对自缩序列生成器的错误攻击

高军涛, 胡予濮, 李雪莲
(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, 陕西 西安 710071)

摘要: 通过改变线性反馈移位寄存器的内部状态对自缩生成器实施了错误攻击, 通过修改控制时钟对互缩生成器进行了相移错误攻击. 结果表明, 对于互缩生成器, 攻击者仅需实施 $n = \max\{n_1, n_2\}$ 次的相移延迟就可以获得密钥种子. 对于自缩生成器, 攻击者至多改变 n 次寄存器的比特值就可以获得密钥种子.

关键词: 错误攻击; 自缩生成器; 互缩生成器
中图分类号: TN918.1 文献标识码: A 文章编号: 1001-2400(2006)05-0809-05

Fault analysis for self-shrinking generator

GAO Jun-tao, HU Yu-pu, LI Xue-lian
(Ministry of Edu. Key Lab. of Computer Network and Information Security, Xidian Univ., Xi'an 710071, China)

Abstract: A fault attack on the self-shrinking generator is presented by changing the internal states in the linear feedback shift registers. Besides, a shifts attack on the shrinking generator is described by modifying the control clock. Results show that, for the shrinking generator, the attacker can obtain the secret keys by implementing $n = \max\{n_1, n_2\}$ times phase shift attacks, and that for the self-shrinking generator, the attacker can obtain the secret keys by changing the bit values in n registers.

Key Words: fault analysis; self-shrinking generator; shrinking generator

1996 年, Boneh[1] 提出了针对 RSA 公钥密码体制的错误攻击, 这是一种不用分解大数就能够攻击 RSA 的方法, 因此受到了广泛的关注. 同年, Biham[2] 等将错误攻击应用到攻击分组密码 DES 上面, 取得了一定的效果. 在 2004 年, Jonathan[3] 等针对一些流密码体制的特点, 对滤波函数生成器、钟控生成器以及满足特定条件的有限状态机为滤波函数的序列生成器实施了错误攻击, 但是对于自缩生成器和互缩生成器却没有涉及. 笔者针对自缩生成器提出了错误攻击, 同时对互缩生成器实施了相移攻击.

对于流密码来说, 用于加密的密钥是非常长的, 但其密钥种子却比较短, 能方便地存储和传输. 攻击者希望获得的就是流密码的密钥种子. 文中所说的错误攻击属于间接攻击, 是针对密码体制的硬件实现来实施的一种攻击方法. 这种方法主要是利用人为引起的比特跳变错误或者时钟变化来观测密钥流的变化, 以此得到生成器内部状态或者密钥种子的一些信息. 错误攻击的目的是获得密钥种子, 这样密码机的真正持有者在使用该密码机加密的时候, 攻击者就可以获得相应的密钥并对密文进行解密. 错误攻击基于以下的假设: 攻击者对于密码机只有有限时间的控制权, 即只有一段时间可以操作密码机; 同时能够对密码机的内部状态寄存器实施影响, 产生比特跳变引起错误. 除此以外, 攻击者能够重启密码机使其恢复到原来的初始状态, 并可以对其实施下一次错误攻击. 可以看出, 所引起的错误是暂时的, 而不是永久的. 因此错误攻击是比较隐秘的.

Skorobogatov[4] 等提出了一种利用光学仪器引起错误的方法, 该技术能够对生成器实施单个比特跳变的错误攻击. 这个比特的位置可以由攻击者按照需要来自己选取, 这为攻击流密码生成器提供了技术保障. 文中假设线性反馈移位寄存器(LFSR)的反馈多项式是已知的, 因此对于互缩和自缩生成器来说, 密码机的

收稿日期: 2005-11-07
基金项目: 国家自然科学基金资助项目(60273084); 高等学校博士点基金资助(20020701013); 现代通信国家重点实验室项目(51436030105DZ0105)
作者简介: 高军涛(1979-), 男, 西安电子科技大学博士研究生.

密钥种子就是 LFSR 的内部状态.

1 互缩生成器的相移攻击

Coppersmith 等[5] 提出了互缩生成器的概念, 该生成器由两个线性反馈移位寄存器(LFSR) 组成, 分别记为 LFSR1 和 LFSR2, 且两个 LFSR 受同一个时钟控制. 其定义的规则是: 若 LFSR2 输出 1, 则生成器输出 LFSR1 的对应比特; 否则放弃输出. 因此称 LFSR2 为钟控 LFSR, LFSR1 为被控 LFSR.

一般情况下, 两个 LFSR 的反馈多项式都是本原多项式, 因此两个 LFSR 的输出流都是 m 序列且周期是互素的, 这样可以使生成器的输出序列的周期达到最大. 该生成器目前主要用于伪随机数的生成. 针对互缩生成器的主动攻击主要有以下的几种: 分别征服攻击、概率相关攻击、低复杂度相关攻击、可区分攻击等, 这些攻击的复杂度都是指数级的.

由于两个 LFSR 受同一个时钟控制. 在某个时刻, 可以通过硬件的方法使 LFSR1 受时钟控制, LFSR2 不受钟控, 输出密钥流. 这样相比于原生成器的输出流来说, 会产生一个相移错误, 相移错误发生以后两个 LFSRS 仍旧由原来的时钟来控制. 这种相移对于攻击者来说是非常有用的, 攻击者可以利用这种相移得到 LFSR 初始状态的信息. 文献[3] 给出了对钟控生成器的相移攻击. 由于互缩生成器和钟控序列的差异, 简单照搬文献[3] 中的方法是不行的, 需要做相应的改进. 下面是攻击步骤.

设 LFSR1 的长度为 n_1, LFSR2 的长度为 n_2, 选定数 m 和 i, 设定 m 的初始值等于 0, i 的初始值为 0, 令 $n = \max\{n_1, n_2\}$. 下面以算法的形式来描述攻击步骤.

启动生成器, 输出长度为 $N = 2n$ 长的密钥流, 即正确的密钥流 $S^0 = s_0^0 s_1^0 s_2^0 \cdots s_{2n-1}^0$;

for (i= 1 to n) do

重启生成器, LFSR2 延迟 i 个时刻, LFSR1 不延迟, 随后两个 LFSR 都受同一个时钟控制, 密码机输出相应的错误密钥流 $S^i = s_0^i s_1^i s_2^i \cdots s_{2n-1}^i$, $i = i + 1$;

endfor

for (m= 0 to $n-1$) do

for (i= 0 to n) do

if $s_{m+1}^i \oplus s_m^{i+1} = 0$, then

LFSR2 的输出流(并非是生成器的输出) 中第 $m+1$ 个“1” 和第 $m+2$ 个“1” 之间是没有“0” 的, 即存在一个“11” 的游程.

else

LFSR2 的输出流中的第 $m+1$ 个“1” 和第 $m+2$ 个“1” 之间至少有一个“0”. 再比较 S^i 和 S^{i+1}, 若 $s_{m+1}^i \oplus s_m^{i+2} = 0$, $i + 2 \leqslant n$, 在 LFSR2 的输出流中的第 $m+1$ 个“1” 和第 $m+2$ 个“1” 之间只有一个“0”, 即存在一个“101” 的片断. 若存在 i 使得 $s_{m+1}^i \oplus s_m^{i+2} = 1$, 则在 LFSR2 的输出流中的第 $m+1$ 个“1” 和第 $m+2$ 个“1” 之间至少有两个“0”. 再比较 S^i 和 S^{i+3}, …, 直到存在一个 j 使得, 对于所有的 i 都有 $s_{m+1}^i \oplus s_m^{i+j} = 0$, 此时证明在 LFSR2 的输出流中的第 $m+1$ 个“1” 和第 $m+2$ 个“1” 之间存在 j 个 0, $j \leqslant n-1$.

endif

endfor

endfor

攻击者利用上述的方法最终会得到 LFSR2 的输出流中一个长度至少为 n_2 的连续的比特串, 该比特串从 LFSR2 的初始状态中第一个“1” 开始. 根据 LFSR2 的反馈多项式和上述计算得到的 n_2 长的比特串, 至多计算 n_2 次可以得到 LFSR2 的初始状态. 与之对应的 LFSR1 的初始状态可以通过 LFSR2 的初始状态和生成器正确输出流 S^0 得到.

下面用一个简单的例子来说明算法的正确性.

设 LFSR2 生成的是周期为 7 的 m 序列, LFSR1 生成的是周期为 15 的 m 序列, 按照互缩序列的方式进

行输出, 生成器正确的输出流为 S^0= 01000011…

LFSR1 相移 1 位生成器的输出流为 S^1= 00011111…

LFSR1 相移 2 位生成器的输出流为 S^2= 10110111…

LFSR1 相移 3 位生成器的输出流为 S^3= 01101110…

LFSR1 相移 4 位生成器的输出流为 S^4= 01011100…

按照攻击方法首先 m = 0 比较 S^0 和 S^1, 可知存在 $s_1^0 \oplus s_0^1 = 1$, 所以 LFSR2 的密钥种子中第 1 个 1 和第二个 1 之间至少有一个 0; 比较 S^0 和 S^2, S^1 和 S^3 可知 $s_1^0 \oplus s_0^2 = 0$, $s_1^1 \oplus s_0^3 = 0$, 所以 LFSR2 输出的序列中存在"101"这样的游程片断. m = 1 时, 可知 $s_2^0 \oplus s_1^1 = 0$, $s_2^1 \oplus s_1^2 = 0$, $s_2^2 \oplus s_1^3 = 0$, $s_2^3 \oplus s_1^4 = 0$, 所以第二个 1 和第三个 1 之间没有 0, 即存在"1011"这样的片断.

由于 LFSR2 生成的是周期为 7 的 m 序列, 所以由"1011"片断可以得到 m 序列为"1011100", 其初始状态有 3 种可能, 分别为: 001, 010, 101. 这三种可能的状态和 S^0 以及 LFSR1 的反馈多项式一起来确定 LFSR1 可能的内部状态, 然后由 S^0 来最终确定正确的初始状态.

LFSR1 和 LFSR2 的初始状态分别为: 1000 和 001. LFSR1 输出的 m 序列是 1000100110101 11.

2　自缩序列的错误攻击

自缩生成器[6] 是一类简单的序列生成器, 其结构非常简单, 仅由一个 LFSR 构成, 一般情况下, 假定该 LFSR 的反馈多项式是本原多项式, 则产生的序列 $a = a_0 a_1 a_2 \cdots$ 为 m 序列. 将序列 a 写为以下的形式: $a = (a_0, a_1), (a_2, a_3), \cdots$, 其输出满足以下的规则: 当 $(a_{2i}, a_{2i+1}) = (1, 0)$ 或者 $(1, 1)$ 时, 输出比特为 0 或 1, $(a_{2i}, a_{2i+1}) = (0, 0)$ 或者 $(0, 1)$ 时, 丢弃输出比特.

目前, 对自缩序列的攻击方法主要包括猜测攻击、时空交换攻击和 FBDD 攻击等. 针对一般的 n 级 m 序列生成的自缩序列, 所有的这些攻击的复杂度都不低于 $O(2^{0.5n})$. 由此可见虽然自缩生成器的结构比较简单, 但是目前仍然没有一种攻击方法能实际威胁到它. 这里提出的攻击方法是建立在一定的假设基础上的.

设自缩生成器中 LFSR 的长度为 n, n 为偶数, 即 LFSR 共有 0 ~ $n-1$ 级寄存器. 此时生成的密钥流序列的周期是 2^{n-1}. 假设 LFSR 是在右端输出, 其输出的 m 序列为 $a = a_0 a_1 a_2 \cdots$, 且寄存器编号为从右至左依次为第 0 级, 第 1 级, …, 第 $n-1$ 级. 攻击的目的就是获得 m 序列的前 n 个比特, 即密钥种子 $a_0 a_1 a_2 \cdots a_{n-1}$. 错误攻击建立在以下的假设上: 攻击者对于 LFSR 的初始状态的偶数位寄存器中 1 的个数是已知的, 记为 l.

在错误攻击下, 如果自缩生成器的密钥种子(LFSR 的内部状态) 是以下的形式, 则是非常不安全的.

(1) (0* 0* …0*), 密钥种子的奇数位全是 0; (2) (1* 1* …1*), 密钥种子的奇数位全是 1;

(3) (* 0* 0…* 0), 密钥种子的偶数位全是 0; (4) (* 1* 1…* 1), 密钥种子的偶数位全是 1;

其中* 表示 0 或者 1, 圆括号内从右至左依次为第 0 位, 第 1 位, …, 第 $n-1$ 位.

因为在第(1)、(2) 种情况下, 攻击者可以选择 LFSR 内部状态的一个奇数位引起跳变, 输出错误的密钥流, 然后与正确的密钥流比较就可以获得与奇数位对应的偶数位的值, 依次类推可以获得所有的密钥种子. 在第(3)、(4) 种情况下, 攻击者可以对内部状态的偶数位引起比特跳变来获得奇数位的比特值. 如果密钥种子的奇数位中有很大一部分是 0(或 1) , 同样也是不安全的. 我们将这些密钥种子称为错误攻击下的弱密钥, 一般不采用. 下面用算法的形式来描述攻击步骤.

1)　初始值 $i = 0, j = 0$, 生成器输出的正确密钥流为 $S^0 = s_0^0 s_1^0 s_2^0 \cdots$;

2)　while $j <= l$, do

3)　　重启生成器, 对 LFSR 的第 $2i+1$ 级寄存器实施错误攻击, 生成的密钥流为 $S^{2i+1} = s_0^{2i+1} s_1^{2i+1} s_2^{2i+1} \cdots$;

4)　　if $s_i^0 \neq s_i^{2i+1}$, then

5)　　$a_{2i} = 1; a_{2i+1} = s_i^0$;　　/* 初始状态中第 $2i$ 级寄存器中的元素为 1, $2i+1$ 级寄存器中的元素为 s_i^0 */

6)　　$i = i + 1$;

7) $j = j + 1$;
8) else
9) 重启生成器, 对第 $2i$ 级寄存器实施错误攻击, 生成的密钥流为 $S^{2i} = s_0^{2i} s_1^{2i} s_2^{2i} \cdots$;
10) $a_{2i} = 0$; $a_{2i+1} = s_i^{2i}$;
11) endif
12) endwhile
13) if $i < (n-2)/2$ do
14) 重启生成器, 对 LFSR 的第 $2i$ 级寄存器实施错误攻击, 生成的密钥流为 $S^{2i} = s_0^{2i} s_1^{2i} s_2^{2i} \cdots$;
15) $a_{2i} = 0$; $a_{2i+1} = s_{i+1}^{2i}$;
16) $i = i + 1$;
17) else
18) endif

攻击步骤的基本思想就是利用奇数级寄存器的比特跳变引起的错误来估计偶数级寄存器中的比特值, 然后根据得到的偶数级寄存器中的比特值来确定奇数级寄存器中的值. 下面以第 0 级和第 1 级寄存器为例进一步说明攻击步骤的基本思想. 假设 n 为偶数, 同时 LFSR 初始状态时偶数级寄存器中 1 的个数 $l > 1$, 首先对第 1 级寄存器实施错误攻击, 引起内部元素的比特跳变, 输出错误的密钥流. 若第 0 级寄存器中的比特值是 1, 则错误密钥流和正确密钥流的第一位肯定是不相等的, 反之, 若第 0 级寄存器中的比特值是 0, 由于 $l > 1$, 在其他偶数级寄存器中一定存在 1, 因此第 1 级寄存器中比特值的变化对于密钥流的第一位是没有影响的. 依此类推, 当攻击者已经知道初始状态时一部分寄存器中的比特值且其中偶数寄存器中 1 的个数已经有 l 个, 此时需要应用攻击步骤中的 13) – 18) 步: 改变偶数级寄存器中的比特值, 使其输出奇数级寄存器中的比特值. 以上所说的都是针对 n 为偶数的情况, 对于 n 为奇数的情况也可以得到类似的算法.

密钥种子的结构和攻击的效率有直接的关系, 如果在初始状态下, LFSR 的第 $n-2$ 级寄存器中的元素为 1, 即密钥种子为(* 1* * …* *), 那么上面算法的第 13) – 18) 步就不再起作用. 同时注意到如果将攻击假设改为"攻击者已知 LFSR 的初始状态中 $a_{n-2} = 1$" 同样也可以得到密钥种子.

互缩序列和自缩序列最显著的不同之处就在于两者的 LFSR 的个数不同, 互缩生成器有两个 LFSR, 此时可以利用它们之间的时钟关系来进行错误攻击, 但是对于自缩生成器, 只有一个 LFSR, 不能用时钟关系来实现错误攻击. 因此对于自缩生成器, 采取引起寄存器内部比特跳变的方式来实施错误攻击, 这种攻击方式要比互缩生成器复杂一些. 对于互缩生成器的错误攻击是没有条件的, 而对于自缩生成器的错误攻击是建立在假设"攻击者对于 LFSR 的初始状态的偶数位寄存器中 1 的个数是已知的" 基础上的, 自缩生成器所要求的条件要高一些.

3 结 论

文中研究的目的就是针对自缩生成器和互缩生成器建立一种有效的错误攻击方法, 研究发现, 针对互缩生成器存在有效的相移攻击方法, 针对自缩生成器, 当 n 值较小时, 在一定的假设条件下, 存在有效的错误攻击方法. 这些攻击方法对于自缩和互缩生成器的硬件实现来说是非常有威胁的. 文献[7] 讨论了一类广义自缩序列的伪随机性, 文中提出的攻击方法对于这一类广义自缩序列是否有效需要做进一步的研究.

参考文献:

[1] Boneh D, DeMillo R, Lipton R. On the Importance of Checking Cryptographic Protocols for Faults[A]. EUROCRYPT'97, LNCS 1233[C]. Berlin: Springer Verlag, 1997. 37-51.

[2] Biham E, Shamir A. Differential Fault Analysis of Secret Key Cryptosystems[A]. CRYPTO'97, LNCS 1294[C]. Berlin: Springer Verlag, 1997. 513-525.

[3] Hoch J J, Shamir A. Fault Analysis of Stream Ciphers[A]. CHES 2004, LNCS 3156[C]. New York: Springer Verlag, 2004. 240-253.

[4] Skorobogatov S, Anderson R. Optical Fault Induction Attacks[DB/OL]. www.lcl.cam.ac.uk/ftp/users/rja14/faultpap3.pdf, 2004-12-07.

[5] Coppersmith D, Krawczyk H, Mansour Y. The Shrinking Generator[A]. CRYPTO'93, LNCS 765[C]. Berlin: Springer-Verlag, 1994. 22-39.

[6] Meier W, Staffelbach O. The Self-shrinking Generator[A]. EUROCRYPT'94, LNCS 905[C]. Berlin: Springer-Verlag, 1994. 205-214.

[7] Dong Lihua, Gao Juntao, Hu Yupu. Pseudo-randomness of a Generalized Self-shrinking Sequences[J]. Journal of Xidian University, 2004, 31(3): 394-398.

(编辑: 高西全)

(上接第 799 页)

不是单峰值分布. 图 8 和图 9 是用 PSO 得到的不同主瓣宽度的平顶方向图. 在优化中将粒子个数设定为 1 200, 迭代 100 步, 调整适应度函数, 得到了结果, 主瓣展宽了, 但副瓣电平略有升高.

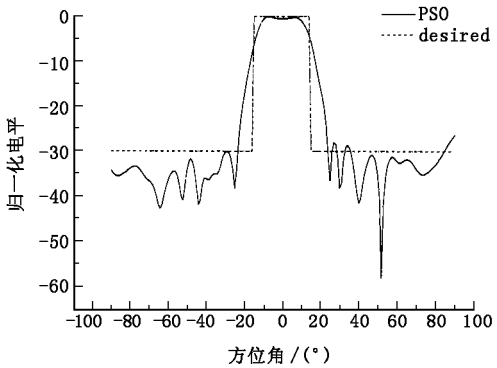

图 8　宽波束方向图

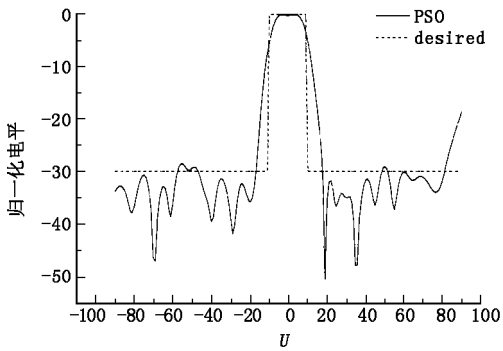

图 9　窄波束方向图

3　结束语

遗传算法和粒子群算法是两种比较新的优化算法, 有着很强大的多变量优化能力. 从文中的实例中不难看出它们在共形相控阵天线波束赋形应用中可得到很好的结果. 作为两种不同的算法, 他们在阵列优化问题中性能也有差别. GA 算法收敛较快, 但是程序代码相对较复杂, 要取得较高的精度就得大幅度增加矩阵维数, 延长计算时间. 对于 PSO 算法, 要提高收敛性并且避免早熟, 需要更多的调试, 但它的可移植性很好, 程序代码较少.

参考文献:

[1] 魏文元, 宫德民, 陈必森. 天线原理[M]. 北京: 国防工业出版社, 1985.

[2] 汪茂光, 吕善伟, 刘瑞祥. 阵列天线分析与综合[M]. 西安: 西安电子科技大学出版社, 1989.

[3] 王小平, 曹立明. 遗传算法——理论、应用[M]. 西安: 西安交通大学出版社, 2002.

[4] Yang Shuyuan, Liu Fang, Jiao Licheng. A Novel Genetic Algorithm Based on the Quantum Chromosome[J]. Journal of Xidian University, 2004, 31(1): 76-81.

[5] Boeringer D W, Douglas H. Werner Particle Swarm Optimization Versus Genetic Algorithms for Phased Array Synthesis [J]. IEEE Trans on Antennas and Propagation, 2004, 52(3): 771-779.

[6] Carlos A. Handling Multiple Objectives with Particle Swarm Optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.

[7] Boeringer D W, Werner D H. Particle Swarm Optimization of a Modified Bernstein Polynomial for Conformal Array Excitation Syntheis[M]. New York: IEEE, 2004.

(编辑: 齐淑娟)