal. Therefore if $z_t = z_{t+1}$, the indices of z_t and z_{t+1} are both the exchange indices at round $t+1$, there are two cases

$$s_t[t] + t + 1 = t + 1, s_t[2t+2] + t + 1 = 2t + 2 \tag{7}$$

or

$$s_t[t] + t + 1 = 2t + 2, s_t[2t+2] + t + 1 = t + 1 \tag{8}$$

For (7), $s_t[2t+2] = t+1 = s_t[t+1]$, thus $t = -1, (\mathrm{mod} N)$. Substitute the value of t to (7), we get $s_{-1}[-1] = s_{-1}[0] = 0$. This contradicts to the fact that s_{-1} is a permutation. Equation(8) implies $s_t[t] = t+1 = s_t[t+1]$, this is also impossible.

In [8,9], S.Paul and B.Preneel gave their discovery about the non-randomness of the event $z_1 = z_2$. However, there hasn't been much research on the distribution of the events $z_t = z_{t+1}$ when $t > 1$. Similar to the analysis of the event $z_t = 0$, when applying to the first N bytes, the non-uniformly distributed s_0 has big influence on the event $z_t = z_{t+1}$, while to the round number t is large enough, the state s_t is expected to be uniformly distributed, we plot the distribution of $Pr(z_t = z_{t+1})$ in Fig. 2, also, we run the RC4 algorithm 1 billion times, each with a randomly selected 16 byte key. We conclude from the figure that when the round number t is small, the probability is lower than random case of $\frac{1}{N}$, and when t is large enough, it is uniformly distributed. The same as the weakness mentioned in Sect. 4.1, it will leads to information leakage when t is large enough.

Theorem 6. *When $z_t \neq z_{t+1}$ and $t \neq -2$, the event $j_{t-1} = 0$, $s_{t-1}[t] = t+1$ happens with probability of $\frac{1}{N^2-N}$, which is larger than $\frac{1}{N^2}$.*

Proof. When $t \neq -2$, using Theorem 5 as well as applying Bayes formula we can derive the following.

$$\begin{aligned} &Pr(s_{t-1}[t] = t+1, j_{t-1} = 0 | z_t \neq z_{t+1}) \\ &= \frac{Pr(z_t \neq z_{t+1} | s_{t-1}[t] = t+1, j_{t-1} = 0) Pr(s_{t-1}[t] = t+1, j_{t-1} = 0)}{Pr(z_t \neq z_{t+1})} \\ &= \frac{\frac{1}{N^2}}{1 - \frac{1}{N}} = \frac{1}{N^2 - N} \end{aligned}$$

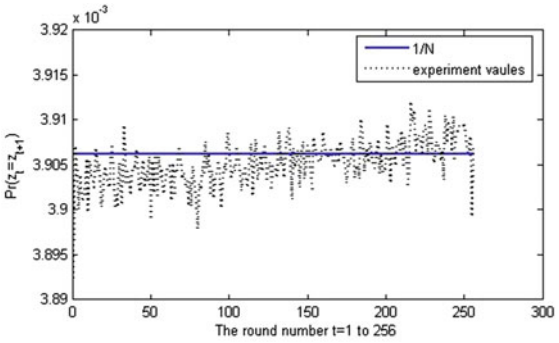

Fig. 2. The probability of $Pr(z_t = z_{t+1})$ at the first N rounds

5 Conclusion

In this paper, we give out the theoretical proof of the negative bias of z_1 towards 0, which is an open problem proposed in [10]. This bias can distinguish RC4 keystream reliably from a random stream of bytes. Further, the bias can be exploited to mount an attack against broadcast RC4. In addition to the 2th to 255th plaintext bytes recovery in [3,10], we are able to recover the first N bytes.

Further, we propose some weaknesses in the whole PRGA phase, contrary to the previous work, the weaknesses still exists even though the first N bytes are dumped, and will lead to the leakage of the state information. We would like to make a small note on a related observation, the probability of $Pr(z_{t+1} = 0|\overline{s_{t-1}[t+1] = 0, j_{t-1} = 0, s_{t-1}[t+1] \neq t+1})$ is smaller than $\frac{1}{N}$, i.e, slightly negative biased, where $\bar{A}$ denotes the complement event of A. But we haven't found the reason, we would like to pose this as an open problem.

Acknowledgment. This work was supported by the National Basic Research 973 Program of China under Grant NO.2013CB338002. The authors would like to thank the anonymous reviewers for their helpful suggestions.

References

1. Maximov, A., Khovratovich, D.: New state recovery attack on RC4. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 297–316. Springer, Heidelberg (2008)
2. Bernstein, D.: Failures of secret-key cryptography. Fast Software Encryption-FSE'2013, inviting talk. http://fse2013.spms.ntu.edu.sg/slides/slides07.pdf
3. Mantin, I., Shamir, A.: A practical attack on broadcast RC4. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 152–164. Springer, Heidelberg (2002)
4. Mantin, I.: Predicting and distinguishing attacks on RC4 keystream generator. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 491–506. Springer, Heidelberg (2005)
5. Mantin, I.: Analysis of the stream cipher RC4. Master's Thesis, The Weizmann Institute of Science, Israel (2001)

6. Isobe, T., Ohigashi, T.: Full plaintext recovery attack on broadcast RC4. In: Fast Software Encryption-FSE'2013 (2013)
7. Knudsen, L.R., Meier, W., Preneel, B., Rijmen, V., Verdoolaege, S.: Analysis methods for (alleged) RC4. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 327–341. Springer, Heidelberg (1998)
8. Paul, S., Preneel, B.: Analysis of non-fortuitous predictive states of the RC4 keystream generator. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 52–67. Springer, Heidelberg (2003)
9. Paul, S., Preneel, B.: A new weakness in the RC4 keystream generator and an approach to improve the security of the cipher. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 245–259. Springer, Heidelberg (2004)
10. Maitra, S., Paul, G., Sen Gupta, S.: Attack on Broadcast RC4 Revisited. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 199–217. Springer, Heidelberg (2011)

http://www.springer.com/978-3-319-05148-2

Information Security Applications
14th International Workshop, WISA 2013, Jeju Island, Korea, August 19-21, 2013, Revised Selected Papers
Kim, Y.; Lee, H.; Perrig, A. (Eds.)
2014, XII, 273 p. 79 illus., Softcover
ISBN: 978-3-319-05148-2