线性反馈移位寄存器的差分能量攻击

臧玉亮　　韩文报
(解放军信息工程大学信息工程学院　郑州　450002)

摘　要：能否有效去除算法噪声的影响，直接关系到能量攻击成败。该文以线性反馈移位寄存器(LFSR)相邻两个时钟周期的能量消耗差异为出发点，提出了一种新的差分能量攻击算法。它从根本上去除了密码算法噪声在攻击过程中带来的影响。由于该算法随机选择初始向量(initialization vector)，从而使攻击者能够容易地将其推广到具有类似结构的流密码体制。为了进一步验证攻击算法的有效性，该文利用软件仿真的方法对 DECIM 进行了模拟攻击。仿真结果表明，该攻击算法能够有效降低 LFSR 的密钥搜索的复杂度。

关键词：流密码；差分能量攻击；线性反馈移位寄存器；DECIM；复杂度
中图分类号：TN918.4　　**文献标识码**：A　　**文章编号**：1009-5896(2009)10-2406-05

Differential Power Attack on Liner Feedback Shift Register

Zang Yu-liang　　Han Wen-bao
(*Institute of Information Engineering*, *PLA Information Engineering University*, *Zhengzhou* 450002, *China*)

Abstract: Whether the algorithm noise can be effectively wiped off decides the success or loss of the power analysis attack. This paper offers a new differential power analysis attack algorithm, which is based on the consumed power differences between two neighboring clock cycles of liner feedback shift register. This new attack algorithm radically wipes off the effect of cipher algorithm noise in the process of attack. Because this algorithm randomly chooses initialization vectors, the attackers can easily extend the algorithm to other stream ciphers that have similar structures. In order to further validate the algorithm's availability, simulative attacks on DECIM are carried on with the method of software simulation. And the result shows that this algorithm can effectively reduce the complexity of the exhaustive search on LFSR.

Key words: Stream cipher; Differential Power Attack (DPA); Liner Feedback Shift Register (LFSR); DECIM; Complexity

1　引言

一个完整的密码系统由密码算法设计者，硬件设计者和软件设计者共同开发完成，但他们的工作又相互独立，各层设计者通常只考虑本层的安全问题，并且一层的设计者并不一定了解其他层设计者的工作，比如密码算法设计者并不一定具有较高的硬件知识水平。其次，由于现代制造工艺的限制，无法做到在硬件中的任何运算都消耗相同的能量。因此综合上述两个原因，使得密码系统在旁道攻击[1,2]面前出现了诸多可被攻击者利用的漏洞。

自密码算法设计之初，设计者就开始考虑其算法自身抗已知攻击的能力，如代数攻击[3]。因此，攻击者从密码算法的角度来分析一个密码体制变得越来越困难。然而，旁道攻击的出现给密码算法带来了严重威胁。

从 Kocher 提出能量攻击方法[4]以来，它被广泛应用于分组密码(如 AES, DES)和公钥密码(如 RSA, ECC)分析。研究的广泛性说明了能量攻击方法对于这两种密码体制的有效性。但是我们却很少看到对流密码的能量攻击研究。文献[5]给出了 eSTREAM[6] 第 3 阶段相关候选算法抗旁道攻击的理论评估。文献[7]对 GSM 手机使用的 A5/1 流密码算法和蓝牙中所使用的 E0 流密码算法进行了能量攻击理论分析，给出了已知初始 IV 的攻击算法。文献[8]通过观察能量消耗与 LFSR 初态之间的关系建立方程，进而求得初态。文献[9]是第 1 篇对基于硬件的流密码进行实际能量攻击的文章。它针对 eSTREAM 中的 Grain 算法和 Trivium 算法提出了选择 IV 的攻击算法。有针对性地选择 16 个 IV 来降低密码算法噪声对攻击的影响，虽然作者成功获得了算法密钥，但是如何针对不同的算法来选择 IV 并没有给出选择规则，因此给该攻击算法的广泛适用带来了影响。

当攻击者对于一个流密码算法进行能量攻击时，如何克服算法噪声给攻击过程带来的影响是他必须考虑的问题。当 LFSR 的状态在某一个时钟发生变化时，攻击者可以从示波器上看到当前时钟整

2008-10-14 收到，2009-06-25 改回

个 LFSR 的电流或电压变化，但是却无法从中看到某一个触发器的变化。在这样的条件下，攻击者根据一个触发器建立的选择方程来划分整个 LFSR 的能量是不切实际的。理想的情况是，攻击者根据一个触发器建立的选择方程所划分的能量只与该触发器有关，而与剩余触发器无关。因此，本文通过考察 LFSR 相邻时钟的能量消耗差异，使得观察到的能量变化差异只与 LFSR 最左端和最右端的触发器有关，并据此提出了一种全新的方法来实施差分能量攻击。与文献[9]有针对性的选择 IV 不同，本文方法能够在随机选择 IV 的情况下，有效去除算法噪声的影响。

本文内容安排如下：在第 2 节中结合 LFSR 的能量模型和相邻时钟的能耗关系提出算法 1—差分能量攻击算法；针对第 2 节中提出的攻击算法，在第 3 节中给出了 DECIM[10]流密码算法的理论攻击方法和密钥穷尽规模分析；在第 4 节中，我们结合模拟攻击的软件仿真结果给出了恢复第 1 轮 32 bit 密钥的时间估计，并分析了软件仿真结果峰值不唯一的原因；第 5 节是结束语。

2　LFSR 的差分能量攻击

2.1 LFSR 的能量模型

在流密码的硬件实现中，触发器(flip flop)和布尔逻辑门是关键的组成部分[11]，其中触发器是组成 LFSR 的基础部件，用来记录 LFSR 的状态。当流密码运行时，能量消耗的大部分集中在触发器上，也就是 LFSR 在状态发生变换的时候。文献[11]给出了触发器和逻辑门的能量消耗对比，从中可以看到触发器的能量消耗较其它逻辑部件高出许多。所以，本文通过触发器的能量消耗来刻画 LFSR 的能量模型。为叙述简单，文中只考虑有限域 F_2 上的 LFSR。

假设　设 a,b 表示 F_2 上触发器的状态，即 $a,b\in\{0,1\}$, $P(ab)$ 表示由 a 变化到 b 所消耗的能量，则有 A：$P(01)$ 和 $P(10)$ 消耗的能量相同；B：$P(00)$ 和 $P(11)$ 消耗的能量相同；C：$P(00)=P(11)\ll P(01)=P(10)$。

本文用汉明距离模型来刻画一个时钟内 LFSR 的能量消耗。LFSR 由时钟信号触发，而后每个时钟改变一次 LFSR 的状态，整个 LFSR 的状态向左或右移一位。攻击者可以在假设基础上，通过计算相邻两个 LFSR 状态的汉明距离来模拟当前时钟消耗的能量。即在一个时钟内，LFSR 中“0”变化到“1”和“1”变化到“0”的触发器个数。用状态发生变化的触发器个数表示一个时钟内 LFSR 的能量消耗。如果记录下每一个相邻时钟触发器状态发生变化的个数，那么就可以得到一条能量消耗轨迹。

2.2 相邻时钟的能量消耗关系

设 LFSR 的级数是 n，反馈多项式为 $f(\cdot)$，其抽头个数为 m，则它有 n 个存储单元，每一个存储单元都是一个触发器，其中影响 $f(\cdot)$ 的触发器个数为 m。$s(l)$ 表示 LFSR 第 l 个输出比特值，$s(l)\in\{0,1\}$，其中 $0\le l\le 2^n-1$。LFSR 的状态每一个时钟向右移 1 位同时输出 1 bit。若 $\mathrm{LS}_t,\mathrm{LS}_{t+1},\mathrm{LS}_{t+2}$ 分别表示 LFSR 在第 $t,t+1,t+2$ 个时钟的状态，则设 $\mathrm{LS}_t=\{s(n-1),\cdots,s(0)\}$，$\mathrm{LS}_{t+1}=\{s(n),\cdots,s(1)\}$，$\mathrm{LS}_{t+2}=\{s(n+1),\cdots,s(2)\}$，其中 $s(n)=f(s(n-1),\cdots,s(0))$，$s(n+1)=f(s(n),\cdots,s(1))$。

由文献[8]可进一步推导出定理 1 和推论 1。

定理 1　设 HD_t 表示 $\mathrm{LS}_t,\mathrm{LS}_{t+1}$ 的汉明距离，PD_t 表示第 t 和 $t+1$ 个时钟的能量消耗差异，通过 2.1 节的讨论，可直接用 PD_{t+i} 表示第 i 轮的实际能量消耗差异，$i\ge 0$。则

$$\begin{aligned}\mathrm{PD}_t&=\mathrm{HD}_t-\mathrm{HD}_{t+1}\\&=\mathrm{HW}(\mathrm{LS}_t,\mathrm{LS}_{t+1})-\mathrm{HW}(\mathrm{LS}_{t+1},\mathrm{LS}_{t+2})\\&=\mathrm{HW}(s(0)\oplus s(1))-\mathrm{HW}(s(n)\oplus s(n+1))\end{aligned}\quad(1)$$

故

当 $s(n)\oplus s(n+1)=0$ 时，则 $\mathrm{PD}_t=s(0)\oplus s(1)$；

当 $s(n)\oplus s(n+1)=1$ 时，则 $\mathrm{PD}_t=s(0)\oplus s(1)-1$。

推论 1　设 HD_{t+i} 表示 $\mathrm{LS}_{t+i},\mathrm{LS}_{t+i+1}$ 的汉明距离，PD_{t+i} 表示第 $t+i$ 和 $t+i+1$ 个时钟的能量消耗差异，则

$$\begin{aligned}\mathrm{PD}_{t+i}&=\mathrm{HD}_{t+i}-\mathrm{HD}_{t+i+1}=\mathrm{HW}(s(i)\oplus s(i+1))\\&\quad-\mathrm{HW}(s(n+i)\oplus s(n+i+1))\end{aligned}\quad(2)$$

故

当 $s(n+i)\oplus s(n+i+1)=0$ 时，则 $\mathrm{PD}_{t+i}=s(i)\oplus s(i+1)$，记为 $\mathrm{PD}_{(t+i)0}$；

当 $s(n+i)\oplus s(n+i+1)=1$ 时，则 $\mathrm{PD}_{t+i}=s(i)\oplus s(i+1)-1$，记为 $\mathrm{PD}_{(t+i)1}$。

下面以第 i 轮为例，分析根据推论 1 去除算法噪声的原因。通过对相邻两个时钟的汉明距离进行比较，可知 HD_{t+i} 和 HD_{t+i+1} 之间有 $n-1$ 项重复，故 PD_{t+i} 只与 $s(i),s(i+1),s(n+i),s(n+i+1)$ 有关。若设选择方程 $C=s(n+i)\oplus s(n+i+1)$，那么第 i 轮的能量消耗差异 PD_{t+i} 将直接由选择方程 C 决定。所以去掉了 $n-1$ 个重复项产生的能量对选择方程 C 的影响，达到了去除算法噪声的目的，进而提高了选择方程 C 的划分精度。

2.3 LFSR 差分能量攻击算法

由推论 1 可知，相邻两个时钟的触发器翻转个

数在 $s(n+i)\oplus s(n+\mathrm{i}{+}1)=0$ 时比 $s(n+i)\oplus s(n+i{+}1)=1$ 时多一个，所以在能量消耗上 $\mathrm{PD}_{(t+i)0}$ 较 $\mathrm{PD}_{(t+i)1}$ 大。另外，LFSR 的初始化一般是通过密钥 K 和 IV 以及它们之间的运算来填充 LFSR 状态。因此本文提出算法 1，它在密钥 K 固定的情况下，通过 LFSR 不同时钟间的能量消耗差异，恢复 LFSR 的密钥 K。

假设恢复 LFSR 的密钥 K 需要 k 轮。在反馈多项式的 m 个抽头中有 r 个抽头完全由 IV 决定，设 IV 集合为 SET_IV，阶为 d；剩下的 $m-r$ 个抽头由密钥 K 或者 K 和 IV 共同决定，则每个时钟的密钥穷尽量为 2^{m-r}，相邻两个时钟的密钥穷尽量最大为 $2^{2(m-r)}$。若选择方程 $C=s(n+i)\oplus s(n+i{+}1)$，则当 $C=0$ 时对应的能量集合为 SET_P0；当 $C=1$ 时对应的能量集合为 SET_P1。在攻击算法开始前，假定已经采集到了由 SET_IV 中不同 IV 和固定密钥 K 所对应的连续 k 轮的能量消耗差异 PD_{t+i}。

算法 1 LFSR 差分能量攻击算法

for i=0 to $k-1$

for $K\_\mathrm{num}=0$ to $2^{2(m-r)}-1$

for IV_num=1to d

步骤1 从 SET_IV 中选取一个 IV；

步骤2 根据 IV 和假定的子密钥 K_num 对 LFSR 的抽头位置进行填充；

步骤3 根据反馈多项式 $f(\cdot)$，计算相邻两个时钟的 $s(n+i)$ 和 $s(n+i+1)$；

步骤4 计算选择方程

$$C=s(n+i)\oplus s(n+i+1) \tag{3}$$

当 C=0 时，将 PD_{t+i} 放入集合 SET_P0；

当 C=1 时，将 PD_{t+i} 放入集合 SET_P1；

结束

步骤5 计算差分公式

$$\mathrm{peak}=\frac{1}{|\mathrm{SET\_P0}|}\sum_{\mathrm{PD}_{t+i}\in\mathrm{SET\_P0}}(1-C)\cdot\mathrm{PD}_{t+i}-\frac{1}{|\mathrm{SET\_P1}|}\sum_{\mathrm{PD}_{t+i}\in\mathrm{SET\_P1}}C\cdot\mathrm{PD}_{t+i} \tag{4}$$

步骤6 选择最大的 peak 对应的子密钥 K_num 为第 i 轮的真子密钥；

结束

步骤7 返回最外层循环，寻找下一轮 LFSR 抽头所用到的子密钥。

结束

3 DECIM 的差分能量攻击

3.1 DECIM

DECIM 是一个面向硬件的流密码算法，是 eSTREAM 计划第 3 阶段候选算法之一。它的密钥 K 长度为 80 bit，记为 $K=(K_0,K_1,\cdots,K_{79})$，IV 长度为 64 bit，记为 $\mathrm{IV}=(\mathrm{IV}_0,\mathrm{IV}_1,\cdots,\mathrm{IV}_{63})$。DECIM 是由一个 F_2 上 192 级线性反馈移位寄存器，一个有 14 个变量的布尔函数 $g(\cdot)$，一个不规则混乱机制(ABSG)和一个 32 bit 输出缓存组成。DECIM 共有 192 bit 内部状态 $(a_{191},a_{190},\cdots,a_1,a_0)$，LFSR 的初始化过程可以通过式(5)表示：

设 K_t 是第 t 个时钟的密钥，IV_t 是第 t 个时钟的初始值，其中 $\{K_t,\mathrm{IV}_t\}\in F_2$，则

$$a_t=\begin{cases}K_t, & 0\le t\le 79\\ K_{t-80}\oplus\mathrm{IV}_{t-80}, & 80\le t\le 143\\ K_{t-80}\oplus\mathrm{IV}_{t-144}\oplus\mathrm{IV}_{t-128}\oplus\mathrm{IV}_{t-112}, & 144\le t\le 159\\ \mathrm{IV}_{t-160}\oplus\mathrm{IV}_{t-128}\oplus 1, & 160\le t\le 191\end{cases} \tag{5}$$

在对 LFSR 进行初始化后，LFSR 空转 768 个时钟，不输出任何值，如图 1。设反馈多项式 $f(\cdot)$ 和布尔函数 $g(\cdot)$ 在第 $192+i$ 个时钟的输出分别记为 c_i 和 b_i。因此 a_{192+i} 在第 $192+i$ 个时钟的值为

$$a_{192+i}=b_i\oplus c_i \tag{6}$$

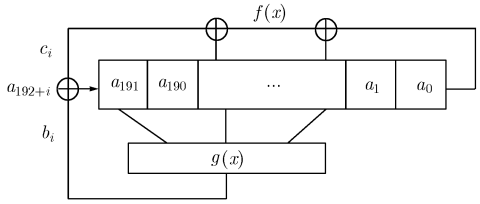

图 1 IV/keysetup 阶段流程图

3.2 能量模型

DECIM 的硬件实现中共用到了 196 个触发器，其中 192 个用于 LFSR，4 个用于 ABSG，其余是加法和异或等逻辑门[10]。在 IV/key setup 阶段，第 $192+i$ 个时钟的能量消耗 $P(192+i)$ 将包含 4 部分：192 个触发器消耗的能量，记为 $\sum_{j=0}^{191}p_{ff_j}(s_1s_2)$，$f(\cdot)$ 消耗的能量 P_f，$g(\cdot)$ 消耗的能量 P_g，一个 $f(\cdot)\oplus g(\cdot)$ 中异或消耗的能量 P_{xor} 和非算法噪声 ε。记为

$$P(192+i)=\sum_{j=0}^{191}p_{ff_j}(s_1s_2)+P_f+P_g+P_{\mathrm{xor}}+\varepsilon \tag{7}$$

其中 $p_{ff_j}(s_1s_2)$ 表示第 j 个触发器由 s_1 翻转到 s_2 所消耗的能量，用汉明距离模型进行刻画，$j\ge 1$。因此在 IV/key setup 阶段算法每个时钟的能量消耗主要集中在 192 个触发器的状态变换上。

本文提出的对于 DECIM 的能量攻击是针对

LFSR 的 IV/key setup 阶段。当 LFSR 在经过初始化后，空转 768 个时钟。在每个空转时钟内，LFSR 的状态将进行更新而不输出产生的密钥流。在自同步阶段，DECIM 算法的其他部件并不发生作用，因此采集到的能量只与 LFSR 有关。在上述 768 个时钟内，a_{192+i} 只与 80 bit K 和 64 bit IV 有关。在对于 DECIM 的攻击中，本文假定密钥 K 保持不变，随机选取不同的 64 bit IV，进行多次加密并且记录能量轨迹，每条能量轨迹中可以使用的部分为 768 个时钟。同时，可以通过对同一个 IV 进行多次加密，然后计算其期望来减少非算法噪声的影响。这是因为非算法噪声服从正态分布，其方差将会随着实验次数增加而减小。

3.3 DECIM 差分能量攻击算法

在攻击算法开始前，假定已经采集到了由 SET_IV 中不同 IV 和固定密钥 K 所对应的空转 768 个时钟的能量消耗，进而求得 767 轮的能量消耗差异 PD_{192+i}。DECIM 算法的反馈值 a_{192+i} 与 LFSR 的 28 个抽头有关，其中有 8 个抽头完全由 IV 决定，6 个由 K 和 IV 共同决定，14 个完全由 K 决定。任意两个相邻时钟的子密钥穷尽量为 2^{40}。DECIM 差分能量攻击算法中的符号定义同算法 1，不再赘述。

DECIM 差分能量攻击算法

for i=0 to 766

for $K\_\mathrm{num}=0$ to $2^{40}-1$

for IV_num=1 to d

步骤1　从 SET_IV 中选取一个 IV；

步骤2　根据 IV 和假定的子密钥 K_num 对 LFSR 的抽头位置进行填充；

步骤3　根据 $f(\cdot)$ 和 $g(\cdot)$，计算相邻两个时钟的 $s(n+i)$ 和 $s(n+i+1)$；

步骤4　计算选择方程

$$C=s(n+i)\oplus s(n+i+1) \tag{8}$$

当 C=0 时，将 PD_{192+i} 放入集合 SET_P0；

当 C=1 时，将 PD_{192+i} 放入集合 SET_P1；

结束

步骤5　计算差分公式

$$\mathrm{peak}=\frac{1}{|\mathrm{SET\_P0}|}\sum_{\mathrm{PD}_{192+i}\in\mathrm{SET\_P0}}(1-C)\cdot\mathrm{PD}_{192+i}-\frac{1}{|\mathrm{SET\_P1}|}\sum_{\mathrm{PD}_{192+i}\in\mathrm{SET\_P1}}C\cdot\mathrm{PD}_{192+i} \tag{9}$$

步骤6　选择最大的 peak 对应的子密钥 K_num 为第 i 轮的真子密钥；

结束

步骤7　返回最外层循环，寻找下一轮抽头所用到的子密钥。

结束

3.4 密钥穷尽规模

实际的攻击过程中每一轮穷尽的密钥量并没有达到 2^{40}。以第 1 轮为例来进行分析。在第 1 轮中用到 768 个空转时钟中的前两个，即第 192 和 193 个时钟。在第 192 个时钟内，根据式(5)可以求得与 $f(\cdot)$ 有关的密钥比特为 $K_0,K_3,K_4,K_{18},K_{23},K_{35},K_{36},K_{37},K_{60},K_{61},K_{66}$，与 $g(\cdot)$ 有关的密钥比特为 $K_1,K_{13},K_{24},K_{28},K_{31},K_{45},K_{54}$,;同理，在第 193 个时钟内，与 $f(\cdot)$ 有关的密钥比特为 $K_1,K_4,K_5,K_{19},K_{24},K_{36},K_{37},K_{38},K_{61},K_{62},K_{67}$，与 $g(\cdot)$ 有关的密钥比特为 $K_2,K_{14},K_{25},K_{29},K_{32},K_{46},K_{55},K_{65},K_{66}$。其中两个时钟重复的密钥比特为 $K_1,K_4,K_{24},K_{36},K_{37},K_{61},K_{65},K_{66}$，因此第 1 轮需要穷尽的密钥比特为 32 个。其他各轮穷尽的密钥比特如表 1 所示。

表 1 DECIM 差分能量攻击各轮需要穷尽的密钥

轮数	穷尽的密钥比特
0	$K_0,K_1,K_2,K_3,K_4,K_5,K_{13},K_{14},K_{18},K_{19},K_{23},K_{24},K_{25},K_{28},K_{29},K_{31},K_{32},K_{35},K_{36},K_{37},K_{38},K_{45},K_{46},K_{54},K_{55},K_{60},K_{61},K_{62},K_{64},K_{65},K_{66},K_{67}$
1	$K_6,K_{15},K_{20},K_{26},K_{30},K_{33},K_{39},K_{47},K_{56},K_{63},K_{68}$
2	$K_7,K_{16},K_{21},K_{27},K_{34},K_{40},K_{48},K_{57},K_{69}$
3	$K_8,K_{17},K_{22},K_{41},K_{49},K_{58},K_{70}$
4	$K_9,K_{42},K_{50},K_{59},K_{71}$
5	$K_{10},K_{43},K_{51},K_{72}$
6	$K_{11},K_{44},K_{52},K_{73}$
7	K_{12},K_{53},K_{74}
8	K_{75}

通过 9 轮的运算可以得到 76 bit 密钥信息，剩余的 4 bit 可以通过直接穷尽得到。对 DECIM 实施能量攻击，其总的密钥穷尽规模是 $2^{32}+2^{11}+2^9+2^7+2^5+2^4+2^4+2^3+2^1+2^4\approx O(2^{32})$。攻击者可以根据自身的能力，在能量攻击和暴力穷尽攻击间寻找一个最佳点来降低获得全部密钥的时间。

4　软件仿真与分析

在 Pentium4 1.5 GHz，384 MB 内存的微机上，本文使用软件仿真方法验证 DECIM 差分能量攻击算法的有效性，在整个过程中没有使用任何外部设备，如示波器，高精度电源。用 C 语言实现了 DECIM 算法及其能量攻击算法。对于攻击算法中涉及到的能量消耗差异 PD_{192+i}，按照如下方法进行了仿真，即当 $s(n+i)\oplus s(n+i+1)=0$ 时，则 $\mathrm{PD}_{192+i}=s(i)$

$\oplus s(i+1)$ ； 当 $s(n+i)\oplus s(n+i+1)=1$ 时， 则 $\mathrm{PD}_{192+i}=s(i)\oplus s(i+1)-1$。为使能量消耗差异表示方便， 设当 $s(n+i)\oplus s(n+i+1)=0$ 时， $\mathrm{PD}_{192+i}=1$；否则，$\mathrm{PD}_{192+i}=0$。将 SET_IV 的阶定为 32。计算前两个相邻时钟的能量消耗差异的密钥穷尽量为 2^{32}，理论峰值 peak 应当为 1。在上述硬件平台上，表 2 给出了对 4 组不同数据进行模拟攻击的实验结果。通过比较可得，选出比例平均为 4.169‰，每 10000 个密钥穷尽平均时间为 8.777 s，因此只恢复前两个时钟所用到的 32 bit 子密钥理论耗时 3,769,692 s，约 43.63 天，筛选出子密钥个数约为 17,905,718。

表 2 模拟攻击数据统计表

穷尽子密钥总数	选出子密钥个数	选出比例(‰)	耗时(s)
200,000	838	4.19	181.7
500,000	2,016	4.302	440.4
1,000,000	4,056	4.056	881.7
10,000,000	41,284	4.128	8399.9

下面分析筛选出的子密钥不唯一的原因。由式(6)可知，在每一个时钟所涉及到的 20 bit 子密钥中，有 11 个与 $f(\cdot)$ 有关，剩余抽头与 $g(\cdot)$ 有关。因为对于每一个假设的子密钥都用相同的 32 个 IV 来计算 $s(n+i)$，所以 IV 对 $s(n+i)$ 的影响被抵消。当 $f(\cdot)$ 抽头位置的汉明重量相同时， $f(\cdot)$ 对于 $s(n+i)$ 的影响被抵消，而对于布尔函数 $g(\cdot)$ 对 $s(n+i)$ 的分析较复杂，在这里不再论述。结合上述分析，可以看到符合峰值 $\mathrm{peak}=1$ 的子密钥并不唯一的原因。

5 结束语

本文给出了 LFSR 差分能量攻击的一般算法。该算法能够高效的去除算法噪声对攻击的影响，进而证明了差分能量攻击对于流密码的切实威胁。对 DECIM 的仿真结果表明，该算法能够为进一步分析工作缩小密钥穷尽空间。因此，该算法在分析以 LFSR 为驱动部件的流密码体制时具有较好的效果。为了更全面地对该算法进行评估，我们将依靠硬件模拟器或物理测试方法对其进行验证，进而将它广泛应用于流密码设计与研究工作中。

参 考 文 献

[1] Jean-Jacques Quisquater and Math RiZK. Side channel attacks. http://www.ipa.go.jp/security/enc/CRYPTREC/fy15 /doc/1047_Side_Channel_report.pdf, 2008.9.

[2] Zhou Yong-bin and Feng Deng-guo, *et al.*. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. http://eprint.iacr.org/ 2005 / 388.pdf, 2008.4.

[3] Courtois N T and Meier W, *et al.*. Algebraic attacks on stream ciphers with linear feedback [C]. Advances in Eurocrypt 2003, Warsaw Poland, Lecture Notes in Computer Science, May 4–8, 2003, Vol. 2656: 345–359.

[4] Kocher P C, Jae J, and Jun B, *et al.*. Differential power analysis [C]. CRYPTO' 99, Santa Barbara, CA, USA, Lecture Notes in Computer Science, Aug 15–19, 1999, Vol. 1666: 388–397.

[5] Gierlichs B, Batina L, and Clavier C, *et al.*. Susceptibility of eSTREAM candidates towards side channel analysis. http://www.ecrypt.eu.org/stream, 2008.3.

[6] ECRYPT. eSTREAM. http://www.ecrypt.eu.org/stream/, 2008.9.

[7] Lano J, Mentens N, and Preneel B, *et al.*. Power analysis of synchronous stream ciphers with resynchronization mechanism [C]. SASC Workshop, Novotel Brugge Centrum, Belgium, Workshop Record, Oct 14–15, 2004: 327–333.

[8] Burman S, Mukhopadhyay D, and Veezhinathan K, *et al.*. LFSR based stream ciphers are vulnerable to power attacks [C]. INDOCRYPT 2007, Chennai, INDIA, Lecture Notes in Computer Science, Dec 9–13, 2007, Vol. 4859: 384–392.

[9] Fischer W, Gammel B M, and Kniffler O, *et al.*. Differential power analysis of stream ciphers [C]. CT-RSA 2007, San Francisco, CA, USA, Lecture Notes in Computer Science, Feb 5–9, 2007, Vol. 4377: 257–270.

[10] Berbain C, Billet O, and Canteaut A, *et al.*. DECIMv2. http://www.ecrypt.eu.org/stream/decim/decim_p3.pdf, 2007.5.

[11] Kumar S, Lemke K, and Paar C, *et al.*. Some thoughts about implementation properties of stream ciphers [C]. SASC Workshop, Novotel Brugge Centrum, Belgium, Workshop Record, Oct 14–15, 2004: 311–319.

臧玉亮： 男，1982 年生，硕士生，研究方向为流密码能量攻击.

韩文报： 男，1963 年生，教授，博士生导师，主要研究方向为信息安全.