Reducing the State Space of RC4 Stream Cipher*

Violeta Tomašević[1] and Slobodan Bojanić[2]

[1] Institute "Mihajlo Pupin", Volgina 15, 11050 Belgrade, Serbia and Montenegro
violeta@impcs.com, http://www.imp.bg.ac.yu

[2] Technical University of Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain
slobodan@die.upm.es

Abstract. The paper introduces an abstraction in form of general conditions for cryptanalytic managing of the information about the current state of the RC4 stream cipher. The general conditions based strategy is used to favor more promising values that should be assigned to unknown entries in the RC4 table. The estimated complexity of the cryptanalytic attack is lower than the best published result although the RC4 remains a quite secure cipher in practice.

1 Introduction

Based on the table-shuffling principle, the alleged RC4 stream cipher is designed for the fast software implementation and widely used in many commercial products and standards [1]. The RC4 cryptanalysis [2] has been mainly devoted to the statistical analysis of the output sequence [3], [4], or to the initialization weaknesses [5]-[7]. The most important results [8, 9] exploit the combinatorial nature of RC4. Although without a practical threat these attacks could be used in completing the internal state of the cipher, given some additional information [10]. They track the RC4 steps and assign the values to unknown table entries. The values are assigned one after one, thus some of them are favored without a reason. Since the number of the assignments until reaching the solution determines the complexity of attack, some improved strategy of selecting the values to be assigned could be very useful.

We propose the tree representation of the RC4 algorithm with the set of trees corresponding to each output symbol. The nodes and the branches encompass all information at given time. However, since the trees progressively increase, we could not practically exploit all available information. This problem is imminent for all other attacks, too. Therefore, we introduce an analytical abstraction, named *the general conditions* to represent all information from a subtree. We defined the examination strategy that favors the choice of the values by reordering the set of unassigned values. In addition to that, for each general condition, the probability that leads to the solution has been found, thus the most probable values are favored.

* This work has been partially supported by the Ministries of Science and Technology of Serbia (# IT.1.24.0041) and Spain (# TIC2003-09061-C03-02 and the "Ramon y Cajal" program).

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 644–647, 2004.

2 General Conditions

The RC4 is a family of algorithms parameterized by a positive integer n (usually n = 8). The RC4 internal state at time t consists of a permutation table S_t of 2^n different n-bit values and of two n-bit pointers i_t and j_t [1]. The output n-bit symbol Z_t is:

$$Z_t = S_t(L_t), \qquad L_t = S_t(t) + S_t(j_t) . \tag{1}$$

The content of the S_t table can be given by:

$$S_t(t) = S_{t-1}(j_t) . \tag{2}$$

$$S_t(k \neq t) = S_{t-1}(k) , k \neq j_t \tag{3}$$

$$S_t(k \neq t) = S_{t-1}(t) , k = j_t. \tag{4}$$

Applying equations (2)-(4), there follows that equation (1) can be represented by the tree structure shown in Fig. 1.

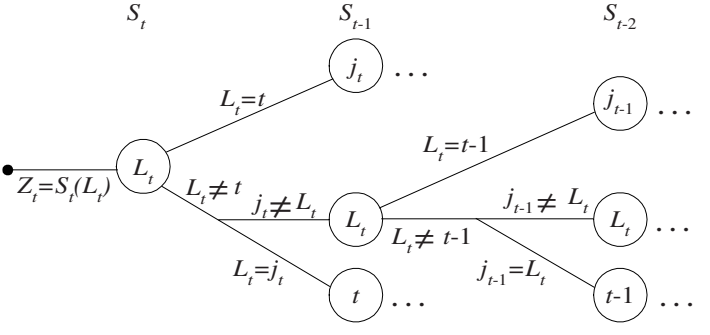

Fig. 1. Tree representation of the RC4 algorithm

Combining algorithm's equations, the whole sub-trees whose roots are given by the expressions like $S_{k-1}(k)$ and $S_{k-1}(j_k)$, can be described by more general conditions *C1…C5* (Table 1) that can be checked instead of all conditions from the sub-tree. The *C3* and *C4* conditions both encompass t-1 conditions. The expression $e(Z_t)$ denotes the position of Z_t in the initial table. All calculations are modulo 2^n. By introducing the general conditions, we neglect some information, but significantly decrease the number of conditions that should be checked.

The probabilities that general conditions lead to the solution can be determined by multiplying the known probabilities of conditions on the path to the given node (Table 2). This fact inspired our further proposal intended to achieve an additional reduction of the search space by observing the probabilities of the general conditions. Consequently, these conditions should be checked in decreasing order of their probabilities while the existing approaches are based on arbitrary guessing.

Table 1. General conditions

C1	*C2*	*C3*($p\in[2,t]$)	*C4*($p\in[2,t]$)	*C5*
$j_t=t-Z_t+j_{t-1}$	$j_t=Z_t+j_{t-1}$	$Z_t=j_{t-p+1}-j_{t-p}$	$Z_t=L_{t-p+1}-j_{t-p+1}+j_{t-p}$	$L_t\neq t,..,1,j_t,..,j_1$
$L_t=t$	$L_t=j_t\neq t$	$L_t=j_{t-p+1}\neq t,..,t-p+1,j_t,..,j_{t-p+2}$	$L_t=t-p+1\neq j_t,..,j_{t-p+2}$	$L_t=e(Z_t)$
$S_{t-1}(t)=t-Z_t$	$S_{t-1}(t)=Z_t$	$S_{t-p}(t-p+1)=Z_t$	$S_{t-p+1}(t-p+1)=Z_t$	$S_{t-1}(j_t)=e(Z_t)-j_t+j_{t-1}$
$S_{t-1}(j_t)=Z_t$	$S_{t-1}(j_t)=j_{t-1}$	$S_{t-1}(j_t)=j_{t-p+1}-j_t+j_{t-1}$ $S_{t-1}(t)=j_t-j_{t-1}$	$S_{t-1}(t)=j_t-j_{t-1}$ $S_{t-1}(j_t)=t-p+1-j_t+j_{t-1}$	$S_{t-1}(t)=j_t-j_{t-1}$

Table 2. Probabilities of the general conditions

C1	*C2*	*C3*($p\in[2,t]$)	*C4*($p\in[2,t]$)	*C5*
$1/2^n$	$(2^n-1)/2^{2n}$	$(2^n-1)^{p-1}(2^n-p)/2^{n(p+1)}$	$(2^n-1)^{p-1}/2^{np}$	$(2^n-1)^t(2^n-t)/2^{n(t+1)}$

3 Efficiency of the Attack

Our approach is to enhance the RC4 cryptanalytic algorithm given in [8] with general conditions examination. Going through the RC4 steps, our algorithm determines the values of entries in the table that have not already been assigned using the general conditions, in order to ensure the next state update of RC4. In the case of contradiction, the backtracking proceeds. Such algorithm with general conditions applies the basic principle of the hill-climbing search strategy [11].

The complexity of the attack given in [8] is measured by total number of the assignments made for all entries of the initial table until the solution is found. Similarly, in order to calculate the complexity of our algorithm, three functions $c_i(a)$, $i = 1, 2, 3$ are defined. For n-bit RC4, it is assumed that at given time t there are a previously assigned values in the table.

$$c_1(a)=(a/2^n)c_2(a)+(1-a/2^n)[(v(C1)+v(C2))c_1(a+1)+ +(v(C5)+\sum(\mathrm{v(C3(p))}+\mathrm{v(C4(p))}))(2^n-a-(1-a/2^n))c_2(a+1)]$$
$$c_2(a)=(1-a/2^n)c_3(a)+(a/2^n)[(a/2^n)(1/2^n)c_1(a)+(1-a/2^n)(a/2^n+(1-a/2^n)c_1(a+1))]$$
$$c_3(a)=(1-a/2^n)[(1-a/2^n)(2^n-a-(1-a/2^n))+ +(a/2^n)(v(C1)+v(C5)(2^n-a-(1-a/2^n)))((a+1)/2^n+(1-(a+1)/2^n)c_1(a+2))$$

The calculation of complexity of our attack starts with the known expressions $c_i(2^n)=0$. Then, going backwards, we calculate total number of the assignments given by $c_1(0)$. The results of the calculation for the RC4 versions with different word size n are presented in Table 3. Compared to the estimated complexity of the original algorithm, an improvement has been made definitely, yet not enough to practically menace the security of the alleged RC4 stream cipher.

Table 3. Complexity of the cryptanalytic attacks on n-bit RC4

Word size n	3	4	5	6	7	8
Knudsen et al. [8]	2^8	2^{21}	2^{53}	2^{132}	2^{324}	2^{779}
Our attack	2^5	2^{17}	2^{46}	2^{120}	2^{300}	2^{731}

4 Conclusions

We proposed a new technique to improve the cryptanalytic attack on the RC4 cipher. It is based on new information from the tree representation of the RC4 algorithm. To make a better choice for the assignment to unknown entries of the cipher's table, we represented analytically similar nodes and corresponding sub-trees by means of general conditions. Then, we defined a search strategy which uses the information derived from the general conditions, determined the probabilities that they lead to the solution, and incorporated it into the algorithm [8]. The complexity was estimated by an analytical calculation for different values of n. The results show that this complexity is lower than the best known result [8], although the RC4 remains a quite secure cipher for the practical applications. Our research is an additional argument which advocates for the security of the RC4 cipher.

References

1. Schneier, B.: Applied Cryptography. Willey, New York, (1996).
2. Tomašević, V., Bojanić, S., O. Nieto-Taladriz: On the Cryptanalysis of Alleged RC4 Stream Cipher. In C. Anias et al. (eds.): Telematics. Edit. Univ. F. Varela, Havana, (2002) 227-232.
3. Golić, J.: Linear Statistical Weakness of Alleged RC4 Keystream Generator. In: Advances in Cryptology - EUROCRYPT '97. LNCS, Vol. 1233, Springer-Verlag, (1997) 226–238.
4. Fluhrer, S., McGrew, D.: Statistical Analysis of the Alleged RC4 Keystream Generator. In: Fast Software Encryption - FSE 2000. LNCS, Vol. 1978, Springer-Verlag, (2000) 19–30.
5. Roos, A.: A Class of Weak Keys in the RC4 Stream Cipher. Sci.crypt. September 1995.
6. Grosul, A., Wallach, D.: A Related-Key Cryptanalysis of RC4. TR00-358, Rice University, October 2000.
7. Fluhrer, S., Mantin, I., Shamir, A.: Weakness in the Key Scheduling Algorithm of RC4, Selected Areas in Cryptography – SAC 2001. Vol. 2259, Springer-Verlag, (2001) 1–24.
8. Knudsen, L., Meier, W., Preneel, B., Rijmen, V., Verdoolaege, S.: Analysis Methods for (Alleged) RC4. In: ASIACRYPT '98. LNCS Vol. 1514, Springer-Verlag, (1998).
9. Mister, S., Tavares, S.: Cryptanalysis of RC4-like Ciphers, Selected Areas in Cryptography - SAC '98. Springer-Verlag, (1998) 136-148.
10. Mantin, I., Shamir, A.: A Practical Attack on Broadcast RC4. Fast Software Encryption FSE, 2001, LNCS Vol. 2355, Springer-Verlag (2002) 152-164.
11. Pearl, J.: Heuristics, Addison Wesley Publishing Company, 1984.