比特搜索生成器的快速密钥恢复攻击

贾艳艳，胡予濮，高军涛

(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】针对比特搜索生成器，利用Martin Hell关于自缩生成器的攻击思想，提出了一种基于多段密钥流的概率快速密钥恢复攻击。与目前已知的最好攻击结果相比，该攻击能够将计算复杂度从$O(2^{0.5L}L^3)$降低到$O(2^{0.43L}L^3)$，特别地，当L为96时，计算复杂度可以达到$O(2^{0.39L}L^3)$，所需的数据复杂度为$O(NL)$。实验结果表明，随着密钥段数的增多，算法的计算复杂度明显减少；密钥长度越长，该算法的攻击效果越好。

关 键 词 比特搜索生成器； 密钥流生成器； 密钥恢复攻击； 缩减生成器； 流密码

中图分类号 TN918.1 **文献标识码** A doi:10.3969/j.issn.1001-0548.2011.05.028

Fast Key Recovery Attack on the Bit-Search Generator

JIA Yan-yan, HU Yu-pu, and GAO Jun-tao

(Key Laboratory of Computer Networks and Information Security of Ministry of Education, Xidian University Xi'an 710071)

Abstract For the bit-search generator, a fast probabilistic key recovery attack based on multi segments of keystream bites is presented using the idea of Martin Hell's attack on the self-shrinking generator. Compared with the best known attack, the attack complexity can be significantly reduced from $O(2^{0.5L}L^3)$ to $O(2^{0.43L}L^3)$ if we have $O(NL)$ of keystream bits. Here L is the length of the linear feedback shift register (LFSR); N is the number of the segments of keystream bits. The experimental results show that: the complexity of this algorithm can be significantly degraded as the number of attacks increases; The longer the length of the key is, the more efficient our attack is.

Key words bit-search generator; keystream generator; key recovery attack; shrinking generator; stream ciphers

密钥流生成器的分析和设计一直是流密码的研究重点。缩减生成器[1]和自缩生成器[2]都是通过不规则删除伪随机序列中的某些比特得到密钥序列。文献[3]提出了一种新的比特搜索密钥流生成器，该生成器与缩减生成器和自缩生成器一样在软件和硬件实现上具有优良的性质。但是，比特搜索生成器(BSG)产生密钥流的速率是1/3，优于缩减生成器和自缩生成器的1/4。比特搜索生成器的变形主要有ABSG和MBSG两种[4]，其中ABSG还被选为eSTREAM的候选算法DECIM的非线性滤波部件。因此，对该类生成器的研究很有现实意义。

比特搜索生成器结构简单，安全性却很高。现有攻击的复杂度关于线性反馈移位寄存器(LFSR)的长度都还是指数级的[3,5-8]。目前已知的关于BSG最好的攻击是文献[7]给出的基于比特搜索的密钥恢复攻击，文献[9]用相似的攻击思想对自缩生成器提出了两种攻击，其中基于长密钥流段的攻击与基于一个短密钥流段的攻击相比，攻击效果要好得多。考虑到文献[7]只用了很短的一段密钥流序列，本文基于N段密钥流序列，提出一种改进的概率快速密钥恢复攻击。

1 基础知识

BSG的原理如图1所示。该生成器主要由一个线性反馈移位寄存器和一个选择逻辑单元组成。缩减生成器和自缩生成器都是基于比特搜索的密钥流生成器，沿着输入序列搜索比特1来确定输出序列。BSG不再搜索比特1而是搜索某个比特b，b随着搜索进程由输入序列决定其具体取0或1，当比特b再次发生时搜索结束。如果搜索进程只读一个比特就结束，则输出0，否则输出1，搜索结束后的下一个比特值作为新的b进行新一轮搜索。文献[3]中给出了BSG

收稿日期：2009– 12 – 28；修回日期：2010 – 03 – 29
基金项目：国家自然科学基金(60833008; 60803149)；国家973计划(2007CB311201)
作者简介：贾艳艳(1983 –)，女，博士生，主要从事流密码分析方面的研究.

的具体定义，本文考虑BSG的另一种等价差分描述。设输入序列s的差分序列为$d=(d_0,d_1,\cdots)$，其中$d_i=s_i\oplus s_{i+1}, i\geqslant 0$。BSG在差分序列$d$上的操作如下：若$d_i$=1，则$s_i\neq s_{i+1}$，输出1，并继续沿着序列$d$搜索，直到搜索到$d_j$=1为止；若$d_i$=0，则$s_i$=$s_{i+1}$，输出0，至此第一次搜索结束，跳过一个比特，进行新一轮搜索。两种构造方法如表1所示。

图1 比特搜索生成器框图

表1 BSG序列的生成

BSG序列(原始构造)	BSG序列(差分构造)
$i=-1; j=-1;$	$i=0; j=0;$
while (1)	while (1)
$i++; j++;$	$z[j]=d[i];$
$b=s[i];$	if ($d[i]=1$) $i++;$
$i++;$	while ($d[i]==0$)
if ($s[i]=b$) $z[j]=0;$	$i++;$
else $z[j]=1;$	$i+=2;$
while ($s[i]!=b$) $i++;$	$j++;$

例1 当线性反馈移位寄存器序列为101101110010010111 1⋯，其差分序列为1001001011100011010 1⋯时，BSG以如下方式生成密钥流序列：

$$\underbrace{1001}_{1}\ \underbrace{00}_{0}\ \underbrace{101}_{1}\ \underbrace{11}_{0}\ \underbrace{00}_{0}\ \underbrace{0110}_{1}\ \underbrace{101}_{1}\cdots$$

$$\underbrace{101}_{1}\ 1\ \underbrace{0}_{0}\ 1\ \underbrace{11}_{1}\ 0\ \underbrace{0}_{0}\ 1\ \underbrace{0}_{0}\ 0\ \underbrace{101}_{1}\ 1\ \underbrace{11}_{1}\cdots$$

由上例可知，LFSR序列和其差分序列生成的BSG序列是相同的，即说明BSG序列可以由LFSR的差分序列唯一确定。因此，若能够重构差分序列，则一定可以唯一确定BSG产生的后续密钥流比特，同时也可以恢复出初始密钥K。

2 攻击方法

2.1 算法基础和思想

若序列s可以由一个LFSR生成，则其差分序列d一定可以由同一个LFSR生成，并且序列s和d是移位等价的[10]。若序列d已知，只需要猜测s的第一个比特，就可以由d重构出序列s。基于此，本文考虑首先用猜测确定攻击的思想从截获的密钥流序列z恢复出差分序列d的L比特信息，进而得到移位寄存器的初态即初始密钥K。算法的基本思想是：首先从截获的密钥流序列中选出N段满足要求的密钥流段，然后对每个选定的密钥流段z'以猜测的方式确定出候选差分序列的L比特信息，由此可以得到关于LFSR内部状态的L个线性方程。通过求解该线性系统，得到一个内部状态，将由该状态产生的比特流与真正的密钥流z进行比较，若相同，则攻击成功；否则，通过删除z'的最后一个比特重新猜测出差分序列的L比特信息进行测试。

观察差分序列生成密钥流的方式易知，z_i=1以1/2的概率对应着差分序列d中的$(1,1,-)$，以1/4的概率对应着d中的$(1,0,1,-)$，以1/8的概率对应着d中的$(1,0,0,1,-)$，那么，z_i=1以2^{-j-1}的概率对应着d中的$(1,0^j,1,-)$。因此，差分序列中两个“1”之间需插入0的数目的期望为$\sum_{j=0}^{\infty}2^{-j-1}j=1$。基于以上分析，通过猜测$z_i$=1对应差分序列的$(1,0^j,1,-)$中0的个数来确定差分序列的$L$比特信息。

从截获的密钥流序列选择一段z'=10011⋯010000，其中包含a个1和b个0，并且满足$2a+b=L$。设序列z'恢复出的d'为：

$$z'\quad 1\quad 0\ 0\quad 1\quad 1\qquad \cdots\ 0\quad 1\qquad 0\ 0\ 0\ 0$$

$$d'\ 10^{j_1}1\times 0\times 0\times 10^{j_2}1\times 10^{j_3}1\times\cdots\times 0\times 10^{j_a}1\times 0\times 0\times 0\times 0$$

其中，$j_1+j_2+\cdots+j_a=k$为z'中a个1所对应d'中的a个位置共插入0的个数；“×”表示不确定的一比特信息。

首先猜测k=0，因为$2a+b=L$，所以能够得到候选差分序列d'的L个比特，即得到关于LFSR内部状态的L个线性方程，解该方程系统得到LFSR的一个状态，由该状态产生的密钥流序列若和z相同，则攻击成功；否则，删除z'的最后一个比特。由于需要关于LFSR内部状态的L个方程，所以删除z'的最后一个比特后，在d'的a个位置需插入k个0使得仍有$2a+b+k=L$成立，所以若z'中的一个0被删除，则令$k=k+1$；若1被删除，则令$k=k+2$。

设共插入k个0的概率为p，那么由n重贝努力试验及组合论知识，可以得到：

当k=0时，$p=\binom{a}{0}2^{-a}$；当k=1时，$p=\binom{a}{1}2^{-2}\times 2^{-(a-1)}=\binom{a}{1}2^{-a-1}$；当$k$=2时，$p=\binom{a}{1}2^{-3}\times 2^{-(a-1)}+\binom{a}{2}2^{-2}\times 2^{-2}\times 2^{-(a-2)}=\binom{a+1}{2}2^{-a-2}$；当$k$=3时，$p=\binom{a}{1}2^{-4}\times 2^{-(a-1)}+2\binom{a}{2}2^{-3}\times 2^{-2}\times 2^{-(a-2)}+\binom{a}{3}2^{-2}\times$

$2^{-2}\times 2^{-2}\times 2^{-(a-3)}=\binom{a+2}{3}2^{-a-3}$，…。归纳可知，共插入$k$个0的概率为 $p=\binom{a+k-1}{k}2^{-a-k}$，那么基于所选择的密钥流段 z'，攻击成功的概率为：

$$q=\sum_{k=0}^{k_{\max}}\binom{a+k-1}{k}2^{-a-k} \tag{1}$$

攻击所需的测试数即攻击的计算复杂度为：

$$c=\sum_{k=0}^{k_{\max}}\binom{a+k-1}{k} \tag{2}$$

式中，$k_{\max}$是使得攻击成功的概率q大于某个给定值的最小整数。

2.2　攻击步骤

考虑N个密钥流段 z'，对于每个 z'，要求单次攻击成功的概率q比较小，使得单次攻击的计算复杂度较小。基于概率论知识，N次攻击总的成功概率为 $Q=1-(1-q)^N$，只要满足Q>0.5，即可以达到文献[7]的攻击效果。在下一节的算法分析中，将给出本文算法的具体结果，并与目前最好的攻击进行比较。

算法的主要步骤如下：

1) 从截获的密钥流序列 z 中挑选出 N 段 z'，使得 z' 满足 $2a+b=L$；初始化 n 为 1；2) 取第 n 段 z'，初始化 k 为 0；3) 用组合论方法将 k 个 0 插入到 d' 中；4) 将新得到的 d' 中的已知的 L 比特写成关于 LFSR 内部状态的 L 个线性方程，求解此线性系统，得到 LFSR 的一个内部状态；5) 将得到的内部状态作为 LFSR 的初态，将由此产生的 BSG 序列与截获的真正的密钥流序列 z 比较，若不同，则说明 d' 不是真正的差分序列，转步骤 6)；若相同，则此时的 d' 即为真正的差分序列，转步骤 9)；6) 删除 z' 的最后一个比特，若 0 被删除，则令 $k=k+1$，否则，令 $k=k+2$；7) 判断 $k\leqslant k_{\max}$ 是否成立，若不成立，转步骤8)；若成立，则转步骤3)；8) 令$n=n+1$，判断 $n\leqslant N$ 是否成立，若成立，则转步骤2)；若不成立，则攻击失败；9) 根据此时的LFSR的内部状态及BSG以1/3的速率产生密钥流序列的特点，反向运行线性反馈移位寄存器就可以得到初始密钥K，攻击成功，算法停止。

3　算法分析

一般密钥流序列都是0和1均衡的。所以，当 z' 的长度为奇数时，令$a=b+1$；否则，令$a=b$。两种情况类似，所以只考虑$a=b$的情形。

当$a=b$时，有：

$$\left.\begin{aligned}2a+b+k&=L\\ a&=b\end{aligned}\right\}\Rightarrow a=\left\lceil\frac{L-k}{3}\right\rceil$$

那么对于一个密钥段 z' 攻击成功的概率即单次攻击成功的概率为：

$$q=\sum_{k=0}^{k_{\max}}\binom{\left\lceil\frac{L-k}{3}\right\rceil+k-1}{k}2^{-\left\lceil\frac{L-k}{3}\right\rceil-k}$$

则本文攻击总的成功概率为：

$$Q=1-(1-q)^N=1-\left(1-\sum_{k=0}^{k_{\max}}\binom{a+k-1}{k}2^{-a-k}\right)^N \tag{3}$$

因为要求攻击成功的概率 $Q=1-(1-q)^N>0.5$，所以式(3)中的$k_{\max}$即为满足单次攻击成功的概率 $q>1-2^{-(1/N)}$ 的最小整数。对每一个密钥段 z' 对应的候选差分序列 d' 共需要进行猜测的次数为：

$$c=\sum_{k=0}^{k_{\max}}\binom{a+k-1}{k}=\sum_{k=0}^{k_{\max}}\binom{\left\lceil\frac{L-k}{3}\right\rceil+k-1}{k}$$

每次猜测后的确定过程还需要求解L个线性方程组成的线性系统。文献[11]提出了一种求解L维线性系统的算法，其复杂度大概为 L^w，理论上 $w\leqslant 2.376$。但是该算法的复杂度有一个非常大的常数因子。本文使用文献[12]中提出的算法，该算法也是目前已知的最快且实际可行的算法，大约需要进行 $7L^{\log_2 7}$ 次操作。为了简洁，计算复杂度约为$O(L^3)$，那么该攻击所需总的计算复杂度为：

$$C=NL^3c=NL^3\sum_{k=0}^{k_{\max}}\binom{a+k-1}{k}=$$
$$NL^3\sum_{k=0}^{k_{\max}}\binom{\left\lceil\frac{L-k}{3}\right\rceil+k-1}{k} \tag{4}$$

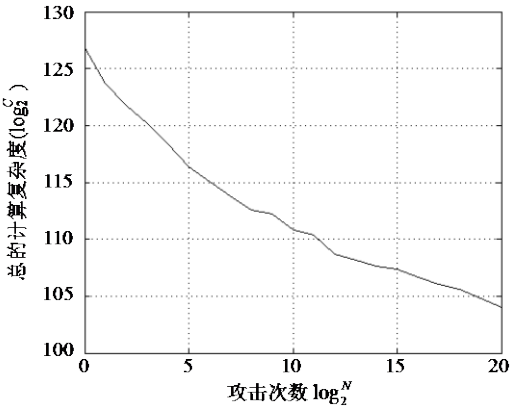

图2　计算复杂度与攻击次数的关系

选取LFSR的长度为256，用MATLAB仿真式(4)，得到总的计算复杂度C随着攻击次数N的变化曲线，如图2所示。由图2可以看出，随着攻击次数的增加，算法的计算复杂度明显降低；如当选取攻击次数为2^{10}=1 024时(此时所需的数据量也是实际可以得到的)，该算法与文献[7]中的算法即N=1的情况相比，复杂度大概可以降低2^{17}。为了进一步证明本文算法的有效性，下面通过对具有不同长度LFSR的BSG进行攻击加以验证。

假设能够从截获的密钥流中选取$N=2^{10}$个密钥段，即总共进行N次攻击，要求总的攻击成功的概率Q大于0.5，所以单次攻击成功的概率需满足$q>1-2^{-(1/N)}=6.766\,7\times10^{-4}$，那么$k_{\max}$即为$q$大于$6.766\,7\times10^{-4}$的最小整数。

表2 本文算法复杂度与现有最好结果的比较

密钥	本文算法		现有最好结果	
长度	$k_{\max}$	计算量	$k_{\max}$	计算量
64	5	$2^{25.70}$	19	$2^{31.74}$
96	9	$2^{37.37}$	27	$2^{47.50}$
128	15	$2^{52.54}$	36	$2^{63.96}$
160	21	$2^{67.40}$	44	$2^{79.82}$
192	27	$2^{81.75}$	52	$2^{95.71}$
224	33	$2^{96.25}$	61	$2^{112.37}$
256	39	$2^{110.80}$	69	$2^{128.29}$

表2列出了当a=b且攻击成功的概率大于0.5时，对于不同长度的LFSR，该算法所需的计算复杂度。由表2可以看出，对于具有不同密钥长度的BSG，该算法与现有关于BSG的最好攻击结果相比，计算复杂度可以降低6～18个数量级，达到$O(2^{0.43L}L^3)$，特别地当L=96时，该算法能够将计算复杂度降低到$O(2^{0.39L}L^3)$；并且可以看出密钥长度越长，该算法的计算复杂度比文献[7]降低更多。

分析算法不难发现，由于本文考虑了多个密钥流段z'，因此不可避免地需要更多的数据量。因单次攻击需要L比特的差分序列信息，而密钥流序列中的$L/3$个“1”和$L/3$个“0”可以给出差分序列的$2\times L/3+L/3=L$比特信息，另外为了验证所得到的差分序列还需要大约$L/2$比特的密钥流信息，所以本文攻击需要总的密钥流长度大约为:

$$N(2\times L/3+L/3+L/2)=N\times3\times L/2$$

由上式可以看出，算法的数据复杂度虽有所增加，但其还是关于密钥长度L的多项式。当选取密钥长度为256时，对于上面所进行的攻击，所需要的数据量约为$2^{18.59}$，这在实际攻击中是可以实现的。

4 结 论

本文基于BSG序列差分构造的概率分析，对比特搜索生成器提出了一种快速密钥恢复攻击算法。与现有的攻击结果相比，该算法的数据复杂度虽从$O(L)$增加到了$O(NL)$，但计算复杂度却能够被指数级地降低。仿真结果也进一步验证了本文算法的有效性，并且表明随着获得的密钥段的增多，算法的攻击效果也将更好。另外，该算法的计算复杂度关于密钥长度L还是指数级的，因此，对更为切实可行的算法还需进一步进行研究。

参 考 文 献

[1] COPPERSMITH D, KRAWCZYK D, MANSOUR Y. The shrinking generator[C]//CRYPTO’93. Santa Barbara, USA: Springer-Verlag, 1993: 22-39.

[2] MEIER W, STAFFELBACH O. The self-shrinking generator[C]//EUROCRYPT’94. Santa Barbara, USA: Springer-Verlag, 1994: 205-214.

[3] GOUGET A, SIBERT H. The bit-search generator[C]//The State of the Art of Stream Ciphers: Workshop Record. Brugge, Belgium: [s.n.], 2004: 60-68.

[4] GOUGET A, SIBERT H, BERBAIN C, et al. Analysis of the bit-search generator and sequence compression techniques[C]//FSE 2005. Berlin, Germany: Spinger-Verlag, 2005: 196-214.

[5] 周亮, 李胜强. 流密码与纠错码联合设计新方向——快速相关攻击译码算法研究进展[J]. 电子科技大学学报, 2009, 38(5): 555-561.
ZHOU Liang, LI Sheng-qiang. New direction for joint design of stream cipher and error-correcting codes — Advances of research on fast correlation attack decoding algorithm[J]. Journal of University of Electronic Science and Technology of China, 2009, 38(5): 555-561.

[6] KANSO A A. Modified clock-controlled alternating step generators[J]. Computer Communications, 2009, 32: 787-799.

[7] HELL M, JOHANSSON T. Some attacks on the bit-search generator[C]//FSE 2005. Berlin, Germany: Springer-Verlag, 2005: 215-227.

[8] ALTUĞ Y, AYERDEN N P, MICÇAK M K. A note on the periodicity and the output rate of bit search type generators [J]. IEEE Trans on Information Theory, 2008, 54(2): 666-679.

[9] HELL M, JOHANSSON T. Two new attacks on the self-shrinking generator[J]. IEEE Trans on Information Theory, 2006, 52(8): 3837-3843.

[10] MCELIECE R J. Finite fields for computer scientists and engineers[M]. [S.l.]: Kluwer Academic Publishers, 1987.

[11] COPPERSMITH D, WINOGRAD S. Matrix multiplication via arithmetic progression[J]. J Symbolic Computation, 1990, 9: 251-280.

[12] STRASSEN V. Gaussian elimination is not optimal[J]. Numerische Mathematic, 1969, 13: 354-356.

编 辑 漆 蓉