一种针对 Grain-v1 的新差分错误攻击

王 璐 胡予濮 张振广
(西安电子科技大学计算机网络与信息安全教育部重点实验室 西安 710071)

摘 要 通过分析流密码算法 Grain-v1,提出了一种针对密钥流生成器的差分错误攻击。该攻击利用了前17轮密钥流次数较低的弱点,向LFSR的指定位置引入错误,通过差分得到17个线性无关的线性方程和80个内部状态,只需要猜测62bits的初始内部状态变量就可得到密钥种子。整个过程的计算复杂度为 $O(2^{74.26})$。结果表明,Grain-v1 抗差分错误攻击的计算复杂度低于设计者宣称的 $O(2^{80})$,也就是说,算法存在安全漏洞。

关键词 流密码,Grain,差分错误攻击,密钥流
中图法分类号 TN918.1 **文献标识码** A

Differential Fault Analysis of Grain-v1

WANG Lu HU Yu-pu ZHANG Zhen-guang
(Key Laboratory of Computer Network and Information Security of Ministry of Education, Xidian University, Xi'an 710071, China)

Abstract By analyzing the weakness in design of the stream cipher Grain-v1, a differential fault attack was presented. The attack makes use of the weakness that the key stream equations in the first 17 times have comparatively low orders. The attacker needs to inject faults to the specified positions of LFSR at the stage of generating key stream. By differentiating, the attacker is able to acquire 17 linear equations which are linear independent and 80 initial states of the stream cipher directly. The attacker just needs to guess 62bits internal states, and then all the internal state can be achieved. The proposed attack algorithm can reduce the complexity to $O(2^{74.26})$. The result shows that the analyzed algorithm has security vulnerabilities, and the computational complexity of attacks is lower than that the designers claimed $O(2^{80})$.

Keywords Stream cipher, Grain, Differential fault attack, Key stream

eSTREAM项目是欧洲ECRYPT[1]的子项目。Grain算法作为eSTREAM项目的最终候选算法之一,是由瑞典学者M. Hell, T. Johansson和W. Meier共同提交的面向硬件实现的二进制同步流密码,其组件具有相应的功能。LFSR保证密钥流序列的周期长度和均衡性,NFSR和非线性布尔函数为密钥流序列引入非线性。Grain算法有3个版本:Grain v0, Grain v1[2]和Grain-v128[3]。其中,Grain v0和Grain v1的内部状态共80位,而Grain-v128内部状态共128位。迄今为止,针对Grain v0算法的主要攻击方法有区分攻击[4];针对Grain v1算法的主要攻击方法有相关密钥选择IV攻击[5]。

本文提出了一种针对Grain-v1的差分错误攻击。通过观察密钥流方程发现,由于LFSR和NFSR的更新函数未参与运算,在前16轮密钥生成方程中,方程的结构相同。本文采用在生成密钥流阶段开始时刻向指定位置引入错误,通过差分运算,将得到的方程与算法中的密钥流方程进行推导,得出内部状态信息。

本文第2节给出Grain算法的基本知识;第3节详细描述了所提的攻击方法,包括差分攻击的基础概念、攻击模型和主要步骤;第4节为计算复杂度分析;最后是结论。

1 Grain v1 简要介绍

Grain v1主要由3个部件构成:NFSR、LFSR、输出函数,具有160bits内部状态变量。算法如图1所示。

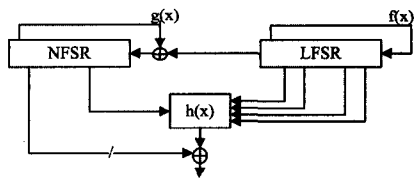

图1 Grain原理图

1.1 符号说明

本文中用到的符号如下:$b_{t+1}, b_{t+2}, \cdots, b_{t+80}, s_{t+1}, s_{t+2}, \cdots, s_{t+80}$,分别表示Grain中的NFSR、LFSR在$t$时刻的160个内部状态;用$\cdots Z_{-3} Z_{-2} Z_{-1} Z_0 Z_1 Z_2 Z_3 \cdots$表示$t$时刻得到的密钥流;"$*$"表示(mod 2)乘法,"$+$"表示(mod 2)加法,即异或运算。$Z_t$为$t$时刻未加入错误的密钥流,$Z_t'$为在某位置加入错误后$t$时刻生成的密钥流。$K$表示为0或1的常量。

1.2 LFSR

LFSR是80位的线性移位寄存器,运算是在域$GF(2)$上

到稿日期:2010-09-02 返修日期:2010-11-11 本文受国家自然科学基金(60833008)和国家973计划(2007CB311201)资助。
王 璐(1984—),男,硕士生,主要研究方向为流密码的分析与设计,E-mail:SagaWong@163.com;胡予濮 教授,博士生导师;张振广 硕士生。

进行，满足以下的本原多项式

$$f(x)=1+x^{18}+x^{29}+x^{42}+x^{57}+x^{67}+x^{80}$$

LFSR 状态更新函数

$$s_{t+80}=s_{t+62}+s_{t+51}+s_{t+38}+s_{t+23}+s_{t+13}+s_t \quad \forall t\geqslant 0$$

1.3 NFSR

NFSR 是 80 位的非线性移位寄存器，运算是在域 $GF(2)$ 上进行，满足以下的特征多项式

$$\begin{aligned}g(x)=1&+x^{18}+x^{20}+x^{28}+x^{35}+x^{43}+x^{47}+x^{52}+x^{59}+\\&x^{66}+x^{71}+x^{80}+x^{17}x^{20}+x^{17}x^{20}x^{28}x^{35}x^{43}+x^{47}x^{52}\\&x^{59}x^{65}x^{71}+x^{20}x^{28}x^{43}x^{47}+x^{17}x^{20}x^{59}x^{65}+x^{43}x^{47}+\\&x^{65}x^{71}+x^{20}x^{28}x^{35}+x^{47}x^{52}x^{59}+x^{17}x^{35}x^{52}x^{71}+x^{28}\\&x^{35}x^{43}x^{47}x^{52}x^{59}\end{aligned}$$

NFSR 状态更新函数为

$$\begin{aligned}b_{t+80}=&b_{t+63}b_{t+45}b_{t+28}b_{t+9}+b_{t+60}b_{t+52}b_{t+37}b_{t+33}+s_t+b_{t+62}+\\&b_{t+60}+b_{t+52}+b_{t+45}+b_{t+37}+b_{t+33}+b_{t+28}+b_{t+21}+\\&b_{t+14}+b_{t+9}+b_t+b_{t+63}b_{t+60}+b_{t+37}b_{t+33}+b_{t+15}b_{t+9}+\\&b_{t+60}b_{t+52}b_{t+45}+b_{t+33}b_{t+28}b_{t+21}+b_{t+52}b_{t+45}b_{t+37}b_{t+33}\\&b_{t+28}b_{t+21}+b_{t+63}b_{t+60}b_{t+52}b_{t+45}b_{t+37}+b_{t+33}b_{t+28}b_{t+21}\\&b_{t+15}b_{t+9}+b_{t+63}b_{t+60}b_{t+21}b_{t+15}\end{aligned}$$

1.4 输出函数

布尔函数 $h(x)$ 为 1 阶相关免疫的代数、次数为 3 次、线性复杂度为 12 的均衡布尔函数。$h(x)$ 的详细表达式为

$$\begin{aligned}h(x)&=h(x_0,x_1,x_2,x_3,x_4)\\&=x_1+x_4+x_0x_3+x_2x_3+x_3x_4+x_0x_1x_2+x_0x_2x_3+\\&\quad x_0x_2x_4+x_1x_2x_4+x_2x_3x_4\end{aligned}$$

密钥流序列为

$$z_t=\sum_{j\in A}b_{t+j}+h(s_{t+3},s_{t+25},s_{t+46},s_{t+64},s_{t+63})$$

式中，$A=\{1,2,4,10,31,43,56\}$。

1.5 初始化过程

初始化过程如下：将 NFSR 用密钥种子加载，LFSR 的低 64 位用初始向量加载，其余 16 位设为 1。将输出的密钥流与 LFSR、NFSR 的输入变量异或后作为寄存器的输入变量，进行密钥混淆。运行 160 个时钟周期后初始化过程结束。

$$(b_0,b_1,\cdots,b_{79})=(k_0,k_1,\cdots,k_{79})$$

$$(s_0,s_1,\cdots,s_{79})=(IV_0,\cdots,IV_{63},1,\cdots 1)$$

2 攻击方法

2.1 基本概念

定义 1 设 $\{a_k\},\{b_k\}$ 为 $GF(2)$ 上的两条序列，对其做如下计算：

$$a_0+b_0,a_1+b_1,\cdots,a_{n-1}+b_{n-1},\cdots$$

则称生成的序列为 $\{a_k\}$ 和 $\{b_k\}$ 的差分序列。

定义 2 设 f 是 $GF(2)^n\to GF(2)$ 的一个映射，$(x_1,x_2,\cdots,x_n)$ 是自变量，改变 $(x_1,x_2,\cdots,x_n)$ 的若干分量值，得到 $(x_1',x_2',\cdots,x_n')$，计算

$$\Delta f(x_1,x_2,\cdots,x_n)=f(x_1',x_2',\cdots,x_n')+f(x_1',x_2',\cdots,x_n')$$

$$x_i'=x_i \text{ 或 } x_i'=x_i+1,(i=1,2,\cdots,n)$$

则称 Δf 为选择差分。

定义 3 假设攻击者控制了密码设备，可以正确地使用密码设备生成密钥流，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密钥流。利用正确的和错误的密钥流进行分析比较，从而得出内部状态，这样的错误引入攻击也称为差分错误分析。

2.2 攻击模型

根据文献[2]提出的错误攻击模型，假设攻击者能够在生成密钥流阶段的开始时刻在 LFSR 某些特定位置上引入一个或两个错误，同时能够以某种方式确定错误引入的确切位置。此外，攻击者可以多次重启密钥生成机，并在 LFSR 不同的位置上重新引入错误，其目的是通过比较错误引入前后密钥流的变化，从而确定寄存器内部初始状态的信息。

2.3 攻击步骤

将初始化过程结束后 NFSR 内部状态标记为 $b_1,b_2,\cdots,b_{80}$，LFSR 内部状态标记为 $s_1,s_2,\cdots,s_{80}$。通过对 Grain-v1 的仿真实验观察发现，其密钥流前 17 轮密钥次数为 3。利用 Grain-v1 前 17 轮密钥流次数较低的缺点，对其进行攻击，将会发现前 16 轮密钥表达式具有相同的结构。即

$$\begin{aligned}z_t=&b_{t+64}+s_{t+26}+b_{t+64}*s_{t+65}+s_{t+4}*s_{t+65}+\\&b_{t+64}*s_{t+4}*s_{t+47}+b_{t+64}*s_{t+47}*s_{t+65}+s_{t+26}*s_{t+4}*\\&s_{t+47}+b_{t+5}+b_{t+57}+s_{t+47}*s_{t+65}+b_{t+64}*s_{t+26}*s_{t+47}+\\&b_{t+11}+b_{t+2}+b_{t+3}+b_{t+32}+b_{t+44}+s_{t+4}*s_{t+47}*s_{t+65}\end{aligned}$$

式中，$0\leqslant t\leqslant 15$。所提出的攻击步骤如下：第一步：直接求出部分内部状态，将得出的内部状态代入密钥流方程，组成线性方程组。第二步：猜测内部状态，利用高斯消元法得出全部内部状态。第三步：得出密钥种子。

第一步　求出 80 个内部状态。观察非线性函数 $h(x)$ 的输入变量为 $b_{i+64},s_{i+4},s_{i+26},s_{i+47},s_{i+65}$。向 LFSR 的 4 个位置引入错误，分别有以下几种情况：在 s_{i+4} 位置加入错误；在 s_{i+26} 位置加入错误；在 s_{i+65} 位置加入错误；在 s_{i+4},s_{i+26} 位置加入错误；在 s_{i+4},s_{i+47} 位置加入错误；在 s_{i+4},s_{i+65} 位置加入错误；在 s_{i+26},s_{i+47} 位置加入错误。通过计算可求出：$b_{i+64},s_{i+4},s_{i+26},s_{i+47},s_{i+65}$ $(0\leqslant i\leqslant 15)$，共 80 个初始内部状态的值。

根据文献[2]，使用 MATLAB 符号编程，得到 Grain-v1 的状态方程。求解过程如下：$b_1,b_2,\cdots,b_{80},s_1,s_2,\cdots,s_{80}$，分别表示 Grain 在初始化过程结束后 NFSR、LFSR 初始时刻的 160 个内部状态。向 LFSR 引入错误，有以下方程 $(0\leqslant i\leqslant 15)$：

$$Z_i''=Z_i+Z_i'=s_{i+65}+b_{i+64}*s_{i+47}+s_{i+26}*s_{i+47}+s_{i+47}*s_{i+65} \tag{1}$$

$$Z_i''=Z_i+Z_i'=1+b_{i+64}*s_{i+47}+s_{i+4}*s_{i+47} \tag{2}$$

$$\begin{aligned}Z_i''&=Z_i+Z_i'\\&=b_{i+64}+s_{i+4}+s_{i+47}+b_{i+64}*s_{i+47}+s_{i+4}*s_{i+47}\end{aligned} \tag{3}$$

$$\begin{aligned}Z_i''&=Z_i+Z_i'\\&=1+s_{i+65}+s_{i+47}+s_{i+47}*s_{i+26}+s_{i+47}*s_{i+4}+s_{i+47}*s_{i+65}\end{aligned} \tag{4}$$

$$\begin{aligned}Z_i''&=Z_i+Z_i'\\&=s_{i+65}+b_{i+64}*s_{i+26}+b_{i+64}*s_{i+4}+b_{i+64}*s_{i+47}+b_{i+64}*s_{i+65}+s_{i+4}*s_{i+26}+b_{i+64}+s_{i+26}*s_{i+47}+s_{i+4}*s_{i+65}+s_{i+65}*s_{i+47}+s_{i+26}\end{aligned} \tag{5}$$

$$\begin{aligned}Z_i''&=Z_i+Z_i'\\&=1+b_{i+64}+s_{i+4}+s_{i+65}+s_{i+26}*s_{i+47}+s_{i+4}*s_{i+47}+s_{i+47}*s_{i+65}\end{aligned} \tag{6}$$

$$Z_i''=Z_i+Z'_i$$
$$=1+b_{i+64}+s_{i+65}+s_{i+4}+b_{i+64}*s_{i+26}+b_{i+64}*s_{i+4}+b_{i+64}*s_{i+47}+b_{i+64}*s_{i+65}+s_{i+4}*s_{i+26}+s_{i+4}*s_{i+47}+s_{i+4}*s_{i+65} \quad (7)$$

由式(2)与式(3)相加,得到

$$b_{i+64}+s_{i+4}+s_{i+47}=K \quad (8)$$

由式(5),式(6),式(7)相加,得到

$$b_{i+64}+s_{i+26}+s_{i+65}=K \quad (9)$$

由式(1),式(2),式(4)相加,得到

$$s_{i+47}=K \quad (10)$$

将式(9),式(10)代入式(1),得到

$$s_{i+65}=K \quad (11)$$

将式(8),式(9),代入式(5),得到

$$b_{i+64}=K \quad (12)$$

这样立即得到以下 80bits

$k_{64},k_{65},\cdots,k_{79},s_4,s_5,\cdots,s_{19},s_{26},s_{27},\cdots,s_{41},s_{47},s_{48},\cdots,s_{62},s_{65},s_{66},\cdots,s_{80}$

将得到的这 80bits 内部状态代入 $Z_0,Z_1,Z_2,\cdots,Z_{15}$,又得到 16 个线性方程,分别为

$$Z_0=b_{11}+b_2+b_3+b_{32}+b_{44}+b_5+b_{57}+K \quad (13)$$
$$Z_1=b_{12}+b_3+b_4+b_{33}+b_{45}+b_6+b_{58}+K \quad (14)$$
$$Z_2=b_{13}+b_4+b_5+b_{34}+b_{46}+b_7+b_{59}+K \quad (15)$$
$$Z_3=b_{14}+b_5+b_6+b_{35}+b_{47}+b_8+b_{60}+K \quad (16)$$
$$Z_4=b_{15}+b_6+b_7+b_{36}+b_{48}+b_9+b_{61}+K \quad (17)$$
$$Z_5=b_{16}+b_7+b_8+b_{37}+b_{49}+b_{10}+b_{62}+K \quad (18)$$
$$Z_6=b_{17}+b_8+b_9+b_{38}+b_{50}+b_{11}+b_{63}+K \quad (19)$$
$$Z_7=b_{18}+b_9+b_{10}+b_{39}+b_{51}+b_{12}+b_{64}+K \quad (20)$$
$$Z_8=b_{19}+b_{10}+b_{11}+b_{40}+b_{52}+b_{13}+b_{65}+K \quad (21)$$
$$Z_9=b_{20}+b_{11}+b_{12}+b_{41}+b_{53}+b_{14}+b_{66}+K \quad (22)$$
$$Z_{10}=b_{21}+b_{12}+b_{13}+b_{42}+b_{54}+b_{15}+b_{67}+K \quad (23)$$
$$Z_{11}=b_{22}+b_{13}+b_{14}+b_{43}+b_{55}+b_{16}+b_{68}+K \quad (24)$$
$$Z_{12}=b_{23}+b_{14}+b_{15}+b_{44}+b_{56}+b_{17}+b_{69}+K \quad (25)$$
$$Z_{13}=b_{24}+b_{15}+b_{16}+b_{45}+b_{57}+b_{18}+b_{70}+K \quad (26)$$
$$Z_{14}=b_{25}+b_{16}+b_{17}+b_{46}+b_{58}+b_{19}+b_{71}+K \quad (27)$$
$$Z_{15}=b_{26}+b_{17}+b_{18}+b_{47}+b_{59}+b_{20}+b_{72}+K \quad (28)$$

第二步　分别在 s_1,s_{42} 位置引入错误,得到

$$Z''_{16}=Z_{16}+Z'_{16}=b_{80}+s_{20}+s_{63}+b_{80}*s_{63}+s_{20}*s_{63} \quad (29)$$
$$Z''_{16}=Z_{16}+Z'_{16}=1+b_{80}*s_{63}+s_{20}*s_{63} \quad (30)$$

可推出方程

$$b_{80}+s_{20}+s_{63}=K \quad (31)$$

猜测 LFSR 剩余的 16bits 内部状态,代入式(31),得到 b_{80}。代入 Z_{16},得到一个线性方程

$$Z_{16}=b_{18}+b_{19}+b_{21}+b_{27}+b_{48}+b_{60}+b_{73}+K \quad (32)$$

将前 17 轮密钥流方程组成线性方程组。由 MATLAB 实验表明,式(13),…,式(28),式(32)这 17 个线性方程是线性无关的,只需在 NFSR 中再猜测 46bits 信息,使用高斯消元法可求解出 17bits,从而得出全部内部状态。

第三步　设任意时刻 t,Grain v1 的内部状态为 $S_t=\{s_t,\cdots,s_{t+79},b_t,\cdots,b_{t+79}\}$,由于在初始化过程中不生成密钥流,密钥流输出作为输入反馈到 LFSR、NFSR 的更新函数中,可以推导出前一时刻的密钥公式

$$z_{t-1}=b_i+b_{i+1}+b_{i+3}+b_{i+9}+b_{i+30}+b_{i+42}+b_{i+55}+h(s_{i+2},s_{i+24},s_{i+45},s_{i+63},b_{i+62})$$

则 s_{t-1},b_{t-1} 的值为

$$s_{t-1}=z_{t-1}+s_{t+79}+s_{t+61}+s_{t+50}+s_{t+37}+s_{t+22}+s_{t+12}$$
$$b_{t-1}=s_{t-1}+b_{t+61}+b_{t+59}+b_{t+51}+b_{t+44}+b_{t+36}+b_{t+32}+b_{t+27}+b_{t+32}b_{t+27}b_{t+20}b_{t+14}b_{t+8}+b_{t+51}b_{t+44}b_{t+36}b_{t+32}b_{t+27}b_{t+20}+b_{t+20}+b_{t+13}+b_{t+8}+b_{t+62}b_{t+59}+b_{t+36}b_{t+32}+b_{t+79}+b_{t+14}b_{t+8}+b_{t+59}b_{t+51}b_{t+44}+b_{t+32}b_{t+27}b_{t+20}+b_{t+62}b_{t+44}b_{t+27}b_{t+8}+b_{t+59}b_{t+51}b_{t+36}b_{t+32}+b_{t+62}b_{t+59}b_{t+20}b_{t+14}+b_{t+62}b_{t+59}b_{t+51}b_{t+44}b_{t+36}$$

则可求出 S_{t-1},进而得出 $S_{t-i}(i\geqslant 0)$,最终恢复出密钥种子。

3　复杂度分析

本文所提出的攻击算法通过第一步分析可以得到 80 个内部状态,将得到的状态信息代入前 16 轮密钥流可得 16 个线性方程。第二步分别在 s_1,s_{42} 位置引入错误,得出两个方程,将它们相加,可得到一个线性方程。猜测 LFSR 剩余 16 位内部状态和 46 位 NFSR 内部状态,最后用高斯消元法求解其余的 17bits,从而得到所有的内部状态。第三步根据全部 160bits 内部状态,恢复密钥种子。其计算复杂度为 $O(2^{62}*(17)^3)\approx O(2^{74.26})$。

结束语　本文主要研究了欧洲流密码发展计划 eSTREAM 的最终候选算法之一 Grain-v1。针对前 17 轮密钥流输出函数次数较低的缺陷,使得在做差分错误攻击时,减少了所需要的密钥流长度,降低了复杂度。结果表明,Grain 在差分错误攻击下是脆弱的,仅需猜测 62bits 内部状态,就可以攻破 Grain-v1,所需的总的计算复杂度为 $O(2^{74.26})$。

攻击算法尽管还不能威胁到全部 160bits 内部状态,且准确确定错误引入位置仍是攻击中的一个难点,但是这种攻击方法可以进一步挖掘 Grain 的设计弱点以实施更有效的攻击,同时对如何设计类似于 Grain 的流密码将有重要的参考意义。

参 考 文 献

[1] ECRYPT. eSTREAM: ECRYPT Stream Cipher Project, IST-2002-507932[EB/OL]. http://www.ecrypt.eu.org/stream/

[2] Hell M, Johansson T, Meier W. Grain-a stream cipher for constrained environments[J]. International Journal of Wireless and Mobile Computing(Special Issue on Security of Computer Network and Mobile Systems), 2006

[3] Hell M, Johansson T, Maximov A, et al. Stream Cipher Proposal: Grain-128[C]//ISIT 2006. 2006

[4] Khazaei S, Hassanzadeh M, Kiaei M. Distinguishing Attack on Grain, posted on eSTREAM webpage[EB/OL]. www.ecrypt.eu.org/stream/, 2005

[5] Isobe T, Ohigashi T, Kuwakado H, et al. A chosen-IV attack against Grain[C]//Proc. Information and Communication System Security(ICSS2007-3). Sep. 2007: 15-20