L-P States of RC4 Stream Cipher

immediate

May 13, 2013

Abstract

The stream cipher RC4 was designed by R.Rivest in 1987, and it is a widely deployed cipher. Many predictive states of RC4 for some special indices i were presented in the last 20 years. In this paper, we present several long term predictive states. These states increase the probability to guess part of the internal state in a known plaintext attack and present a cryptanalytic weakness of RC4. This paper also analyzes possible long term bias in the keystream and further propose a search method for the long term predictive states.

Keywords: RC4, Distinguishing attack, predictive states, L-P states.

1 Introduction

RC4 is the most widely used software based stream cipher. The cipher has been integrated into SSL and WEP implementations. RC4 was designed by Rivest in 1987 and kept as a trade secret until it was leaked out in 1994, it is extremely fast and its design is simple, the simplicity of the keystream generating algorithm of RC4 has attracted many cryptanalysis efforts.

In this paper we study a family of RC4 like stream ciphers named RC4-N, where N is the modulus of operations. The internal state of RC4 is two indices i, $j \in Z_N$ and a permutation of Z_N denoted s. Thus, RC4 has a state of $log_2(N^2 * N!)$ bits. For original version when $N = 256$, the size of the state is ≈ 1700 bits.

The initial bytes (the first N outputs) of RC4 have been thoroughly analysed in a large amount of papers, to name but a few [2, 8, 9]. These results show that the initialision of RC4 is weak, thus distinguishing attacks and plaintext recovery attacks can be applied to RC4 with low data and time complexity. Mantain and Shamir described in [2] that the second output of RC4 is greatly biased towards 0, the bias of the first and second output word was presented in [8], also,a plaintext recovery attack using these new biases was demonstrated. However, if one dumps the initial N words, the cipher becomes secure against these attacks.

The state recovery attack has attracted many attentions. It is known that the predictive states play an important role in the state recovery attack, where predictive states means partial states that suffice for determining keystream output for several rounds. Two predictive states were presented in [2] to improve the state recovery attack proposed in [6]. Several properties of predictive states were studied in [7], besides, an algorithm was proposed to compute the predictive states . The best state recovery attack so far was proposed in [1], proper predictive states were searched in the precomputation stage of the attack. They assumed in the paper that 'good' predictive states always begin at $i = 0$, under this assumption, they experimentally discovered several long-window predictive states.

In this paper, we give the concept of long term predictive states, which are predictive states sequence. To our best knowledge, the first long term predictive state was proposed in [3], with the construction $i = x-1$, $j = x+2$, $s[x] = -1$ and $s[x+1] = x+1$, this state predicts the outputs at $i = x$ and $i = x+1$, where $x \in \{0 \cdots N-1\}$. In [5], two types of long term predictive states about $z_t = z_{t+1}$ and $z_t = 0$ were showed. Long term predictive states increase the probability to guess part of the internal state in a known plaintext attack and present a cryptanalysis weakness of RC4. We present two new long term predictive states in this paper, then analysis whether these predictive states will lead to long term bias. What's more, a general method to compute the long term predictive states is proposed.

The paper is organized as follows: In section 2 we briefly introduce RC4 cipher and the notations we use throughout this paper. Section 3 details the concept of predictive state and L-P state. Then, we present previous discoveries of predictive states in section 4. In section 5 we introduce our new L-P states. Section 6 describes possible long term bias and a search method of the long term predictive states. Finally, conclusions and directions of our future work will be presented in section 7.

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  while i ≥ 0
10       do i ← i + 1
11          j ← j + s[i]
12          swap (s[i], s[j])
13          output s[s[i] + s[j]]
```

Where $s[N]$ is a permutations of all the elements of Z_N. All internal variables of RC4 are over ring Z_N, where N is the size of the ring. To specify a particular instance of the cipher we denote it by RC4-N. Thus, the original design is RC4-256. Whenever applicable, '+' and '-' are performed in modular N, $\emptyset$ denotes the empty event. At any round t, the notation a_t denotes the value of a variable at time t and z_t denotes the tth output of the keystream, $s_t[l]$ denotes the lth element of the array after the swapping period at round t.

3 L-P States

In [2], a set of special RC4 states, known as predictive states, have been conceptualized. In the following part we give the definition of a predictive state with a little modification to one given in [2] to suit our analysis.

Definition 1. *Let A be an a-state(i.e only a elements of the array s and the indices i, j are known), if all*[1] *the RC4 states that are compatible with A produce the same non-trivial event B in the keystream, then A is said to be a* ***predictive state****.*

Non-trivial event means the event with a prior probability smaller than 1. For example, the event $z_t = z_{t+1}$ happens with a prior probability $\frac{1}{N} < 1$, so it is a non-trivial event, while the event $z_t \in \{0 \cdots N-1\}$ is a trivial event since it happens with probability 1, we assume that B is non-trivial event since nothing can be detected by a trivial event in the keystream. We can see from the definition that a predictive state starts at some special index i, in other words, at some special round $t(i = t(modN))$, while the long term predictive state exits at most of the rounds.

[1] We can generalize the definition without affecting the analysis by allowing some exceptions due to rare coincidence.

Definition 2. *Let A_t be a internal state at round t, i.e $s_t[\sigma_m(t)] = \xi_m(t)$, $m{=}1 \cdots a$, and the indices $i = t(modN)$, $j_t = \varphi(t)$, where σ_m, ξ_m and φ are functions of t. B_t is an external non-trivial event in the keystream. If all*[2] *the RC4 states that are compatible with A_t at round t produce B_t, then A_t is said to be a* ***long term predictive state****, or* ***L-P state*** *for short, and we say A_t* ***predicts*** *B_t.*

Corollary 1 follows immediately from Definition 1 and 2.

Corollary 1. *At any time t, a L-P state A_t is also a predictive state.*

The existence of predictive states and long term predictive states are important for the cryptanalyst, as a elements of the array s_t together with j_t can be extracted with non-trivial probability by observing specific output in the output segment.

Considering the event A_t (that the current state is compatible with the a-state A_t), and B_t(that the output event predicted by A_t), A_t includes $(a+1)$ constraints(a elements of s and j) and thus has a probability of $\frac{1}{N} \cdot \frac{(N-a)!}{N!} \approx N^{-(a+1)}$, whenever A_t occurs, B_t occurs with probability 1. So the probability of $A_t|B_t$ is computed as follows:

$$Pr(A_t|B_t) = \frac{Pr(A_t, B_t)}{Pr(B_t)} = \frac{Pr(B_t|A_t)Pr(A_t)}{Pr(B_t)} = \frac{Pr(A_t)}{Pr(B_t)}$$

For the RC4 stream cipher, A_t can be regarded as the internal state which is unknown to us, while B_t is an external event which can be observed in the keystream segment. So a L-P state can be used to extract partial information of the internal state with non-trivial probability at any round, or at least most of the rounds. As we know, the event B_t can be used for distinguishing attacks if B_t has a non-trivial probability. We denote the prior probability of B_t by q. Under the assumption in [2] that when the event A_t dose not occur, B_t happens with probability q, then the probability of B_t is computed as follows:

$$\begin{aligned} Pr(B_t) =& Pr(B_t|A_t)Pr(A_t) + Pr(B_t|\bar{A}_t)Pr(\bar{A}_t) \\ =& Pr(A_t) + q \cdot Pr(\bar{A}_t) \\ =& q[1 + (q^{-1} - 1)N^{-a-1}] \end{aligned}$$

From this equation, it seems that any L-P state will lead to long term bias. However, this assumption is not always true as we will discuss in section 6.

4 Previous Work on Predictive and L-P States

There are a lot of researches on predictive states in the last 20 years, some of which are presented in this section.

[2]The same as footnote 1.

4.1 Two Predictive States

The concept of predictive state first appears in [2], in which Shamir and Mantin claimed that $z_2 = 0$ whenever $s_0[2] = 0$ and $s_0[1] \neq 2$. This state was used for ciphertext-only attacks.

Theorem 1. *$z_2 = 0$ whenever $s_0[2] = 0$ and $s_0[1] \neq 2$.*

Proof. At round 1, j_1 is updated by $j_1 = j_0 + s_0[1] = s_0[1] \neq 2$, thus we get $s_1[s_0[1]] = s_0[1]$, $s_1[1] = s_0[s_0[1]]$. During round 2, j_2 is updated by $j_2 = j_1 + s_1[2] = s_0[1] + s_1[2]$. Since $i_1, j_1 \neq 2$, $s_1[2] = s_0[2] = 0$, $j_2 = s_0[1]$. Therefore $s_2[s_0[1]] = s_1[2] = 0$, $s_2[2] = s_1[s_0[1]] = s_0[1]$. The output is $z_2 = s_2[s_2[2] + s_2[s_0[1]]] = s_2[s_0[1]] = 0$. □

Paul and Preneel presented in [8] a new predictive state which will influence the distribution of the first two output words, we present their theorem here without proof.

Theorem 2. *If $s_0[1] = 2$ then the first two output words of RC4 are always different.*

4.2 Analysis For the Predictive States

An analysis for the non-fortuitous states was presented in [7], they formally proved the conjecture in [2]. We start by interpreting the concept of "b-predictive", where the b-predictive means that b output words are predicted by the a-state.

Theorem 3. *If any b-predictive a-state exits then $a \geq b$.*

Besides, a general approach for searching for a-predictive states was proposed. The approach first determines the possible relative positions of a elements at the initial round, and then determines the states. Suppose the a elements at the initial round $p_1, p_2 \cdots p_a$, and the distances $d_t = p_{t+1} - p_t + 1$. The advantage of their method by limiting d smaller than d_t^{max} instead of exhaustive research, however, how to compute d_t^{max} is still an open problem. Several predictive states were found with this algorithm. See [7] for details.

4.3 The Best State Recovery Attack So Far

In [1], long window predictive states were generated for their state recovery attack, where long window means a long segment of js are known, but the number of known elements of s are expected to be small. Good predictive states play a pivotal role in their state recovery attack.

Definition 3. *A pattern A is called w-generation if for any internal state complaint with A the next w clockings allow to derive w equations of $s_k^{-1}[z_k] = s_k[i_k] + s_k[j_k]$, i.e. consecutive $w + 1$ values of js are known.*

Some observations on the construction of the pattern show that long window patterns are likely to have values from a short interval $I_\delta = [-\delta \cdots \delta]$, thus the index i is supposed to start at 0. The time complexity of the state recovery attack for RC4-256 will fall to $O(2^{241})$, if a 29-state 168-long pattern is found. We conclude from the definition that our predictive state defined in Definition 1 is a bit different from the pattern. In fact, the pattern may ensures no special output event, while the predictive state ensures a non-trivial event, on the other hand, the proof of Theorem 1 indicates that the index j may lost its value before the certain outputs(j_1 and j_2 are unknown).

4.4 Previous L-P States

The first L-P state was proposed in [3], in the paper they called the L-P state they discover as an recyclable state.

Theorem 4. *If $j_t = t + 3$, $s_t[t + 1] = -1$, $s_t[t + 2] = t + 2$, then we have $z_{t+1} = t + 2$, $z_{t+2} = -1$.*

Jing Lv and Bin Zhang generalized in [5] the predictive states illustrated in [2] and [8] to L-P states.

Theorem 5. *If $s_{t-1}[t+1] = 0$, $j_{t-1} = 0$ and $s_{t-1}[t] \neq t+1$, then $z_{t+1} = 0$.*

This theorem spreads the imbalance of the event $z = 0$ at time 0 to all the keystream segment.

Theorem 6. *When $t \neq -1, -2$, if $j_{t-1} = 0$, $s_{t-1}[t] = t+1$. Then $z_t \neq z_{t+1}$.*

We have explained in Definition 2 that the condition $t \neq -1, -2$ is permitted since some rare exceptions are allowed.

5 Our New L-P States

Our first observation is a L-P state which leads to same outputs of adjacent positions, while the L-P state in Theorem 6 leads to different outputs of adjacent positions.

Theorem 7. *If $j_t = t+1$, $s_t[t+1] = -1$, $s_t[t] \neq t+1, t+2$, then $z_t = z_{t+1}$.*

Proof. The proof comes from the execution process of the cipher. First, the tth output is

$$z_t = s_t[s_t[t] + s_t[j_t]] = s_t[s_t[t] + s_t[t + 1]] = s_t[s_t[t] - 1]$$

Next, during the $(t + 1)$th round, the index j_{t+1} is updated by
$j_{t+1} = j_t + s_t[t + 1]$=$t$, so we exchange the value of $s_t[t + 1]$ and $s_t[t]$, and output

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

Since the indices of z_t and z_{t+1} are equal, so they are equal except $s_t[s_t[t]-1]$ is changed during round $t+1$, in other words, the value of $s_t[t]-1$ is t or $t+1$, i.e $s_t[t] = t+1, t+2$. □

Our second discovery is a state which will predict a special output in the keystream.

Theorem 8. *When N=256, then if* $j_t = 0$, $s_t[t+1] = 131+t$ *and* $s_t[t+2] = 128$, *we have* $z_{t+3} = 131 + t$.

Proof. At the $(t+1)$th round, j_{t+1} is updated by $j_{t+1} = j_t + s_t[t+1] = 131+t$, thus we get

$$\begin{aligned} s_{t+1}[t+1] &= s_t[131+t] \\ s_{t+1}[131+t] &= s_t[t+1] = 131+t \end{aligned}$$

During the $(t+2)$th round, j_{t+2} is updated by $j_{t+2} = j_{t+1} + s_{t+1}[t+2] = t + 131 + 128 = t + 3$, in fact, since $t+2$ is not the exchange index at round $t+1$, $s_{t+1}[t+2] = s_t[t+2] = 128$, therefore we swap $s_{t+1}[t+2]$ and $s_{t+1}[t+3]$, so $s_{t+2}[t+3] = s_{t+1}[t+2] = 128$. Finally, at round $t+3$, $j_{t+3} = j_{t+2} + s_{t+2}[t+3] = t+3+128 = t+131$, so we exchange $s_{t+2}[t+3]$ and $s_{t+2}[t+131]$, then

$$\begin{aligned} z_{t+3} =& s_{t+3}[s_{t+3}[t+3] + s_{t+3}[t+131]] = s_{t+3}[s_{t+2}[t+3] + s_{t+2}[t+131]] \\ =& s_{t+3}[128 + s_{t+1}[t+131]] = s_{t+3}[128+131+t] = s_{t+3}[t+3] \\ =& s_{t+2}[t+131] = t+131. \end{aligned}$$

We use the equation $s_{t+2}[t+131] = s_{t+1}[t+131]$ in the proof, the reason for this equation is that $t+131$ is not the exchange index at round $t+2$. □

6 Possible Long Term Bias

6.1 Previous Work

Let A be the internal event that $s_0[2] = 0$ and $s_0[1] \neq 2$, while B be the external event that $z_2 = 0$, then $Pr(B|A) = 1$ [2]. Under the assumption that when A does not happen, B occurs with the trivial probability $\frac{1}{N}$, and the state s is uniformly distributed, the probability of B is computed as follows:

$$\begin{aligned} Pr(B) =& Pr(B|A)Pr(A) + Pr(B|\bar{A})Pr(\bar{A}) \\ =& 1 * \frac{1}{N}(1-\frac{1}{N}) + \frac{1}{N}[1 - \frac{1}{N}(1-\frac{1}{N})] \\ \approx& \frac{2}{N}. \end{aligned} \tag{1}$$

The assumptions were also used in [5] to produce their long term bias. When $t \neq -1, -2$, let B_t denotes the event that $z_t = z_{t+1}$, while A_t denotes the event that $j_{t-1} = 0$ and $s_{t-1}[t] = t + 1$. Then the probability of B_t is computed as follows

$$\begin{aligned} Pr(B_t) =& Pr(B_t|A_t)Pr(A_t) + Pr(B_t|\bar{A}_t)Pr(\bar{A}_t) \\ =& 0 * \frac{1}{N^2} + \frac{1}{N}(1 - \frac{1}{N^2}) \\ =& \frac{1}{N}(1 - \frac{1}{N^2}). \end{aligned} \tag{2}$$

The event $z_t = 0$ is computed in a similar way by anther L-P state in [5], the probability that $z_t = 0$ turn out to be

$$Pr(z_t = 0) = \frac{1}{N}(1 + \frac{1}{N}(1 - \frac{1}{N})^2). \tag{3}$$

6.2 Correctly Calculate the Probability of External Event

Experiment shows that when t is large, (2) is far from correct. what's more, though $z_t = 0$ is positive biased, the bias is not so large as (3) claims. We will analyze the reason in this subsection. As defined before, we let B_t denotes the event $z_t = z_{t+1}$, while A_t the event that $j_{t-1} = 0$ and $s_{t-1}[t] = t+1$. We discover from the experiment that though the theoretical result of $Pr(A_t)$ is in accordance with the experiment, the probability of $B_t|A_t$ is higher than $\frac{1}{N}$. Theorem 7 indicates that there are also L-P states which lead to the contrary event, i.e $z_t = z_{t+1}$, this is an important reason for higher $Pr(B_t|\bar{A}_t)$. Let C_t denotes the event $j_t = t + 1$, $s_t[t + 1] = -1$ and $s_t[t] \neq t + 1, t + 2$. If we assume that when the events A_t and C_t do not happen, then B_t happens with trivial probability $1/N$.
Then B_t(t$\neq$ -1,-2) can be computed as

$$\begin{aligned} Pr(B_t) =& Pr(B_t|A_t)Pr(A_t) + Pr(B_t|C_t)Pr(B_t|C_t) + Pr(B_t|\overline{A_t \cup C_t})Pr(\overline{A_t \cup C_t}) \\ =& 0 + \frac{1}{N^2}(1 - \frac{2}{N}) + \frac{1}{N}[1 - \frac{1}{N^2} - \frac{1}{N^2}(1 - \frac{2}{N})] \\ =& \frac{1}{N}(1 + \frac{1}{N} - \frac{4}{N^2} + \frac{2}{N^3}) \end{aligned} \tag{4}$$

which is larger than $\frac{1}{N}$. Certainly, it is possible that calculating $Pr(B_t)$ by (4) is also not accurate, since the the existence of other L-P states which is related to B_t may deny the assumption that $Pr(B_t|\overline{A_t \cup C_t}) = \frac{1}{N}$. However, (4) does explain why a single L-P state is not enough to ensure a long term bias. Notice the fact that the probability of A_t and C_t are nearly the same, what if the probability of C_t is much smaller than A_t? We start our analysis by the following definition.

Definition 4. *Let A_t be an external event with prior probability of $N^{-a}(a > 0)$, B_t and C_t are internal events with B_t predicts A_t, C_t predicts $\bar{A}_t$. Then* ***the estimate of*** *A_t* ***by*** *B_t* ***and*** *C_t is*

$$Pr(B_t) + N^{-a}(1 - Pr(B_t) - Pr(C_t)).$$

Especially, when $B_t = \emptyset$, the estimate is $N^{-a}(1 - Pr(C_t))$. When $C_t = \emptyset$, the estimate is $Pr(B_t) + N^{-a}(1 - Pr(B_t))$.

For an external event A_t, if a internal event B_t with B_t predicts A_t is found, whether $Pr(A_t)$ is the estimate of A_t by B_t depends on $Pr(A_t|\bar{B}_t)$, the estimate is accurate when $Pr(A_t|\bar{B}_t) = N^{-a}$. However, a internal event C_t will lead to higher $Pr(A_t|\bar{B}_t)$ when C_t predicts A_t, and lower when C_t predicts $\bar{B}_t$. It is well known that when $Pr(A_t)$ is different from the prior probability N^{-a}, i.e, a bias exits, a distinguishing attack can be applied to RC4, the data complexity of the attack depends on the bias.

Definition 5. *Let $0 < p_1, p_2 < 1$, A_t be an external event , then we say p_1* ***and*** *p_2* ***are equivalent for*** *A_t if the data complexity of distinguishing attack by A_t with $Pr(A_t) = p_1$ is the same*[3] *as by A_t with $Pr(A_t) = p_2$.*

Intuitively, when the internal event C_t contributes little to $Pr(A_t)$, it is reasonable to be neglected.

Theorem 9. *Let A_t be an external event of RC4 keystream with prior probability of N^{-a}, B_t be a internal event which predicts A_t with $Pr(B_t) = N^{-b}$, C_t an internal event which predicts A_t or $\bar{A}_t$ with $Pr(C_t) = N^{-c}$, $a, b, c > 0$, $B_t \cap C_t = \emptyset$. Then the estimate of A_t by B_t is equivalent to the estimate of A_t by B_t and C_t for A_t when $c > b$.*

In order to prove Theorem 9, we present two lemmas. The proof of Lemma 1 can be found in [2].

Lemma 1. *If event e occurs in a distribution X with probability p and in Y with probability $p(1 + q)$. Then, for small p and q, $O(1/pq^2)$ samples are required to distinguish X from Y with non-negligible probability of success.*

Lemma 2. *Let A_t be an external event with prior probability p, q is positive real number satisfied $q = O(N^{-s})$, then $p(1+q)$ and $p(1+N^{-s})$ are equivalent for A_t.*

Proof. By Lemma 1, the time complexity of distinguishing the RC4 keystream from random is $O(pq^2)$.

$$\begin{aligned} O(pq^2) =& O(p * (N^{-s} + o(N^{-s}))^2) \\ =& O(p * (N^{-2s} + 2 * N^{-s} * o(N^{-s}) + (o(N^{-s})^2))) \\ =& O(p * N^{-2s}) = O(p * (N^{-s})^2) \end{aligned}$$

□

[3]Where X and Y is the same means X∼O(Y) by N.

The proof of Theorem 9:
By Definition 4, the estimate of A_t by B_t is

$$Pr(B_t) + N^{-a}(1 - Pr(B_t)) = N^{-a}(1 + N^{a-b} - N^{-b}).$$

We will discuss the problem in two cases

- C_t predicts A_t.

  Let $B\prime_t = B_t \cup C_t$. Since $B_t \cap C_t = \emptyset$, $Pr(B\prime_t) = Pr(B_t) + Pr(C_t) = N^{-b} + N^{-c}$. The estimate of A_t by $B\prime_t$ is

$$\begin{aligned} Pr(A_t) =& Pr(B\prime_t) + N^{-a}(1 - Pr(B\prime_t)) \\ =& N^{-a}(1 + N^{a-b} + N^{a-c} - N^{-b} - N^{-c}) \end{aligned}$$

  Since $\frac{N^{a-c}-N^{-c}}{N^{a-b}-N^{-b}} = N^{b-c} \to 0$,
  $N^{a-b} + N^{a-c} - N^{-b} - N^{-c} = O(N^{a-b} - N^{-b})$.

- C_t predicts $\bar{A}_t$, then the estimate of A_t by B_t and C_t is

$$\begin{aligned} Pr(A_t) =& Pr(B_t) + N^{-a}(1 - Pr(B_t) - Pr(C_t)) \\ =& N^{-a}(1 + N^{a-b} - N^{-b} - N^{-c}) \end{aligned}$$

  Since $\frac{N^{-c}}{N^{a-b}-N^{-b}} = \frac{1}{N^{c-b}(N^a-1)} \to 0$,
  $N^{a-b} - N^{-b} - N^{-c} = O(N^{a-b} - N^{-b})$.

Then by Lemma 2, we complete the proof.

Notice that the external event we consider in this theorem are the events with probability of N^{-a}, there are also events which do not satisfy the condition, like $z_t = z_{t+1}$. In this situation, we can consider the complement event instead.

6.3 Compute Possible Long Term Bias

For a L-P state B_t, assume B_t predicts the external event A_t, in order to calculate the probability of A_t, all the L-P states which predicts A_t or $\bar{A}_t$ have to be considered. Fortunately, theorem 9 implies that the L-P state C_t with $Pr(C_t) = o(Pr(B_t)/N)$ which predicts A_t or $\bar{A}_t$ can be ignored, let alone $Pr(A_t|C_t)$ is just positive or negative biased.

The L-P states are constructed with the current value j_t, several states of the current array s_t. Our algorithm takes the L-P state B_t and the external state A_t predicted by B_t as input, and outputs all the L-P states that can not be neglected as output, first we compute the probability of B_t, if $N^{-a-1} < Pr(B_t) < N^{-a}$, then only the L-P states with determined elements smaller than a will be considered. The algorithm does an exhaustive research on these non-neglect L-P states of the form $s_t[\sigma_m(t)] = \xi_m(t)$, $m{=}1 \cdots a$,

$j = \varphi(t)$, since the index i is increment gradually , it is reasonable to assume $\sigma_m(t)$, $\varphi_m(t)$ and $\xi_m(t)$ as linear functions. If not, the known element will be swapped to an unknown position $s_t[t]$ ar round t.

Proposition 1. *Only the L-P states with linear functions $\sigma_m(t)$, $\varphi_m(t)$ and $\xi_m(t)$ need to be considered when computing the probability of external event B_t, where the signal $\sigma_m(t)$, $\varphi_m(t)$ and $\xi_m(t)$ are defined in Definition 2.*

There are states which predictive external events at later rounds, i.e., L-P state A_{t-m}, $m > 0$, may predict the event B_t at time t, we suppose that the probability becomes smaller when m increase. In our algorithm, we denote θ the upper bound of m.

Our algorithm for searching the non-neglect L-P states is present in the next page. The time complexity of the algorithm is $O(N^{4a+5})$. It is necessary to point out that the forbidden states of RC4 should be dumped from the output states, where the forbidden states are states like $j_t = t + 1$, $s_t[t+1] = 1[4]$. Once all the non-neglect L-P states are found, we can obtain $\xi_t(\eta_t)$, which is the union of all the non-neglect L-P states lead to $A_t(\bar{A}_t)$. It is reasonable to estimate $Pr(A_t|\overline{\xi_t \cup \eta_t})$ by the prior probability q since other events have neglect influence on the bias. So the probability of A_t can be calculated as follows:

$$\begin{aligned} Pr(A_t) =& Pr(A_t|\xi_t)Pr(\xi_t) + Pr(A_t|\eta_t)Pr(\eta_t) + q * Pr(A_t|\overline{\xi_t \cup \eta_t}) \\ =& q(1 + (q^{-1} - 1)Pr(\xi_t) - Pr(\eta_t)). \end{aligned}$$

Algorithm 1 L-P states for external event A_t

Require: A_t and the L-P state B_t which predicts A_t.

1. $\mathscr{A}$ runs the algorithm of RC4 which takes temple key, or the current state as input. i.e., $\mathscr{A}(key)$ or $\mathscr{A}(s_t, j_t)$. And run the RC4 algorithm, when given a round number t, it outputs the state s_t , j_t and the keystream z_t.
2. a is the upper bound of the number of elements of the L-P state.

compute $a = \lceil -log_N Pr(A_t) \rceil$.
for number=0 **to** a-1 **do**
  set count[θ]={0},
  for all the tuples (a_i^0, a_i^1, b_i^0, b_i^1) $\in Z_N^4$, i=0$\cdots$ number-1 and (c,d)$\in$ Z_N^2. **do**
    generate N temple keys $\{key_n\}_{n=0}^{N-1}$.
    for n=0 **to** N-1 **do**
      for t=0 **to** N-1 **do**
        run $\mathscr{A}(key_n)$ until it outputs s_t and j_t;
        j_t=$ct + d$;
        for i=0 to number-1 **do**
          if $s_t[a_i^0 t + a_i^1] \neq b_i^0 t + b_i^1$ **then**
            find m_i satisfied $s_t[m_i] = b_i^0 t + b_i^1$;
            swap($s_t[m_i]$, $s_t[a_i^0 t + a_i^1]$);
          end if
        end for
        for k=0 **to** $\theta - 1$ **do**
          keep on running $\mathscr{A}(s_t, j_t)$ until it outputs event A_{t+k} or $\overline{A_{t+k}}$.
          if A_{t+k} is outputted **then**
            count[k]++;
          end if
        end for
      end for
    end for
    if count[k]$\geq N^2$-cN or count[k]$\leq cN$, (where c is set to be a small constant) **then**
      output the current tuple, (c, d), and k.
    end if
    reset count[θ] to {0}.
  end for
end for

7 Conclusions and Open Problems

In this paper, we conclude previous work on predictive states, analysis these states, and give out the concept of L-P states. What's more, we analysis the properties of the L-P states, give our new L-P states which will predict a special output or equal adjacent outputs. At the end, we propose an algorithm to search for some special L-P states, which will contribute to the long term bias. However, a L-P state must be found first for the algorithm, so the way to find the initial L-P state is a question needs to be considered. In our algorithm, the L-P state which satisfied $count[k] \geqslant N^2 - cN$ or $count[k] \leqslant cN$ is output, maybe the range can be wider since C_t with $Pr(A_t|C_t)$ is positive or negative biased can be considered. What's more,

maybe the conditions when σ_m, φ_m and ξ_m mentioned in Theorem 9 are not linear functions should be considered.

References

[1] A.Maximov and D.Khovratovich, "New state recovery attack on RC4" *Crypto'2008* LNCS vol.5157, pp.297-316, 2008.

[2] I.Mantin, A.Shamir "A practical Attack on Broadcast RC4" *Fast Software Encryption-FSE'2001* LNCS vol. 2355, pp. 152-164, 2002.

[3] I.Mantin "Predicting and distinguishing attacks on RC4 keystream Generator" *Advances in Cryptology-Eurocrypt'2005*, LNCS vol. 3494, pp. 491-506, 2005.

[4] H.Finney "An RC4 cycle that can't happen" September, 1994.

[5] J.Lv, B.Zhang "Distinguishing attacks on RC4 and a new improvement of the cipher". *Cryptology ePrint Archive*, Report 2013/176.

[6] R.Knudsen and W.Meier "Analysis methods for(Alleged) RC4 " *Advances in Cryptology-asiacrypt'1998* LNCS vol.1514, pp.327-341, Springer-Verlag, 1998.

[7] S.Paul, B.Preneel " Analysis of Non-fortuitions Preditive states of the RC4 keystream Generator" . *Progress in Cryptology-Indocrypt'2003.*LNCS vol 2904,2003,pp. 52-67, Springer-Verlag, 2003.

[8] S.Paul, B.Preneel "A New weakness in the RC4 keystream generator and an approach to improve the security of the cipher," *Fast Software Encryption-FSE'2004* LNCS vol. 3017, pp. 245-259, Springer-Verlag, 2004.

[9] T.Isobe and T.Ohigashi "Full plaintext recovery attack on broadcast RC4 " *Fast Software Encryption-FSE'2013*