对RC4算法的错误引入攻击研究

杜育松[1]，沈 静[2]

(1. 广州大学数学与信息科学学院 广州 510006; 2. 广州大学华软软件学院 广州 510990)

【摘要】错误引入攻击假设攻击者可以向密码设备(智能卡)引入错误，使其出现错误的加密结果。攻击者同时利用正确的和错误的加密结果来发现隐藏在密码设备中的秘密信息(密钥)。该文给出了一种对RC4算法的错误引入攻击方法。模拟实验表明，一轮攻击有可能找出RC4初始状态中3个位置的值，连续使用该算法能以较高(大于1/2)的概率恢复RC4的整个初始状态。恢复整个初始状态所需的密钥字个数约为$O(2^{16})$，引入的错误数量约为$O(2^{16})$。

关 键 词 差分错误分析; 错误引入攻击; RC4; 秘密信息

中图分类号 TP309 **文献标识码** A doi: 10. 3969/j. issn. 1001-0548. 2009. 02. 23

Research on Fault Induction Attack on RC4 Algorithm

DU Yu-song[1] and SHEN Jing[2]

(1. School of Mathematics and Information Science, Guangzhou University Guangzhou 510006;

2. Huaruan College of Software, Guangzhou University Guangzhou 510990)

Abstract Fault induction attack assumes that the attacker is able to induct faults into the cryptographic device (smartcard) and make it output incorrect encryption results. The attacker exploits the correct and incorrect encryption results to disclose the secret information (key) hidden in the cryptographic device. A method of the fault induction attack on RC4 algorithm is brought forward. The simulations show that one attack may find the values of 3 positions in the initial state of RC4 and continuous attacks can recover the whole initial state of RC4 with a considerable probability (more than 1/2). About $O(2^{16})$ key stream bytes at most are needed to recover the whole initial state of RC4 after about $O(2^{16})$ fault inductions at most.

Key words differential fault analysis; fault induction attack; RC4; secret information

错误引入攻击[1-4]假设攻击者控制了密码设备，可以正确地使用密码设备进行加密，还可以向密码设备中引入错误，影响加密过程，使密码设备输出错误的加密结果。攻击者的目标是利用错误的加密结果找出隐藏在密码设备中的秘密信息，并且可以利用正确的和错误的加密结果进行分析比较，这样的攻击也被称为差分错误分析[2,5]。

有关RC4密码分析的研究比较丰富[6-7]，但到目前为止，对RC4的错误引入攻击的研究并不多见。文献[8]在研究对流密码的错误引入攻击时包含了对RC4的研究，其结果表明需要2^{26}个密钥字和2^{16}次错误引入方可恢复RC4的初始状态。文献[9]继文献[10]之后专门研究了对RC4的错误引入攻击，利用“RC4中的不可能状态”提出了不可能错误分析和差分错误分析方法。恢复RC4的初始状态前者需要2^{21}个密钥字和2^{16}次错误引入，而后者则需要2^{16}个密钥字和2^{10}次错误引入。本文给出一种新的对RC4的差分错误分析方法。模拟实验表明，该方法能以较高的概率恢复RC4的整个初始状态。

1 RC4算法

RC4是以分组长度n(通常n=8)为参数的一类二元加法流密码体制[10-11]。它包含密钥调度算法和伪随机数生成算法。

(1) RC4的密钥调度算法如下：

Initialization:

$S_0[i]=i\ (i=0,1,2,\cdots,2^n-1)$

$i_0=0,\ j_0=0$

Scrambling:

$j_t=j_{t-1}+S_{t-1}[i_{t-1}]+K[i_{t-1}\bmod l]$

$S_t[i_t]=S_{t-1}[j_t],\ S_t[j_t]=S_{t-1}[i_t]$

$i_t=i_{t-1}+1$

where $t=1,2,\cdots,2^n$

它以一个l字长的初始密钥K作为输入，根据初

收稿日期：2007 – 11 – 08；修回日期：2008 – 05 – 23

基金项目：国家自然科学基金(10871222)

作者简介：杜育松(1982 –)，男，博士，主要从事密码学方面的研究.

始密钥K连续交换S盒中的值，产生一个新的依赖于K的初始状态。

(2) RC4的伪随机数生成算法如下：

Initialization:

$i_0 = 0$

$j_0 = 0$

Generation Loop:

$i_t = i_{t-1} + 1$

$j_t = j_{t-1} + S_{t-1}[i_t]$

$S_t[i_t] = S_{t-1}[j_t], S_t[j_t] = S_{t-1}[i_t]$

$Z_t = S_t[S_t[i_t] + S_t[j_t]]$

它不断地改变S盒中的置换，并从S盒中选择一个值作为输出，即一个n比特的密钥字。

2 RC4的差分错误分析

攻击基于“暂时随机”的字节错误。“随机”的字节错误是指在加密的中间结果中仅引入一个字节的错误，即只有一个字节发生了变化，而其他字节保持不变，并且认为改变后字节值在0～255之间均匀分布。“暂时”是指发生的字节错误将很快消失，只会使得密码设备当前运算出错，而不会影响到下一次的加密。这种错误类型与文献[4]中提到的相同。

假设攻击者能够向给定时刻的S盒的任意位置引入字节错误。具体地说，即可以在i和j已更新，且在执行交换操作之前向S盒中的任意位置引入字节错误。在这一假设下，本文给出一种新的对流密码体制RC4的差分错误分析方法(算法1)，利用该算法有可能找出RC4初始状态中3个位置的值。

为了方便地描述算法，本文引入以下符号：

Z_t：正确运行RC4后的第t个输出密钥字；

Z_t'：引入错误后得到的第t个输出密钥字；

Normal_Run：正确运行RC4算法一次，记录足够多的输出密钥字；

Fault_Run $T(t)$：重新运行RC4算法，在i_t和j_t已更新为i_{t+1}和j_{t+1}，且交换操作Swap($S_t[i_{t+1}]$和$S_t[j_{t+1}]$)执行之前向S盒的第t个位置引入错误，记录第一个错误的输出时刻T'；

Fault_Run (t,n)：重新运行RC4算法，在i_{t-1}和j_{t-1}已更新为i_t和j_t，且交换操作Swap($S_{t-1}[i_t]$和$S_{t-1}[j_t]$)执行之前向S盒的第n个位置引入错误，记引入错误后得到的第t个输出密钥字为Z_t'和正确的第n个输出密钥字为Z_t。

新的对流密码体制RC4的差分错误分析方法(算法1)的步骤描述如下：

1) Normal_Run；

2) Fault_Run T (1)；

3) $n \leftarrow 0$；

4) IF(n=1) THEN $n \leftarrow n+1$；ELSE goto 5；

5) Fault_Run (1,n)；

6) IF ($Z_1' \neq Z_1$) THEN

(1) 继续运行步骤5)中引入的错误的RC4算法，记录下一个不正确的输出时刻T；

(2) IF (T=T') THEN {S_0[1]$\leftarrow n$；$j_1 \leftarrow n$；type_$A$$\leftarrow$1}；

(3) IF ($T \neq T'$) THEN {$S_0[n] \leftarrow Z_1$; temp$\leftarrow n$; type_$B$$\leftarrow$1}；

(4) IF (type_A=1&& type_B=1) THEN {$S_0[j_1] \leftarrow$ temp$-S_0$[1]；goto 7}；

(5) IF(type_A=1&& type_B=0&& n=256) THEN {IF(S_0[1]= Z_1) THEN {$S_0[j_1] \leftarrow j_1 - S_0$[1]; goto 7} ELSE {$S_0[j_1]$= Z_1; goto 7}}；

(6) IF(type_A=0&& type_B=1&& n=256) THEN S_0[1]$\leftarrow$1; goto 7}；

(7) IF(n<256) THEN {$n \leftarrow n+1$; goto 4}；

(8) ELSE {$n \leftarrow n+1$; goto 4}；

END

算法1的可行性描述如下：由于攻击者控制了密码设备，所以它以一串空字符串为明文进行加密就可以得到一串正确的RC4输出密钥字。

在向S盒的第n个位置引入错误后，如果得到的第一个输出密钥字 Z_1'和正确的第一个输出密钥字Z_1相同，通常可认为第一个输出密钥字的产生与S盒的第n个位置无关，也即j_1和$S_1[i_1]+S_1[j_1]$都不会是n。此时，算法会向下一个位置n+1(除1外)再次引入错误。相反地，如果引入错误后得到的第一个输出密钥字 Z_1'和正确的第一个输出密钥字Z_1不同，说明第一个输出密钥字的产生与S盒的第n个位置一定有关，或者$j_1=n$，或者$S_1[i_1]+S_1[j_1]= S_0[i_1]+S_0[j_1]=n$，而判断$n$是否等于1，可保证错误不会被引入到第$i_1$个位置，如图1所示。

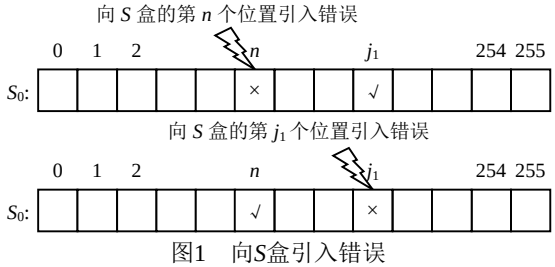

图1 向S盒引入错误

本文把发生在第j_1个位置的错误称为A类型错

误，把发生在第$S_1[i_1]+S_1[j_1]$个位置的错误称为B类型错误。算法下一步的目的是判断引入的错误属于哪一种类型。

判断的方法基于：(1) 如果发生的是A类型错误，那么在执行了Swap($S_0[i_1]$, $S_0[j_1]$)之后，错误将会被交换到$S_1[i_1]$中，而S_1中的其他位置的值保持正确；(2) 如果重新运行RC4算法，并在i_1和j_1已更新为i_2和j_2后，且交换操作Swap($S_1[i_2]$,$S_1[j_2]$)执行之前向S盒的第1个位置引入错误，将把S_1带入到与上述类似的错误状态中，即在S_1中只有第i_1个位置是错误的，而其他位置的值保持正确。分别继续执行两个错误状态下的RC4算法，它们下一次分别输出错误密钥字的时刻很可能相同。可以看到，如果发生的是B类型错误，那么在进行了同样的步骤后，它们分别输出错误密钥字的时刻很可能不同，如图2所示。

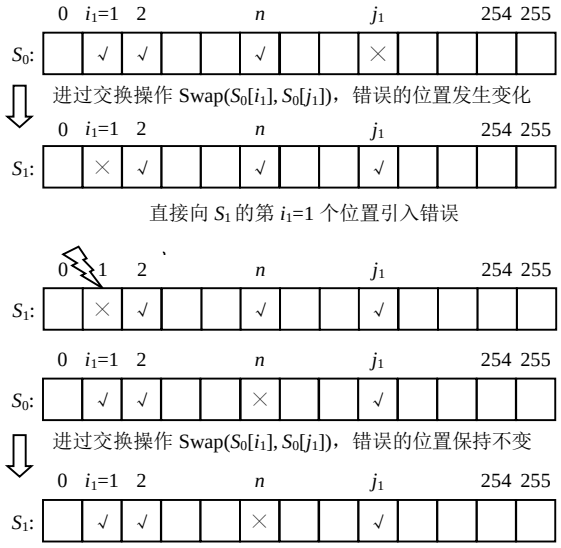

图2　判断错误类型

算法1利用上述事实进行判断。如果输出错误密钥字的时刻相同，那么认为发生了A类型错误，反之认为发生了B类型错误。注意用以上方法得到的判断结果是概率型的，即判断结果也有可能不正确，但是判断不正确的概率不大。这一点可以在本文的软件模拟结果中看到。

如果得到的判断结果正确，就可以成功地找出初始状态的几个值。事实上，当发生A类型错误时，因为S_0的第j_1个位置就是引入错误的位置，因此有$j_1=n$，而$j_1=S_0[1]$，所以又有$S_0[1]=n$；当发生B类型错误时，S_0的第$S_1[i_1]+S_1[j_1]$个位置就是引入错误的位置，因为该位置的值本应是第一个输出密钥字Z_1，因此有$S_0[n]= Z_1$。利用已得到的$S_0[1]$和$S_0[n]$，还可以进一步推算出$S_0[j_1]$。

算法1最后还分析了在连续地255次错误引入后A类型错误或B类型错误不发生的情况。当A类型错误发生而B类型错误未发生，有两种可能的情况，一种是B类型与A类型错误重合，另一种是B类型错误发生在第1个位置。

3　恢复RC4的整个初始状态

在算法1的基础上可以给出恢复RC4的整个初始状态的算法(算法2)，该算法步骤描述如下：

1) Normal_Run

2) $t\leftarrow 1$；$J\leftarrow 0$；

3) 对每一个$i\in[0,255]$，置is_det $[i]\leftarrow 0$；

4) Fault_Run$T(t)$

5) $n\leftarrow 0$；

6) IF($n=t$) THEN $n\leftarrow n+1$，goto 7；ELSE goto 7；

7) Fault_Run(t, n)

8) IF($Z_t'=Z_t$) THEN {IF(n<256) THEN{$n\leftarrow n$+1;goto 6}；ELSE goto 9}；ELSE

(1) 继续运行步骤7)中引入的错误的RC4算法，记录下一个不正确的输出时刻T；

(2) IF($T=T'$) THEN{temp$\leftarrow(n-J)$；$J\leftarrow n$；type_A $\leftarrow 1$；IF(is_det[t]=0) THEN $S_0[t]\leftarrow(n-J)$}

(3) IF($T\neq T'$) THEN {temp2$\leftarrow n$；type_$B\leftarrow 1$；IF(is_det [n]=0) THEN $S_0[n]\leftarrow Z_t$;}

(4) IF(type_A=1&& type_B=1&& is_det [J]=0) THEN {$S_0[J]\leftarrow$(temp2−temp)；goto 9;}

(5) IF(type_A=1&& type_B=0&& n=256&& is_det [J]=0) THEN { IF(temp=Z_t) THEN {$S_0[J]\leftarrow(J-$ temp)；goto 9} ELSE {$S_0[J]=Z_t$; goto 9}

(6) IF(type_A=0&& type_B=1&& n=256&& is_det [t]=0) THEN {$S_0[t]\leftarrow(t-J)$；goto 9}

(7) IF(n<256) THEN {$n\leftarrow n$+1；goto 6}；

9) IF(t<256) THEN {$t\leftarrow t$+1；goto 4} ELSE END

算法2的本质是以t为循环参数，连续使用算法1的思想，在必要的时候对算法1进行修改，以尽可能地恢复RC4的整个初始状态。

算法2恢复的是初始状态S_0，而第t次使用算法1得到的几个值是S_t中的值，需要考虑的是S_t的这几个位置的值是否与S_0的相同。换句话说，需要判断RC4经过t轮运算后，S盒中的这几个位置是否发生过交换操作。判断这一点并不困难。事实上，如果每次使用算法1得到的结果正确，那么由算法1确定的几个值一定包含发生了交换操作的两个位置的值，所以，只要当前得到的S_t的这几个位置的值在前面t–1轮攻击中没有被确定过，则可以认为S_t的这几个位置

的值就是S_0的相应位置的值。为此，本文设置变量is_det[t]=1表示S的第t个位置已经确定。相反，is_det[t]=0表示S的第t个位置没有确定。另外，由于算法2连续使用了算法1，因此算法2也是概率型算法，即由算法2得到的结果可能不正确。

从所需的输出密钥字个数和引入的错误数量两个指标考察算法2的复杂度，不难推出：算法2所需的输出密钥字个数至多为$2^9+2^{16}+2^8+2^9=O(2^{16})$；需要引入的错误个数至多为$2^{16}+2^8=O(2^{16})$。

4 差分错误分析的软件模拟

本文用C语言编写程序，模拟实现了本文的两个攻击算法。令算法1重复运行10^7次，并且每次均匀随机地选取初始密钥，以统计算法1的攻击成功概率。运行发现，算法1攻击不成功的概率很小，未能正确找出3个相关的初始状态的次数仅为50 927。对于算法2，可选择不同的攻击轮数运行该算法。令每个循环参数重复运行算法10^4次，以统计算法2的攻击成功概率。运行结果如表1所示。

表1 算法2的模拟实现结果

攻击的轮数(t)	部分值成功概率/(%)	未确定位置数/个
50	80.0	139
80	73.9	93
100	69.5	71
110	67.9	61
130	62.4	45
150	59.3	33
180	57.3	19
210	55.2	9
220	54.5	6.570
230	49.3	4
240	55.2	2.479
256	52.1	0.122

表中“部分值成功概率”是指算法运行完毕后能以某个概率成功地获得RC4初始状态的部分位置的值，而且所获的值都是正确的。当进行256轮攻击后，部分值成功的概率达到52.1%，且初始状态S盒中未确定位置的平均个数仅为0.122，几乎找出了RC4所有的初始状态值。另外，攻击者也可以减少攻击轮数以获得更高的“部分值成功概率”和更低的复杂度，S盒中剩余的未确定位置则可以由状态猜测算法等其他攻击方法[6]得到。

注意到，文献[10]给出的对RC4一个差分错误分析方法只需要2^{10}次错误引入[10]。本文分析了该方法错误引入次数少的原因。一方面，该方法在一次错误引入后，多次利用了同一串密钥流进行分析，极大地减少了错误引入。另一方面，该方法以局部的状态猜测来判断错误类型，无需再引入错误，也减少了错误引入的次数。

然而，错误引入次数的减少也带来了一定的问题。首先，多次利用同一串密钥流进行分析，成功概率会降低很多。文献[10]也指出了这一点，并给出了几个解决这一问题的方法，但却没有详细说明，因而方法是不确定的。其次，如果以局部的状态猜测判断错误类型，判断正确的概率也会逐渐降低。而本文给出的判断错误类型的方法判断正确的概率是基本确定的。另外，文献[10]没有给出具体的模拟实现结果，真正的成功概率不得而知。

5 结 论

本文给出了一种新的对RC4的差分错误分析方法。分析要求攻击者能够向给定时刻的S盒的任意位置引入字节错误。模拟实验表明，一次分析有可能找出RC4初始状态中3个位置的值，连续使用该算法能以较高的概率恢复RC4的整个初始状态。最坏情况下，恢复整个初始状态所需的密钥字个数至多为$O(2^{16})$，引入的错误数量至多为$O(2^{16})$。

参 考 文 献

[1] BONEH D, DEMILLO R A, LIPTON R J. On the importance of checking cryptographic protocols for faults[C]//EUROCRYPT ’97. Berlin, Germany: Springer-Verlag, 1997: 37-51.

[2] BIHAM E, SHAMIR A. Differential fault analysis of secret key cryptosystems[C]//CRYPTO ’97. Berlin, Germany: Springer-Verlag, 1997: 513-525.

[3] PIRET G, QUISQUATER J J. A differential fault attack technique against SPN structures, with applications to the AES and Khazad[C]//CHES 2003. Berlin, Germany: Springer-Verlag, 2003: 77-88.

[4] CHONG H K, QUISQUATER J J. Faults, injection methods, and fault attacks[J]. IEEE Design and Test of Computers, 2007, 24(6): 544-555.

[5] WANG C, GU D, ZHU L. Formalization of fault analysis and a new approach of fault detection[J]. Journal of Shanghai Jiaotong University (Science English Version), 2006, 11(3): 301-306.

[6] KSNUDSEN L R, MEIER W, PRENEEL B, et al. Analysis methods for (alleged) RC4[C]//ASIACRYPT ’98, Berlin, Germany: Springer-Verlag, 1998: 327-341.

[7] MCKAGUE M E. Design and analysis of RC4-like stream ciphers[D]. Waterloo: University of Waterloo, 2005.

[8] HOCH J, SHAMIR A. Fault analysis of stream ciphers[C]//CHES 2004. Berlin, Germany: Springer-Verlag, 2004: 240-253.

[9] BIHAM E, GRANBOULAN L, NGUYEN P Q. Impossible fault analysis of RC4 and differential fault analysis of RC4[C]//FSE 2005. Berlin, Germany: Springer-Verlag, 2005: 359-367.
[10] SCHNEIER B. Applied cryptography protocols, algorithm, and source code in C[M]. 2nd. NJ, USA: John Wiley & Sons Inc, 1996.
[11] YANG J, ZHOU X, QIN B. On the selection of random numbers in the ElGamal algorithm[J]. Journal of Electronic Science and Technology of China, 2006, 4(1): 55-58.

编 辑 熊思亮

(上接第248页)

参 考 文 献

[1] LIU Hang, MAO Yu-ming. A wireless LAN bridging solution based on campus network[J]. Journal of Electronic Science and Technology of China. 2005, 3(1):14-17.
[2] CHANG Ben-jye, LIN Shu-yu, Liang Ying-hsin. Minimizing roaming overheads for vertical handoff in heterogeneous wireless mobile networks[C]// ACM IWCMC'06: Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing. Canada: ACM Press, 2006: 957-962.
[3] MUSTAFA N, MAHMOOD W, CHAUDHRY A A, et al. Pre-Scanning and dynamic caching for fast handoff at MAC layer in IEEE 802.11 wireless LANs[C]//Proc 2005 IEEE Int'l Conf Mobile Adhoc and Sensor Systems. Washington: IEEE Press, 2005: 114-122.
[4] MISHRA M S A, ARBAUGH W. An empirical analysis of the IEEE802.11 MAC layer handoff process[J]. ACM Sigcomm Computer Communi-cation Review, 2003, 33(2): 94-102.
[5] CHAUDHARY V, TRIPATHI R, SHUKLA N K. A new channel allocation scheme for real-time traffic in wireless cellular networks[C]// IPCCC2007: IEEE Performance, Computing, and Communications Conference. Marquette: IEEE Press, 2007: 551-555.
[6] MOHANTY S, AKYILDIZ I F. A cross-layer (Layer2+3) handoff management protocol for next-generation wireless systems[J]. IEEE Trans Mobile Computing, 2006, 5(10): 1347-1360.
[7] SHIN S, RAWAT A S, SCHULZRINNE H S. Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs[C]// ACM MobiWac'04: Proceedings of the Second International Workshop on Mobility management & wireless access protocols. USA: ACM Press, 2004: 19-26.
[8] TRIPATHI N. Generic adaptive handoff algorithms using fuzzy logic and neural networks[D].VA: VA Polytechnic Inst. and State Univ,1997.
[9] GUO Qiang, ZHU Jie, XU Xiang-hua. An adaptive multi-criteria vertical handoff decision algorithm for radio heterogenous network[C]// ICC2005: IEEE Int Conf Communication, Souel: IEEE Press, 2005: 2769-2773.
[10] Vivek Mhatre, Konstantina Papagiannaki,Using smart triggers for improved user performance in 802.11 wireless networks [C]//Proceedings of the 4th international Conference on Mobile Systems, Applications and Services. Sweden: ACM Press, 2006: 246-259.
[11] 刘乃安. 无线局域网(WLAN)—— 原理、技术与应用[M]. 西安: 西安电子科技大学出版社, 2004: 53-70.
LIU Nai-An. Wireless Local Area(WLAN) —principle, technique and application[M]. Xian: Xidian University Press, 2004: 53-70.
[12] 谢 峰. 地铁无线通信系统网络层移动性的研究与实现[D]. 南京: 南京航空航天大学, 2006.
XIE Feng. Research and implementation of mobility in network layerfor subway wireless communication system [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2006.

编 辑 张 俊