第3章　相关分析方法

本章内容提要

相关分析从对基于LFSR的序列密码的分析起步，最早是由Blaser等[1]提出的，但真正有价值的工作是由Siegenthaler[2]提出的非线性组合生成器的分别征服相关分析，其基本思想是利用组合函数的输出与输入分量或某些输入分量子集之和的相关性，穷举搜索某个特定LFSR的初始状态或者某几个LFSR的初始状态，而各个LFSR的初始状态就是非线性组合生成器的子密钥，这就是最早的相关分析。分别征服(divide and conquer)来源于一种图论算法，体现了“分而治之”的思想(因此也译为分治)，意为将一个待求解的问题分成许多子问题，然后对每个子问题求解，最后再综合求解。随后，Meier等[3]给出了加速上述分别征服相关分析的两个算法，即算法A和算法B，称为快速相关分析，其出发点是上述相关分析的复杂度与LFSR的长度成指数关系，因此，这种相关分析只适用于长度较短的LFSR。针对此问题，他们对LFSR的抽头数较少的非线性组合序列密码提出了一种使用概率迭代译码算法的快速相关分析方法，不需要搜索整个LFSR的所有可能初始状态，就能找出正确的初始状态，这个方法是相关分析发展的里程碑。之后，又陆续出现了一系列对相关分析核心思想的改进方法。例如，Zhang等[4]提出了多步快速相关分析方法，他们指出，以前的工作主要是把LFSR的初始状态看成一个整体，并且仅仅使用一种校验等式来进行译码，但实际上可以充分利用不同种类的校验等式，在不增加渐近复杂度的情况下，逐个部分地恢复初始状态，这种方法对反馈多项式的汉明重量没有要求和限制；Lee等[5]基于Anderson[6]对于采用满足某些密码学性质的滤波生成器的条件相关分析思想提出了条件相关分析的框架，其核心思想是考察增量函数在特定输出情况下输入变量的相关性。条件相关分析又被扩展到两种类型的分析，即混成相关攻击(hybird correlation attack)和集中攻击(concentration attack)[7]，这两种分析的目标都是通过条件相关分析和快速相关分析恢复LFSR未知的初始状态。Lu等[8]把条件相关分析扩展为在猜测部分未知输入的情况下考察向量函数输出的相关性，这里假定部分输入信息服从随机均匀分布，特别地，这个方法在分析蓝牙二级E0算法时被证明非常有效。在此基础上，Zhang等人[9]发展并提出了基于条件掩码的条件相关分析方法。

本章主要介绍分别征服相关分析、快速相关分析、多步快速相关分析、条件相关分析和熵漏分析5种方法。

本章重点

- 分别征服相关分析方法的基本原理。
- 快速相关分析方法的适用范围和基本原理。
- 多步快速相关分析方法的基本思想。
- 条件相关分析方法的基本思想。
- 熵漏分析方法的基本思想。
- 相关免疫阶的发展背景、基本概念和特征。

3.1　分别征服相关分析方法

本节主要介绍分别征服相关分析方法的统计模型、基本原理、应用实例和应对措施。

3.1.1　二元加法非线性组合序列密码模型

二元非线性组合生成器由 s 个线性反馈移位寄存器(LFSR)和一个非线性组合函数组成。s 个 LFSR 为非线性组合函数提供随机性较好的序列,通常为最大长度序列,即 m-序列;非线性组合函数主要用来提高密钥序列的线性复杂度。所谓二元加法非线性组合序列密码是指将二元非线性组合生成器的输出序列作为密钥流或密钥序列,将密钥序列与明文序列进行逐位模 2 加后所得的序列作为密文序列的密码,见图 3.1.1。

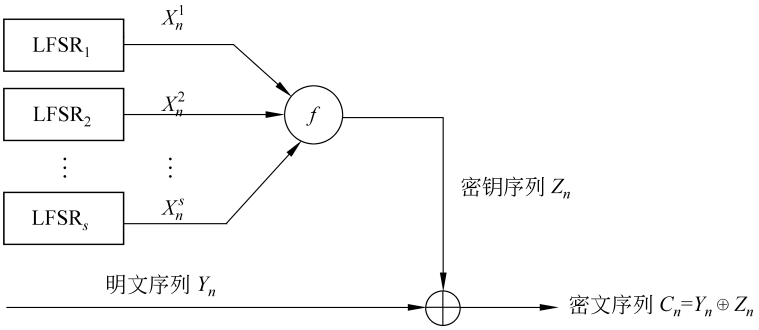

图 3.1.1　二元加法非线性组合序列密码

一个密码的密钥量是一个相对的概念,它依赖于密码设计者假定密码分析者知道该密码的参数的多少。对图 3.1.1 所示的密码而言,一般假定密码分析者仅知道如下参数:

(1) 足够长的密文序列(使用唯密文攻击方法)。

(2) 非线性组合函数 $f(x)$。

(3) 所有 LFSR 的级数 $r_i(1\leqslant i\leqslant s)$。

(4) 语言编码及语言统计特性。

假定密码分析者不知道所有 LFSR 的初始状态及其联结多项式。如果以 R_i 记 $F_2[x]$ 中所有次数为 r_i 的本原多项式,那么对密码分析者来说,第 i 个 LFSR 的未知参数有 $R_i(2^{ri}-1)$个(要去除每个 LFSR 的全零初始状态,因为全零初始状态产生全零序列),这部分密钥称为 LFSR$_i$ 的子密钥。因此,图 3.1.1 所示的密码的密钥量为 $\prod_{i=1}^{s}R_i(2^{ri}-1)$。如果使用穷举搜索密钥攻击方法,那么在最坏的情况下所有 $\prod_{i=1}^{s}R_i(2^{ri}-1)$ 个密钥都需要尝试一次。如果 r_i 足够大,那么穷举搜索方法所需计算是无法实现的。

这里最关键的问题是理论上需要多长的密文才能破译这个密码。

3.1.2　分别征服相关分析方法的基本原理

分别征服相关分析是一种唯密文攻击方法。因为 C_n 与 Z_n 和 Y_n 有关,而 Z_n 又与 X_n^i

有关，因而 C_n 间接地与 X_n^i 有关。这表明在一般情况下 C_n 中必定包含 X_n^i 的信息，从而含有 LFSR_i 的子密钥的信息。现在有两个问题：一个是密文 $C_1C_2\cdots C_N$ 中含 LFSR_i 的子密钥的信息量由什么参数确定，另一个是如何提取或间接地利用这些信息。

分别征服相关分析方法是利用某些输入 x_i 与输出 z 之间的相关性逐步确定每个 LFSR_i 的子密钥。为此，首先需要根据最大长度序列的统计特性建立一个统计模型，见图 3.1.2。设函数 f 的输入 $x_n^i(1\leqslant i\leqslant s)$ 是由一些相互独立且服从同一分布的随机变量 X_n^i 所产生的，且对所有的 i 和 n 都有 $P(X_n^i=0)=P(X_n^i=1)=1/2$。函数 f 生成相互独立且服从同一分布的随机变量 $Z_n=f(X_n^1,X_n^2,\cdots,X_n^s)$，且对所有的 n 都有 $P(Z_n=0)=P(Z_n=1)=1/2$。置 $P(Z_n=X_n^i)=q_i$。再假定明文是一个二元无记忆信源(Binary Memoryless Source，BMS)的输出，且 $P(Y_n=0)=p_0$。

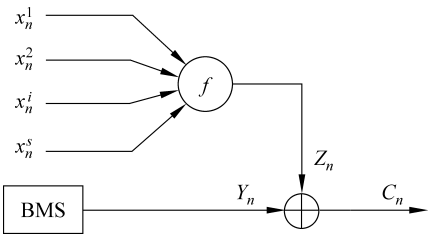

图 3.1.2　分别征服相关分析的统计模型

其次，定义 C_n 与 X_n^i 之间的相关性的一个估计——相关度。

定义 3.1.1　N 个密文符号 $C_1C_2\cdots C_N$ 与 $X_1^iX_2^i\cdots X_N^i$(LFSR_i 的输出序列，$i=1,2,\cdots,s$)之间的相关度(又称符合度)是一个随机变量 α，其定义如下：

$$\begin{aligned}\alpha&=(|\{j\mid C_j=X_j^i,j=1,2,\cdots,N\}|-|\{j\mid C_j\neq X_j^i,j=1,2,\cdots,N\}|)/N\\&=\sum_{n=1}^{N}(1-2(C_n\oplus X_n^i))/N\\&=1-2\sum_{n=1}^{N}(C_n\oplus X_n^i)/N\end{aligned}$$

α 就是 C_n 与 X_n^i 之间的相关性的一个估计。

C_n 与 X_n^i 之间的符合率为

$$\begin{aligned}p_e&=P(C_n=X_n^i)=P(C_n\oplus X_n^i=0)\\&=P(Z_n=X_n^i)P(Y_n=0)+P(Z_n\neq X_n^i)P(Y_n=1)\\&=q_ip_0+(1-q_i)(1-p_0)=1-(q_i+p_0)+2p_0q_i\end{aligned}$$

因此，

$$P(C_n\neq X_n^i)=P(C_n\oplus X_n^i=1)=1-p_e$$

显然，p_e 是关于 q_i 和 p_0 的对称函数。p_e 越大，说明 C_n 与 X_n^i 之间的符合率就越大，从而密文序列段中含有 LFSR_i 的子密钥的信息量就越大。当 p_e 接近于 1 时，密文序列段与对应的 LFSR_i 的输出段近似相同，从而说明该密码是极不安全的。从 p_e 和 α 的定义和表达式可以看到，密文序列段 $C_1C_2\cdots C_N$ 中所含 LFSR_i 的子密钥的信息量由明文特性 p_0、Z_n 和 X_n^i 的符合率 q_i 以及密文长度 N 所决定。这就回答了本节开头提出的第一个问题。

现在来讨论随机变量 α 的分布。对任意给定的 $i(1\leqslant i\leqslant s)$，可将 $C_n\oplus X_n^i(n=1,2,$

3,…)视作一些相互独立且服从同一分布的二元随机变量。因而,随机变量 $\beta=\sum_{n=1}^{N}(C_n \oplus X_n^i)$ 服从二项分布,其均值(也称期望值或数学期望)m_β 和方差 σ_β^2 分别为

$$m_\beta = N(1-p_e),\quad \sigma_\beta^2 = Np_e(1-p_e)$$

因此,随机变量 α 的均值 m_α 和方差 σ_α^2 分别为

$$m_\alpha = 1-2(1-p_e) = 2p_e-1,\quad \sigma_\alpha^2 = 4p_e(1-p_e)/N$$

设 X_n^0 是一个独立于 $X_n^i(i=1,2,\cdots,s)$ 的随机变量,且相互独立同分布,即 $P(X_n^0=0)=P(X_n^0=1)=1/2$。由于 Z_n 与 X_n^0 统计独立,所以 $q_0=P(Z_n=X_n^0)=1/2$,从而 $p_e=1/2$,此时 $m_\alpha=0$,$\sigma_\alpha^2=1/N$。

由中心极限定理可知,当 N 足够大时,随机变量 α 服从均值为 m_α、方差为 σ_α^2 的正态分布。

在分别征服相关分析中,首先需要确定 LFSR_i 的子密钥。为此,对一个级数为 r_i 的 LFSR_0(用于检测),任选一个初态,从 R_i 个可能的反馈多项式中任选一个,由该 LFSR_0 产生 N 个符号,再用这 N 个符号与 N 个密文符号一起计算出相关度 α 的一个确切值 α_0,它表现出以下两种情形的假设:

H_1:LFSR_0 的这 $N(>r_i)$ 个符号与 LFSR_i 所对应的 N 个符号一致,这种情形对应的 α_0 表现的是 C_n 和 $X_n^i(1\leqslant i\leqslant s)$ 的相关性。

H_0:LFSR_0 的这 $N(>r_i)$ 个符号与 LFSR_i 所对应的 N 个符号不一致(至少有一个不同),这种情形对应的 α_0 表现的是 C_n 和 X_n^0 的相关性。

为了对假设进行检验,必须利用相关度 α_0 的值。为了对检验结果给出一个判决,必须对两个假设 H_0 和 H_1 设定一个判决门限值 T,使得当 $\alpha_0<T$ 时,接受 H_0;当 $\alpha_0\geqslant T$ 时,接受 H_1。设 H_0 所对应的概率密度分布函数为 $p_{\alpha|H_0}(x)$,H_1 所对应的概率密度分布函数为 $p_{\alpha|H_1}(x)$,由中心极限定理可知,当 N 足够大时,$p_{\alpha|H_0}$ 是均值为 $m_0=0$、方差为 $\sigma_{02}=1/N$ 的正态分布函数,即

$$p_{\alpha|H_0} = \frac{1}{\sqrt{2\pi}\sigma_0}\mathrm{e}^{-(x-m_0)^2/2\sigma_0^2}$$

$p_{\alpha|H_1}$ 是均值为 $m_1=2p_e-1$、方差为 $\sigma_1^2=4p_e(1-p_e)/N$ 的正态分布函数,即

$$p_{\alpha|H_1} = \frac{1}{\sqrt{2\pi}\sigma_1}\mathrm{e}^{-(x-m_1)^2/2\sigma_1^2}$$

当 $q_i=1/2$ 或 $p_0=1/2$ 时,$p_e=1/2$。此时,$p_{\alpha|H_0}=p_{\alpha|H_1}$,在这种情形下,无法进行判决。

假设检验的计算工作量依赖于错误判决的数目。错误判决分为两类:一类是由事件 $\alpha\geqslant T|H_0$ 所引起的,称它为假真错误,即把假的参数判决为真的;另一类是由事件 $\alpha<T|H_1$ 所引起的,称它为真假错误,即把真的参数判决为假的。这些错误判决的次数主要由密码体制自身的参数 p_0 和 q_i(即密码本身的强度)以及使用的密文长度决定。我们主要感兴趣的是假真错误的概率 $P(\alpha\geqslant T|H_0)=P_\mathrm{f}$,但是为了确定判决门限值,还必须考虑真假错误的概率 $P(\alpha<T|H_1)=P_\mathrm{m}$,其中,

$$P_\mathrm{f}=\int_T^\infty p_{\alpha|H_0}(x)\mathrm{d}x,\quad P_\mathrm{m}=\int_{-\infty}^T p_{\alpha|H_1}(x)\mathrm{d}x$$

引入如下函数(称为错误函数):

$$Q(x)=\frac{1}{\sqrt{2\pi}}\int_x^{\infty}\mathrm{e}^{-y^2/2}\mathrm{d}y$$

则有

$$P_{\mathrm{f}}=Q(|T\sqrt{N}|),\quad P_{\mathrm{m}}=Q\left[\left|\frac{(2p_{\mathrm{e}}-1)-T}{2\sqrt{p_{\mathrm{e}}(1-p_{\mathrm{e}})}}\sqrt{N}\right|\right]$$

记 $\gamma_0=\dfrac{(2p_{\mathrm{e}}-1)-T}{2\sqrt{p_{\mathrm{e}}(1-p_{\mathrm{e}})}}\sqrt{N}$,于是有

$$T\sqrt{N}=\sqrt{N}(2p_{\mathrm{e}}-1)-\gamma_0 2\sqrt{p_{\mathrm{e}}(1-p_{\mathrm{e}})}$$

从而有

$$P_{\mathrm{m}}=Q(|\gamma_0|),\quad P_{\mathrm{f}}=Q\left(\left|\sqrt{N}(2p_{\mathrm{e}}-1)-2\gamma_0\sqrt{p_{\mathrm{e}}(1-p_{\mathrm{e}})}\right|\right)$$

算法 3.1.1 给出了攻击图 3.1.1 所示的序列密码的分别征服相关分析方法。

算法 3.1.1

第 1 步:由函数 f 确定概率 $q_i(i=1,2,\cdots,s)$,由明文编码和语言统计特性确定 p_0,并计算符合率 $p_{\mathrm{e}}=1-(p_0+q_i)+2p_0q_i$。

第 2 步:选定 P_{m},由关系式 $P_{\mathrm{m}}=Q(|\gamma_0|)$确定 γ_0,从而假真错误概率仅仅是密文个数 N 的函数。

第 3 步:确定 LFSR_i 的子密钥。选择 R_i 个可能的反馈多项式中的一个,并任选一个初始状态,进而生成一个周期为 $2^{ri}-1$ 的最大长度序列$\{S_i\}$。对$\{S_i\}$的 $2^{ri}-1$ 个可能位置中的每一个位置和 N 个密文符号计算相关度α,对每个事件 $\alpha\geqslant T$,假定所使用的反馈多项式和位置正确,从而 LFSR_i 的子密钥被确定。

由于事件$\alpha\geqslant T|H_0$ 以概率 P_{f} 发生,所以这里的判决可能是错误的。因此,对于使$\alpha\geqslant T$ 的所有位置,需要用新密文段进行附加检测。

如果对所有的 $2^{ri}-1$ 个位置,H_1 均被拒绝,则可认为选择的反馈多项式不对。当然,也有可能出现多项式是正确的情形,这种事件 $\alpha<T|H_1$ 发生的概率 P_{m} 事先可以控制得很小。因此,在 R_i 个可能的反馈多项式中选择一个新的,再重复上述过程。在最坏的情况下,所有 $2^{ri}-1$ 个位置和所有可能的 R_i 个反馈多项式都需要被检测,因而 LFSR_i 的子密钥大约需 R_i2^{ri} 次检测。

假真错误($\alpha\geqslant T|H_0$)的次数(从而所需要的检测次数)依赖于所使用的密文符号的个数 N。如果选择 N_1 使得 $P_{\mathrm{f}}=1/R_i2^{ri}$,那么在所有的约 R_i2^{ri} 个基本检测中,假真错误的次数的期望值为 1,并且要找到 LFSR_i 的子密钥所需的全部检测次数约为 R_i2^{ri}。选择 $N>N_1$一般不能降低需要的检测次数。

可证明,$Q(x)$满足如下关系:

$$\left(\sqrt{2\pi}x\right)^{-1}\mathrm{e}^{-\frac{x^2}{2}}(1-x^2)<Q(x)<\left(\sqrt{2\pi}x\right)^{-1}\mathrm{e}^{-\frac{x^2}{2}},\quad x\geqslant 0$$

显然使用 $Q(x)$的上界函数和下界函数中的任何一个即可得到 N_1 的精确估计,但遗憾的是,它们两个都不便于使用。现在利用另一个上界函数 $Q(x)<\dfrac{1}{2}\mathrm{e}^{-x^2/2}(x\geqslant 0)$来估计 N_1。

由于

$$P_f=\frac{1}{R_i 2^{ri}}=Q\left(\left|\sqrt{N}(2p_e-1)-2\gamma_0\sqrt{p_e(1-p_e)}\right|\right)$$

所以

$$\frac{1}{R_i 2^{ri}}<\frac{1}{2}e^{[(\sqrt{N_1}(2p_e-1)-2\gamma_0\sqrt{p_e(1-p_e)})]^2/2}$$

于是

$$N_1<\left[\frac{2^{-1/2}\sqrt{\ln(R_i 2^{ri-1})}+\gamma_0\sqrt{p_e(1-p_e)}}{p_e-\frac{1}{2}}\right]^2$$

上式中的上界可以近似地用来估计分别征服相关分析中攻击每个 LFSR_i 的子密钥所需要的密文符号个数。由于 $0\leqslant\sqrt{p_e(1-p_e)}\leqslant 1/2$ 和 $\sqrt{\ln(R_i 2^{ri-1})}$ 随 R_i 和 r_i 增长得很慢，因而，N_1 近似地随 $\left(\frac{1}{2}-p_e\right)^{-2}$ 增长。

由上述的讨论过程可知，当图 3.1.1 所示的序列密码的非线性组合函数与其某些输入分量存在相关性时，特别地，当输出 Z 与输入 X_i 相关时，相关分析方法利用这种相关性可通过大约 $R_i 2^{ri}$ 次检测独立于 LFSR_j（$j=1,2,\cdots,s,j\neq i$）找到 LFSR_i 的子密钥，利用找到的伪随机生成器的子密钥，可将其密钥搜索量从 $\prod_{i=1}^{s}R_i(2^{r_i}-1)$ 降低到大约 $\sum_{i=1}^{s}R_i 2^{ri}$。

3.1.3　分别征服相关分析方法的应用实例

Geffe 序列生成器是 Geffe 于 1973 年提出的[10]，原始的 Geffe 序列生成器可描述为 $C=A_1A_2\oplus\overline{A}_1A_3$，其中 A_1、A_2 和 A_3 是 3 个 m-序列，已知其反馈多项式分别为 $f_1(x)=1\oplus x^4\oplus x^{39}$、$f_2(x)=1\oplus x^3\oplus x^{20}$ 和 $f_3(x)=1\oplus x^3\oplus x^{17}$，但 3 个寄存器的状态未知，见图 3.1.3。

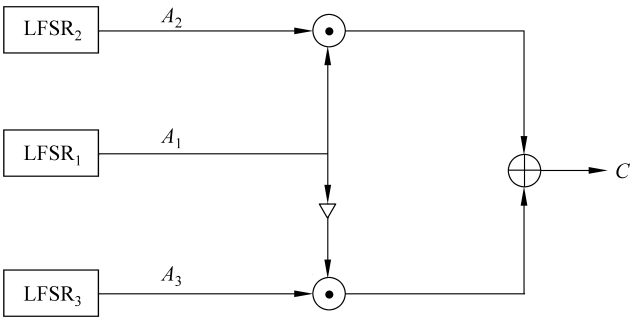

图 3.1.3　Geffe 序列生成器

虽然 Geffe 序列生成器生成的序列具有周期长、线性复杂度高、统计特性好等许多优点，但该序列生成器是密码学上弱的，针对该序列生成器已有很多分析方法。本节以 Geffe 序列生成器为例，说明应用分别征服相关分析方法分析密码算法的过程。

Geffe 序列生成器的非线性组合函数可表示为

$$f(x_1,x_2,x_3)=x_3\oplus x_1x_2\oplus x_1x_3=x_1x_2\oplus\bar{x}_1x_3$$

对 Geffe 序列生成器，参数 $q_i(i=1,2,3)$ 分别为 $q_1=p(f(x)=x_1)=0.5$、$q_2=p(f(x)=x_2)=0.75$ 和 $q_3=p(f(x)=x_3)=0.75$。分别征服相关分析方法无法攻击 $LFSR_1$，然而，在利用算法 3.1.1 找到 $LFSR_2$ 和 $LFSR_3$ 的子密钥部分后，基于该密码的特点，可通过 A_2 和 A_3 来确定 $LFSR_1$ 的子密钥。实际上，这里也渗透了一种分析方法，我们应掌握这种分析技巧。

由于

$$x_i^1=\begin{cases}c_i, & x_i^2=1 \wedge x_i^3=0\\ \bar{c}_i, & x_i^2=0 \wedge x_i^3=1\end{cases}$$

因此，当 A_2、A_3 确定之后，上述关系式就给出了 A_1 与 C 的相关性。但 $P(x_i^2 \oplus x_i^3=1)=\frac{1}{2}$，所以利用 C 的一个长为 k 位的截段就可以确定 A_1 的 $s=k/2$ 位：$x_{i_1}^1, x_{i_2}^1, \cdots, x_{i_s}^1$。为了利用这些位来计算 A_1 的初态，设 α 是 $f_1(x)$ 的互反多项式 $\overline{f_1(x)}$ 的一个根，并将 A_1 的输出位表示成如下形式(利用定理 1.4.1)：

$$x_i^1=\mathrm{Tr}(\beta\alpha^i), \quad \beta \in F_{2^{39}}$$

因此，如果对某个 i_k 有 $\alpha^{i_k}=c_{0,k} \oplus c_{1,k}\alpha \oplus c_{2,k}\alpha^2 \oplus \cdots \oplus c_{38,k}\alpha^{38}$(这里使用的是第二种递归关系式)，那么就可以得到线性方程 $c_{0,k}x_0^1 \oplus c_{1,k}x_1^1 \oplus \cdots \oplus c_{38,k}x_{38}^1=x_{i_k}^1$ 来制约 A_1 的初态：

$$s_0^1=(x_0^1, x_1^1, \cdots, x_{38}^1)$$

记 $M=(c_{j,k})_{39\times s}$，则有

$$(x_0^1, x_1^1, \cdots, x_{38}^1)M=(x_{i_1}^1, x_{i_2}^1, \cdots, x_{i_s}^1)$$

从 M 中任取一个 39 阶非奇异子方阵，都可利用这一关系式解出 A_1 的初态 s_0^1。当 $s=49$ 时，在 M 中可以找到这样一个非奇异子方阵的概率在 99%以上。由此可见，当 A_2、A_3 均已确定时，为了确定 A_1，利用已知的一个 100b 长的截取段已足够了。

3.1.4 应对分别征服相关分析方法的措施

由 3.1.2 节中的分析过程可知，要想让分别征服相关分析方法对非线性组合序列密码不可行，必须使得 N_1 很大；而要使 N_1 很大，必须使得 p_e 接近 1/2。特别地，当 q_i 接近 1/2 时，p_e 就接近 1/2。因此，得到了选择非线性组合密码函数的一条准则。

准则 3.1.1 要使非线性组合序列密码可抵抗分别征服相关分析方法，必须尽可能地选择使得所有 q_i 接近 1/2 的非线性组合函数。

为了对抗分别征服相关分析方法，Siegenthaler 提出了布尔函数的相关免疫阶的概念[11]，用于度量和刻画非线性组合序列密码抵抗分别征服相关分析的能力。当准则 3.1.1 中的所有 $q_i=1/2$ 时，就是 Siegenthaler 提出的 1 阶相关免疫的概念。自从相关免疫阶这个概念提出之后，人们对其进行了大量系统、深入的研究[12-15]，包括结构特征刻画、构造、计数以及次数与相关免疫阶的折中关系等。

定义 3.1.2 设 $f(x)$：$F_n^2 \to F_2$，$x_1, x_2, \cdots, x_n$ 是 F_2 上的独立的、均匀分布的随机变量，如果对任意的 $(a_1, a_2, \cdots, a_m) \in F_2^m (m \leqslant n)$ 及 $a \in F_2$，都有

$$p(f=a, x_{i_1}=a_1, x_{i_2}=a_2, \cdots, x_{i_m}=a_m)=\frac{1}{2^m}p(f=a)$$

则称 f 与变量 $x_{i_1}, x_{i_2}, \cdots, x_{i_m}$ 统计无关。如果 f 与 $x_1, x_2, \cdots, x_n$ 中的任意 m 个变量都统

计无关,则称 f 是 m 阶相关免疫的。

特别地,如果 f 既是平衡的又是 m 阶相关免疫的,也称 f 是 m 阶弹性的。

定理 3.1.1 给出了 f 与其变量 $x_{i_1},x_{i_2},\cdots,x_{i_m}$ 统计无关的一些等价条件。

定理 3.1.1 设 $f(x)$如定义 3.1.2 中所述,则下列 3 个条件等价:

(1) $f(x)$与变量 $x_{i_1},x_{i_2},\cdots,x_{i_m}$ 统计无关。

(2) 对任意的 $w=(0,\cdots,w_{i_1},\cdots,w_{i_2},\cdots,w_{i_m},\cdots,0)\in F_2^n$,$1\leqslant W_H(w)\leqslant m$,$f(x)$与 $w\cdot x$ 统计无关。

(3) 对任意的 $w=(0,\cdots,w_{i_1},\cdots,w_{i_2},\cdots,w_{i_m},\cdots,0)\in F_2^n$,$1\leqslant W_H(w)\leqslant m$,$f(x)+w\cdot x$ 是平衡的。

证明:(1)⇒(2)。显然成立。

(2)⇒(3)。设对任意的 $w=(0,\cdots,w_{i_1},\cdots,w_{i_2},\cdots,w_{i_m},\cdots,0)\in F_2^n$,$1\leqslant W_H(w)\leqslant m$,$f(x)$与 $w\cdot x$ 统计无关,则对任意的 $i\in F_2$,有

$$\begin{aligned}p(f(x)+w\cdot x=i)&=\sum_{a\in F_2}p(f(x)=a,w\cdot x=i-a)\\&=\sum_{a\in F_2}p(f(x)=a)p(w\cdot x=i-a)\\&=\frac{1}{2}\sum_{a\in F_2}p(f(x)=a)=\frac{1}{2}\end{aligned}$$

从而有,

$$W_H(f+w\cdot x)=|\{x\in F_2^n\mid f(x)+w\cdot x=1\}|=2^n\times\frac{1}{2}=2^{n-1}$$

故 $f(x)+w\cdot x$ 是平衡的。

(3)⇒(1)。设对任意的 $w=(0,\cdots,w_{i_1}\cdots,,w_{i_2},\cdots,w_{i_m},\cdots,0)\in F_2^n$,$1\leqslant W_H(w)\leqslant m$,$f(x)+w\cdot x$ 是平衡的。对任意的 $a,a_1,a_2,\cdots,a_m\in F_2$,记

$$\begin{aligned}&A=(a,a_1,a_2,\cdots,a_m)\\&F(x)=(f(x),x_{i_1},x_{i_2},\cdots,x_{i_m})\\&N_A=|\{x\in F_2^n\mid F(x)=A\}|\\&N_a=|\{x\in F_2^n\mid f(x)=a\}|\end{aligned}$$

因为

$$\begin{aligned}\sum_{x\in F_2^n}\sum_{y\in F_2^{m+1}}(-1)^{A\cdot y+F(x)\cdot y}&=\sum_{y\in F_2^{m+1}}(-1)^{A\cdot y}\sum_{x\in F_2^n}(-1)^{F(x)\cdot y}\\&=2^n+\sum_{y\in F_2^{m+1}\setminus\{0\}}(-1)^{A\cdot y}\sum_{x\in F_2^n}(-1)^{F(x)\cdot y}\end{aligned}$$

利用假设条件可知,当 $y\neq(0,0,\cdots,0),(1,0,\cdots,0)$时,有

$$\sum_{x\in F_2^n}(-1)^{F(x)\cdot y}=0$$

所以

$$\begin{aligned}\sum_{x\in F_2^n}\sum_{y\in F_2^{m+1}}(-1)^{A\cdot y+F(x)\cdot y}&=2^n+(-1)^a\sum_{x\in F_2^n}(-1)^{f(x)}\\&=2^n+\sum_{x\in F_2^n}(-1)^{f(x)+a}=2N_a\end{aligned}$$

又

$$\sum_{x\in F_2^n}\sum_{y\in F_2^{m+1}}(-1)^{A\cdot y+F(x)\cdot y}=\sum_{x\in F_2^n}\sum_{y\in F_2^{m+1}}(-1)^{(A+F(x))\cdot y}=N_A\cdot 2^{m+1}$$

故由上述两式可得 $N_A\cdot 2^m=N_a$，即

$$p(f=a,x_{i_1}=a_1,x_{i_2}=a_2,\cdots,x_{i_m}=a_m)$$

$$=\frac{1}{2^m}p(f=a)=p(f=a)p(x_{i_1}=a_1)\cdots p(x_{i_m}=a_m)$$

由 A 的任意性可知，$f(x)$与变量 $x_{i_1},x_{i_2},\cdots,x_{i_m}$ 统计无关。

引理 3.1.1 设 $f(x)$如定义 3.1.2 中所述，$f(x)$与变量 $x_{i_1},x_{i_2},\cdots,x_{i_m}$ 统计无关，则 $W_{\mathrm{H}}(f)=2^m k_0$，$k_0$ 为非负整数。

证明：因为 $f(x)$与变量 $x_{i_1},x_{i_2},\cdots,x_{i_m}$ 统计无关，因此，

$$P(f=1\mid x_{i_1},x_{i_2},\cdots,x_{i_m})=P(f=1)$$

而

$$P(f=1\mid x_{i_1},x_{i_2},\cdots,x_{i_m})=\frac{W_{\mathrm{H}}(f')}{2^{n-m}}$$

$$P(f=1)=\frac{W_{\mathrm{H}}(f)}{2^n}$$

所以

$$\frac{W_{\mathrm{H}}(f)}{2^{n-m}}=\frac{W_{\mathrm{H}}(f)}{2^n}$$

即

$$W_{\mathrm{H}}(f)=2^m W_{\mathrm{H}}(f')=2^m k_0$$

其中 f'表示给定 $x_{i_1}=c_{i_1},x_{i_2}=c_{i_2},\cdots,x_{i_m}=c_{i_m}$ 的条件下，f 关于 $n-m$ 个变量$\{x_1,x_2,\cdots,x_n\}\backslash\{x_{i_1},x_{i_2},\cdots,x_{i_m}\}$的函数，$k_0=W_{\mathrm{H}}(f')$。

Walsh 变换(也称 Walsh 谱)是研究布尔函数的一个强有力的工具。下面给出相关概念。

定义 3.1.3 设 $x=(x_1,x_2,\cdots,x_n)$，$w=(w_1,w_2,\cdots,w_n)\in F_2^n$，$x$ 和 w 的点积定义为 $w\cdot x=w_1x_1\oplus w_2x_2\oplus\cdots w_nx_x\in F_2$。$n$ 个变量的布尔函数 $f(x)$的 Walsh 变换定义为

$$S_f(w)=2^{-n}\sum_{x\in F_2^n}f(x)(-1)^{w\cdot x}$$

其逆变换为

$$f(x)=\sum_{x\in F_2^n}S_f(w)(-1)^{w\cdot x}$$

上式中将 $f(x)$视作实数，求和是指实数求和。$f(x)$的循环 Walsh 谱定义为

$$S_f(w)=2^{-n}\sum_{x\in F_2^n}(-1)^{f(x)}(-1)^{w\cdot x}$$

其逆变换为

$$f(x)=\frac{1}{2}-\frac{1}{2}\sum_{x\in F_2^n}S_{(f)}(w)(-1)^{w\cdot x}$$

由两种谱的定义，并注意到$(-1)^{f(x)}=1-2f(x)$，直接可推出两种谱有如下关系：

$$S_{(f)}(w)=\begin{cases}-2S_f(w), & w\neq 0\\ 1-2S_f(w), & w=0\end{cases}$$

这里需要说明的是,有的文献中将 $f(x)$的 Walsh 变换定义为 $H_f(w)=2^nS_f(w)$或 $H_{(f)}(w)=2^nS_{(f)}(w)$,二者之间只差一个常数因子 2^n,无本质差别,实际应用中究竟选用哪种定义方式可根据具体应用环境而定。为简单起见,特定的场景下也可省去下标。有的文献中为了方便起见,也将 $f(x)$的 Walsh 变换 $H_f(w)$记为$\hat{f}(w)$,即$\hat{f}(w)=H_f(w)$。

给定 $F:F_2^n\to F_2$,若将 f 的 Walsh 变换$\hat{f}$定义为

$$\hat{f}(w)=2^nS_f(w)=\sum_{x\in F_2^n}f(x)(-1)^{w\cdot x}$$

则其逆变换为

$$f(x)=2^{-n}\sum_{w\in F_2^n}\hat{f}(w)(-1)^{w\cdot x}$$

再设 $g:F_2^n\to F_2$,将 f 和 y 的卷积(用⊗表示)定义为

$$(f\otimes g)(a)=\sum_{b\in F_2^n}f(b)\cdot g(a\oplus b),a\in F_2^n$$

利用定义可直接证明,卷积和 Walsh 变换是可转换的,即

$$\widehat{f\otimes g}(a)=\hat{f}(a)\cdot\hat{g}(a),a\in F_2^n$$

这样,为了计算卷积函数$(f\otimes g)(a)$,就可以首先分别完成 f 和g 的 Walsh 变换,然后把它们相乘,最后使用逆 Walsh 变换。计算 Walsh 变换有快速算法[13,14],称之为快速 Walsh 变换(Fast Walsh Transformation,FWT),FWT 的时间和存储复杂度分别为 $O(n2^n)$和 $O(2^n)$。

下面简要介绍快速计算 Walsh 变换的基本思路。

设 $\boldsymbol{f}(x)=(f(0),f(1),\cdots,f(2^n-1))$,$\boldsymbol{S}_f(w)=(S_f(0),S_f(1),\cdots,S_f(2^n-1))$,则 $\boldsymbol{S}_f(w)=2^{-n}\boldsymbol{f}(x)\boldsymbol{H}_n$。其中 $\boldsymbol{H}_n$ 由下式迭代地定义:

$$\boldsymbol{H}_0=[1]$$

$$\boldsymbol{H}_n=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\otimes\boldsymbol{H}_{n-1}=\begin{bmatrix}H_{n-1} & H_{n-1}\\ H_{n-1} & -H_{n-1}\end{bmatrix}$$

⊗表示矩阵的 Keronecker 积。因为 $\boldsymbol{H}_n^2=2^n\boldsymbol{I}_n$,所以 Walsh 逆变换为

$$\boldsymbol{f}(x)=\boldsymbol{S}_f(w)\boldsymbol{H}_n$$

设 $\boldsymbol{f}^1(x)$和 $\boldsymbol{f}^2(x)$分别表示 $\boldsymbol{f}(x)$的前一半和后一半,则

$$\boldsymbol{S}_f(w)=2^{-n}\boldsymbol{f}(x)\boldsymbol{H}_n=2^{-n}(\boldsymbol{f}^1(x)\boldsymbol{H}_{n-1}+\boldsymbol{f}^2(x)\boldsymbol{H}_{n-1},\boldsymbol{f}^1(x)\boldsymbol{H}_{n-1}-\boldsymbol{f}^2(x)\boldsymbol{H}_{n-1})$$

直至迭代到 $\boldsymbol{H}_0$ 为止。

由定理 3.1.1 立即可推出 $f(x)$与变量 $x_{i_1},x_{i_2},\cdots,x_{i_m}$统计无关的谱特征。

定理 3.1.2 设 $f(x)$如定义 3.1.2 中所述,则 $f(x)$与变量 $x_{i_1},x_{i_2},\cdots,x_{i_m}$统计无关,当且仅当对任意的 $w=(0,\cdots,w_{i_1},\cdots,w_{i_2},\cdots,w_{i_m},\cdots,0)\in F_2^n$,$1\leqslant W_{\mathrm{H}}(w)\leqslant m$,$S_{(f)}(w)=0$。

证明:由定理 3.1.1 可知,$f(x)$与变量 $x_{i_1},x_{i_2},\cdots,x_{i_m}$统计无关,当且仅当对任意的 $w=(0,\cdots,w_{i_1},\cdots,w_{i_2},\cdots,w_{i_m},\cdots,0)\in F_2^n$,$1\leqslant W_{\mathrm{H}}(w)\leqslant m$,$f(x)+w\cdot x$ 是平衡的;而 $f(x)+w\cdot x$ 是平衡的,当且仅当 $S_{(f+w\cdot x)}(0)=S_{(f)}(w)=0$。定理 3.1.2 得证。

由定理 3.1.1 和定理 3.1.2 可得到如下两个定理。

定理 3.1.3 设 $f(x)$如定义 3.1.2 中所述,则下列 3 个条件等价:

(1) $f(x)$是 m 阶相关免疫的。

(2) 对任意的 $w\in F_2^n$,$1\leqslant W_{\mathrm{H}}(w)\leqslant m$,$f(x)$与 $w\cdot x$ 统计无关。

(3) 对任意的 $w\in F_2^n$,$1\leqslant W_{\mathrm{H}}(w)\leqslant m$,$f(x)+w\cdot x$ 是平衡的。

定理 3.1.4 设 $f(x)$如定义 3.1.2 中所述,则 $f(x)$是 m 阶相关免疫的,当且仅当对任意的 $w\in F_2^n$,$1\leqslant W_{\mathrm{H}}(w)\leqslant m$,$S_f(w)=0$。

由定理 3.1.4 和两种谱之间的关系立即可推出以下定理。

定理 3.1.5[12] 设 $f(x)$如定义 3.1.2 中所述,则 $f(x)$是 m 阶相关免疫的,当且仅当对任意的 $w\in F_2^n$,$1\leqslant W_{\mathrm{H}}(w)\leqslant m$,$S_f(w)=0$。

定理 3.1.5 即著名的 Xiao-Massey 定理。

定理 3.1.6 给出了构造相关免疫函数的一个递归方法。

定理 3.1.6 设 f_1 和 f_2 是两个 n 个变量的 m 阶相关免疫函数,令 $f(x_1,x_2,\cdots,x_{n+1})=x_{n+1}f_1(x_1,x_2,\cdots,x_n)\oplus\overline{x_{n+1}}f_2(x_1,x_2,\cdots,x_n)$,则 f 是一个有 $n+1$ 个变量的 m 阶相关免疫函数。次数$\partial^0 f=\max\{\partial^0 f_1,\partial^0 f_2\}+1$。

易知,$f:F_2^n\to F_2$ 是 $n-1$ 阶相关免疫函数的充要条件是

$$f(x)=x_1\oplus x_2\oplus\cdots\oplus x_n\oplus c,\quad c\in F_2$$

最后,讨论相关免疫阶和非线性次数之间的关系。

设 $f(x):F_2^n\to F_2$ 的多项式表示为式(1.4.8)。现在用 $f(x)$的循环 Walsh 谱来表示式(1.4.8)中的系数,将由分量下标 $i_1i_2\cdots i_r$ 指定的 r 维及 $n-r$ 维子空间记为

$$S_{i_1i_2\cdots i_r}=\{x\in F_2^n\mid x_j=0,\text{对所有的 } j\notin\{i_1,i_2,\cdots,i_r\}\}$$

$$\overline{S}_{i_1i_2\cdots i_r}=\{x\in F_2^n\mid x_j=0,\text{对所有的 } j\in\{i_1,i_2,\cdots,i_r\}\}=S_{i_1i_2\cdots i_r}^{\perp}$$

在式(1.4.8)中,除系数为 $a_{i_1i_2\cdots i_r}$ 的项之外,其余各项在 $S_{i_1i_2\cdots i_r}$ 上模 2 求和的结果均为 0,因此,有

$$\begin{aligned}a_{i_1i_2\cdots i_r}&=\sum_{x\in S_{i_1i_2\cdots i_r}}f(x)=\sum_{x\in S_{i_1i_2\cdots i_r}}\left(\frac{1}{2}-\frac{1}{2}\sum_{x\in F_2^n}S_{(f)}(w)(-1)^{w\cdot x}\right)\\&=-\frac{1}{2}\sum_{x\in F_2^n}S_{(f)}(w)\sum_{x\in S_{i_1i_2\cdots i_r}}(-1)^{w\cdot x}\qquad(\mathrm{mod}\ 2)\\&=-\frac{1}{2}\sum_{x\in S_{i_1i_2\cdots i_r}^{\perp}}S_{(f)}(w)\cdot 2^r\qquad(\mathrm{mod}\ 2)\\&=-2^{r-1}\sum_{x\in S_{i_1i_2\cdots i_r}}S_{(f)}(w)\qquad(\mathrm{mod}\ 2)\end{aligned}\tag{3.1.1}$$

$\overline{S}_{i_1i_2\cdots i_r}$ 中的 w 的汉明重量 $W_{\mathrm{H}}(w)\leqslant n-r$。

当 $f(x):F_2^n\to F_2$ 为 m 阶相关免疫函数时,如果 $r\geqslant n-m$,则根据定理 3.1.4,式(3.1.1)中仅有 $S_{(f)}(0)$,于是

$$a_{i_1i_2\cdots i_r}=-2^{r-1}S_{(f)}(0)\ (\mathrm{mod}\ 2)$$

又

$$S_{(f)}(0)=2^{-n}(2^n-2W_{\mathrm{H}}(f))$$

所以当 $r \geqslant n-m$ 时，

$$a_{i_1 i_2 \cdots i_r} = -2^{r-1} \times 2^{-n}(2^n - 2W_H)(f))\ (\mathrm{mod}\ 2) = 2^{r-n+m} k_0 (\mathrm{mod}\ 2)$$

这里 $W_H(f)=2^m k_0$（由引理 3.1.1 可知）。所以当 $r>n-m$ 时，$a_{i_1 i_2 \cdots i_r}=0$。当 $r=n-m$ 时，若 k_0 为奇数，则所有的 $n-m$ 次项都出现；若 k_0 为偶数，则所有的 $n-m$ 次项都不出现。

当 $W_H(f)=2^{n-1}$，$m \leqslant n-2$ 时，可知 k_0 为偶数，于是对于 $r \geqslant n-m$ 都有 $a_{i_1 i_2 \cdots i_r}=0$。

综上所述，如果 $f(x): F_2^n \to F_2$ 是非线性次数为 k 的 m 阶相关免疫函数，则 $k+m \leqslant n$。特别地，当 f 是平衡布尔函数且 $m \leqslant n-2$ 时，则 $k+m \leqslant n-1$。这表明 f 的非线性次数 $k=\partial^0 f$ 和其相关免疫阶数 m 之间存在着某种制约关系，因此，在具体构造相关免疫函数时必须适当折中考虑。目前，消除这种制约关系的办法主要有两种：一种是引进记忆；另一种是使用广义相关免疫函数。

3.2 快速相关分析方法

在非线性组合生成器中，当组合函数的输出与某些输入变量的符合率 p 达到 0.75 时，从计算量的角度来说，利用分别征服相关分析方法可破译每个 LFSR 的长度 k 不超过 50 的非线性组合生成器。本节主要介绍两个参数适用范围更广的关于非线性组合生成器的相关分析方法，即文献[3]中所称的算法 A 和算法 B。

假定非线性组合生成器的输出序列为 $\underline{z}=\{z_n\}$，$\underline{z}$ 与其中一个 LFSR 序列 $\underline{a}=\{a_n\}$ 的相关概率 $p=P(z_n=a_n)>0.5$，则算法 A 和算法 B 的目的都是用来确定 $\underline{a}$ 的初态。这两个算法都要求反馈的抽头数 t 较小。特别地，当 $p \leqslant 0.75$ 时，要求 $t<10$。算法 A 是一个有效的指数时间分析方法，其计算复杂度为 $O(2^{ck})$，其中 k 表示 LFSR 的长度，$c(<1)$ 依赖于攻击的输入参数。算法 B 是一个多项式时间分析方法，其计算复杂度是 LFSR 的长度 k 的多项式。这两个算法实质上比穷举搜索整个初始状态更快，而且适用于相当长的 LFSR（如 $k=1000$ 或更长）。然而，通过比较可知，当 $c \ll 1$ 且 p 在 0.75 左右时，算法 A 更好；而当 p 在 0.5 左右时，算法 B 更有效。这两个算法可应用于已知明文攻击和唯密文攻击。已证明，当 $p \leqslant 0.75$ 时，如果较长的 LFSR 的抽头数较大（大约 $k \geqslant 100$，$t \geqslant 10$），则这两个算法都是不可行的。

3.2.1 快速相关分析的统计模型

假定一个二元密钥流生成器的输出序列 $\underline{z}$ 与一个 LFSR 序列 $\underline{a}$ 的相关概率 $p=P(z_n=a_n)>0.5$。LFSR 序列 $\underline{a}$ 可通过如下形式的线性递归关系式给出：

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} \tag{3.2.1}$$

其中 $c(x)=c_0+c_1 x+c_2 x^2+\cdots+c_k x^k$（$c_0=1$）是这个关系式的反馈多项式。反馈多项式的抽头数 t 等于 $\{c_1, c_2, \cdots, c_k\}$ 的非零项的个数。因此，式(3.2.1) 可表示成如下含 $t+1$ 项的等式：

$$\sum_{\{i;\,0 \leqslant i \leqslant k,\, c_i \neq 0\}} a_{n-1} = 0 \tag{3.2.2}$$

通过移位序列 $\underline{a}$，可以观测到，每一个固定的数字 a_n 在式(3.2.2) 的 $t+1$ 个位置都出现，也就是说它同时满足形式为式(3.2.2) 的 $t+1$ 个等式。

另外，$c(x)$的每一个多项式倍式都定义了 $\underline{a}$ 的一个线性递归关系式，特别地，对 $j=2^i$，$c(x)^j$ 就是 $\underline{a}$ 的一个线性递归关系式，此时 $c(x)^j=c(x^j)$。这样就比单纯通过移位能获得更多的线性关系式，而且这些关系式的抽头数都是 t。这一特性很重要，因为算法 A 和算法 B 的可行性依赖于抽头数。事实上，对于给定的序列 $\underline{z}$，快速相关分析需要测试所有这些线性关系式来确定对于给定的 n 是否 z_n 与 a_n 一致。

假定 a_n 是固定的，那么按上述方式获得的线性关系式可写成如下形式：

$$\begin{cases} L_1=a_n+b_1 \\ L_2=a_n+b_2 \\ \quad\vdots \\ L_m=a_n+b_m \end{cases} \tag{3.2.3}$$

这里 $b_i(i=1,2,\cdots,m)$恰好是序列 $\underline{a}$ 的 t 个不同项的和，m 是获得的线性关系式的个数，其值在后面确定。

在式(3.2.3)中，对同一下标位置，用序列 $\underline{z}$ 来代替序列 $\underline{a}$，可得到如下表达式：

$$L_i=z+y_i,\quad i=1,2,\cdots,m \tag{3.2.4}$$

这里 L_i 未必为 0。

通过以上分析和相关事实，可以引入一个一般的统计模型。用二元随机变量集$\{a,b_{11},b_{12},\cdots,b_{1t},b_{21},b_{22},\cdots,b_{2t},\cdots,b_{m1},b_{m2},\cdots,b_{mt}\}$代替式(3.2.3) 中序列 $\underline{a}$ 的数字，并满足如下相应的等式：

$$\begin{cases} a+b_{11}+b_{12}+\cdots+b_{1t}=0 \\ a+b_{21}+b_{22}+\cdots+b_{2t}=0 \\ \quad\vdots \\ a+b_{m1}+b_{m2}+\cdots+b_{mt}=0 \end{cases} \tag{3.2.5}$$

类似地，用二元随机变量集$\{z,y_{11},y_{12},\cdots,y_{1t},y_{21},y_{22},\cdots,y_{2t},\cdots,y_{m1},y_{m2},\cdots,y_{mt}\}$表示式(3.2.4)中序列 $\underline{z}$ 的数字。

两个随机变量集有如下关系：

$$P(z=a)=p,\quad P(y_{ij}=b_{ij})=p \tag{3.2.6}$$

除了式(3.2.5)和式(3.2.6)外，这里假定这些二元随机变量都是相互独立且同分布的，它们是 1 或 0 的概率等于 0.5。

对 $i=1,2,\cdots,m$，可导出如下随机变量：

$$\begin{cases} b_i=b_{i1}+b_{i2}+\cdots+b_{it} \\ y_i=y_{i1}+y_{i2}+\cdots+y_{it} \\ L_i=z+y_i \end{cases} \tag{3.2.7}$$

设 b_i 和 y_i 相等的概率是 s，即

$$s=P(y_i=b_i) \tag{3.2.8}$$

显然，s 独立于 i 且是 p 和 t 的函数，即 $s=s(p,t)$。

s 可通过如下递归关系式来计算：

$$\begin{cases} s(p,t)=ps(p,t-1)+(1-p)(1-s(p,t-1)) \\ s(p,1)=p \end{cases} \tag{3.2.9}$$

接下来，考虑随机变量 $L_1, L_2, \cdots, L_m$。

由于 $L_i=0$ 暗含着 $z=a, y_i=b$ 或 $z\neq a, y_i\neq b_i$，因此，对给定的 $h(0\leqslant h\leqslant m)$ 个下标集合 $\{i_1, i_2, \cdots, i_h\}$，恰好在这 h 个对应位置的随机变量等于 0(也称满足关系式或关系式成立)、其他对应位置的随机变量等于 1 的概率为

$$\begin{aligned} &P(L_1=1,\cdots,L_{i_1}=0,\cdots,L_{i_2}=0,\cdots,L_{i_h}=0,\cdots,L_m=1)\\ &=ps^h(1-s)^{m-h}+(1-p)(1-s)^h s^{m-h} \end{aligned} \tag{3.2.10}$$

不失一般性，假定 $L_1=0, L_2=0, \cdots, L_h=0, L_{h+1}=1, L_{h+2}=1, \cdots, L_m=1$，则由贝叶斯公式可知，下列结论成立：

$$\begin{aligned} &P(z=a \mid L_1=L_2=\cdots=L_h=0, L_{h+1}=L_{h+2}=\cdots=L_m=1)\\ &=\frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h}+(1-p)(1-s)^h s^{m-h}} \end{aligned} \tag{3.2.11}$$

$$\begin{aligned} &P(z\neq a \mid L_1=L_2=\cdots=L_h=0, L_{h+1}=L_{h+2}=\cdots=L_m=1)\\ &=\frac{(1-p)(1-s)_s^{h\ m-h}}{ps^h(1-s)^{m-h}+(1-p)(1-s)^h s^{m-h}} \end{aligned} \tag{3.2.12}$$

实际上，式(3.2.11)给出了当 m 个关系式中的 h 个关系式成立时，$z_n=a_n$ 的概率，将这个概率记为 p^*。

根据上面介绍的统计模型以及一些事实，考虑一个随机实验。可访问 z 和 y_{ij} 的输出，因此，可得到 $L_i=z+y_i$，而不能访问 a 和 b_{ij} 的输出，这是因为在我们的应用中 z 和 y_{ij} 对应给定序列的某些数字，而 a 和 b_{ij} 是指未知的 LFSR 序列。特别地，当 z 对应固定数字 z_n 时，我们希望确定 a 对应的固定数字 a_n。从一个先验概率 $p=P(z=a)>0.5$ 开始，记 h 是使得 $L_i=0$ 的下标 i 的个数。然后根据式(3.2.11)把这个先验概率 $p=P(z=a)$ 更新为新的概率 p^*。直观上，我们期望 p^* 在 $z=a$ 的情况下增加，而在 $z\neq a$ 的情况下降低。为了证实这个观点，对这两种情况分别计算 p^* 的期望值。

情况 1：$z=a$。

$$\begin{aligned} E_0[p^*]&=E[p^* \mid z=a]\\ &=\sum_{h=0}^{m}\mathrm{C}_m^h\frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h}+(1-p)(1-s)^h s^{m-h}}s^h(1-s)^{m-h} \end{aligned} \tag{3.2.13}$$

情况 2：$z\neq a$。

$$\begin{aligned} E_1[p^*]&=E[p^* \mid z\neq a]\\ &=\sum_{h=0}^{m}\mathrm{C}_m^h\frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h}+(1-p)(1-s)^h s^{m-h}}s^{m-h}(1-s)^h \end{aligned} \tag{3.2.14}$$

值得一提的是，由式(3.2.13)和式(3.2.14)可知：

$$E[p^*]=pE_0[p^*]+(1-p)E_1[p^*]=p$$

这暗含着，尽管我们期望这个新的概率 p^* 在 $z=a$ 的情况下增加，而在 $z\neq a$ 的情况下降低，但总的期望值是不变的。另外，这里计算的是 p^* 的均值，因此，式(3.2.14)是正确的。

例 3.2.1 设先验概率 $p=P(z=a)=0.75, t=2, m=20$，则可得到 $E_0[p^*]=0.9$，$E_1[p^*]=0.3$。

事实上，新的概率 p^* 是 h 的一个函数，并且可使得在两种情况下的概率分布有明显的区别，这将给我们提供了确定 $z=a$ 或 $z\neq a$ 的一个主要准则。

上述统计模型可以推广到非线性关系式的情况，从而可以将非线性关系式扩展到下面介绍的分析方法中。关键点不是线性而是只有一些数字包括在这些关系式中这一事实。线性本质的优点是产生的许多关系式(通过移位或迭代平方)的概率对同一数字成立。

本节最后介绍一个常用的、比较典型的统计模型。大多数基于 LFSR 的序列密码的分析往往涉及解决下面这样一个问题：假设攻击者收到了序列 $z=a\oplus x$ 的一个适当长的截取段，其中：

(1) a 是一个 m 序列，其反馈本原多项式 $f(x)$是已知的。

(2) 序列 x 的代数结构不明，但已知数字 0 在这个序列中占某种优势(当数字 1 在这个序列中占某种优势时，令 $z'_n=1\oplus z_n$，$x'_n=1\oplus x_n$，则 $z'=a\oplus x'$，在 x' 中 0 占某种优势)，即有 $P(x_n=0)=0.5+\varepsilon$，$P(x_n=1)=0.5-\varepsilon$，$\varepsilon>0$。称 ε 为序列 a 的数字在序列 z 中所占的优势，称 $0.5-\varepsilon$ 为 a 在 z 中的失真率。

现在要做的事情是：设法根据上述两点知识还原序列 a，主要是确定其初态。

因此，如果一个二元密钥流生成器的输出序列 z 与一个 LFSR 序列 a 的相关概率 $p=P(z_n=a_n)>0.5$，则可将这种情况一般化为图 3.2.1 的统计模型。

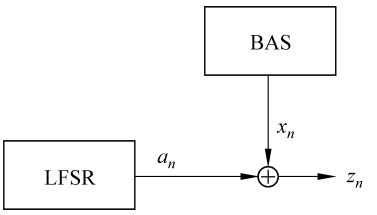

图 3.2.1　一个常用的统计模型

其中 $z_n=a_n\oplus x_n$，BAS 表示二元非对称信源(Binary Asymmetric Source)，$P(x_n=0)=P(z_n=a_n)=p$。a 是一个二元随机序列且 $P(a_n=0)=P(a_n=1)=0.5$。这样，将要介绍的快速相关分析方法实际上是对这种模型的一种分析方法。此外，这种模型也可以用其他分析方法进行分析，如线性校验子分析方法(见 4.2 节)，可参阅文献[16-18]。

3.2.2　算法 A 的基本思想及其描述

假定已给定了序列 $\underline{z}$ 的长度为 N 的一个截取段，LFSR 的反馈多项式、长度 k 和抽头数 t，以及 LFSR 的输出序列 $\underline{a}$ 与给定序列 $\underline{z}$ 的相关概率 $p=P(z_n=a_n)$。现在要解决的问题是：找到未知的 LFSR 序列 $\underline{a}$。基本上，这个序列可通过求解由它的任何 k 个数字构建的关于初始状态的线性方程组被恢复出来。如果这些方程是线性依赖的，可以选择一些附加的数字获得一个线性独立的方程组。因此，为了得到序列 $\underline{a}$ 的一个估计，我们实际上以最高的概率 p^* 选择 $\underline{z}$ 的 k 个数字，这等价于选择满足式(3.2.3)的最多关系式的 k 个数字。

算法 A 的基本思想是：通过测试找出正确的数字，即 $z=a$ 的数字 z。具体测试办法是选择满足更多等式的数字。用这种办法可获得序列 $\underline{a}$ 的相应位置的一个估计。在一定的条件下，这些数字是正确的概率很高，亦即只要对这些数字稍作修改即可。实际上，我们是利用 LFSR 序列 $\underline{a}$ 的线性关系式找出正确的数字，即使得 $z=a$ 的数字。线性关系式可由反馈多项式来描述。通过对反馈多项式进行迭代平方，对每个数字 a 可获得一组线性关系式，每个线性关系式涉及 $\underline{a}$ 的 t 个其他数字。用这种办法获得的关系式的平均数 m 可由后面要介绍的式(3.2.17)计算。

一个固定的数字 z 至少满足 m 个关系式中的 h 个关系式的概率可通过下式来计算：

$$Q(p,m,h)=\sum_{i=h}^{m}C_m^i(ps^i(1-s)^{m-i}+(1-p)(1-s)^i s^{m-i}) \tag{3.2.15}$$

式(3.2.15)可由式(3.2.10)推出。设 $R(p,m,h)$ 表示 $z=a$ 且 m 个关系式中至少有 h 个关系式成立的概率,则有:

$$R(p,m,h)=\sum_{i=h}^{m}C_m^i p s^i (1-s)^{m-i} \tag{3.2.16}$$

这样,在给定的 m 个关系式中至少有 h 个关系式成立的条件下,$z=a$ 的概率为

$$T(p,m,h)=\frac{R(p,m,h)}{Q(p,m,h)}$$

因此,有 $Q(p,m,h)\cdot N$ 个数字满足至少 h 个关系式且正确的概率是 $T(p,m,h)$。对固定的 p、m,$T(p,m,h)$ 是 h 的递增函数。这样,为了最大可能地找到充分多的(至少 k 个)数字,需要确定使得 $Q(p,m,h)\cdot N\geqslant k$ 的最大值 h。

选择 z 中至少满足 h 个关系式的数字,并使用这些数字作为 $\underline{a}$ 在相应下标位置的参考猜测 I_0,则 $(1-T(p,m,h))\cdot Q(p,m,h)\cdot N$ 是 I_0 中被期望的错误数字数。如果这个数很小,则对 I_0 稍作修改即可找到 $\underline{a}$。测试修改 I_0 时利用了 LFSR 序列 $\underline{a}$ 相应的段(phase)和给定序列 $\underline{z}$ 的相关性。如果其相关性超过了一个适当的门限值,则接受这个状态。

下面来估计可获得的关系式的平均数 m,它是 N、k 和 t 的函数。$i(i\geqslant 0)$次迭代平方操作获得的线性关系式(3.2.3) 的长度为 $2^i k$,可建立 $N-2^i k$ 个线性关系式。但必须有 $N-2^i k\geqslant 0$,因此 $i\leqslant \log_2(N/k)$。因为 i 是整数,所以 i 不能大于 $\log_2(N/k)$的整数部分。用$[\log_2(N/k)]$表示 $\log_2(N/k)$的整数部分。因此,可按下列办法估计线性关系式的总量:

$$\begin{aligned}T&=\sum_{i=0}^{[\log_2(N/k)]}(N-2^i k)=N([\log_2(N/k)]+1)-\sum_{i=0}^{[\log_2(N/k)]}2^i k\\&=N([\log_2(N/k)]+1)-(2^{[\log_2(N/k)]+1}-1)k\\&\approx N([\log_2(N/k)]+1)-(2N/k-1)k\\&=N([\log_2(N/k)]-1)+k\end{aligned}$$

因为每一个关系式需要 $\underline{z}$ 的 $t+1$ 个数字,因此,每个数字的关系式的平均数 m 是

$$T\cdot\frac{t+1}{N}=([\log_2(N/k)]-1)(t+1)+\frac{k}{N}(t+1)$$

在我们的应用中$\frac{k}{N}(t+1)\ll 1$,因此,上式可简化为

$$m=m(N,k,t)\approx\left(\log_2\frac{N}{2k}\right)(t+1) \tag{3.2.17}$$

算法 3.2.1 算法 A

第 1 步:根据式(3.2.17)确定 m。

第 2 步:寻找使得 $Q(p,m,h)\cdot N\geqslant k$ 的最大值 h。

第 3 步:对 $\underline{z}$ 中至少满足 h 个关系式的数字进行搜索,并使用这些数字作为 $\underline{a}$ 在相应下标位置的一个参考猜测 I_0。

第 4 步:利用相应的 LFSR 序列 $\underline{a}$ 与序列 $\underline{z}$ 的相关性,通过测试修改 I_0,找到正确的猜测。

值得注意的是,在第 1 步确定的 m 仅仅是一个平均值。一般地,在 $\underline{z}$ 的给定部分中,靠近中间的数字比靠近边界的数字满足更多的关系式。因此,在中间部分,在正确和不正确的数字之间有明显的区别是可能的。这就导致了算法 3.2.1 的一个改进,用下面的第 3′步替

换第 3 步。

第 3′步：根据式(3.2.11)对 $\underline{z}$ 的给定的数字计算新的概率 p^*，并选择 k 个具有最高概率 p^* 的数字。

在第 3 步，I_0 中错误数字的平均数 $\bar{r}=(1-T(p,m,h))\cdot k$。在合适的条件下(如 $\bar{r}\ll 1$)，第 4 步是不必要的。

例 3.2.2 假定 $\underline{z}$ 的截断长度 $N=5000$，$p=0.75$，$k=100$，$t=2$，则可由式(3.2.17)得到测试 $\underline{z}$ 的数字的关系式个数 $m=12$。通过计算函数 $Q(p,m,h)$ 和 $T(p,m,h)$ 可知：要使 $Q(p,m,h)\cdot N=0.02189\times 5000\approx 109$ 成立，期望有 $h\geqslant 11$ 个关系式。此时，$(1-T(p,m,h))\times 109=0.001855\times 109\approx 0.2<1$，这说明在这些数字中期望不正确的数字个数小于 1，这样在第 3 步选择的数字是正确的概率很高。第 4 步是不必要的。

下面讨论算法 3.2.1 的计算复杂度。因为第 1 步至第 3 步的计算时间是可忽略的，所以仅仅估计在第 4 步需要尝试的平均数。假定在第 3 步找到的数字中恰好有 r 个是不正确的，那么在第 4 步需要尝试的最大次数为

$$A(k,r)=\sum_{i=0}^{r}C_k^i$$

对这个公式，存在一个使用二元熵函数的著名估计。二元熵函数的定义如下：

$$H(x)=\begin{cases}0, & x=0,1\\ -x\log_2 x-(1-x)\log_2(1-x), & 0<x<1\end{cases}$$

引理 3.2.1[19]

$$A(k,r)=\sum_{i=0}^{r}C_k^i\leqslant 2^{H(\theta)k} \tag{3.2.18}$$

其中 $\theta=r/k$。

在本书的应用中，只有平均数 $\bar{r}=(1-T(p,m,h))\cdot k$ 对 r 是可达的。对于大的 k，r 大于 $\bar{r}$ 的概率被限定在大约 1/2 内。因此，用 $\bar{r}$ 代替式(3.2.18)中的 r 可获得第 4 步中尝试次数的一个估计。这样，算法 3.2.1 的计算复杂度是 $O(2^{ck})$，$0\leqslant c=H(\bar{r}/k)\leqslant 1$。$c=1$ 的情况对应穷举搜索 LFSR 的所有状态。然而，在合理的条件下 $c\ll 1$，意味着这个攻击要比穷举搜索攻击快。

很显然，c 是 p、t、N 和 k 的一个函数。但事实上，c 仅仅是 p、t 和 N/k 的一个函数，这一点可从算法 3.2.1 的第 1 步和第 2 步直观地观察到。在高安全性要求的应用中，不得不考虑较大的 $d=N/k$，甚至大到 10^6 或更大都是可能的，也是合理的。因此，对不同的但固定的 $d=N/k$，研究 c 作为 p 和 t 的函数的变化规律是一件很有意义的事情，如 $d=N/k=10^2$ 或 $d=N/k=10^6$。

对一个与 LFSR 序列 $\underline{a}$ 的相关性为 p、长度为 N 的序列 $\underline{z}$，$\underline{a}$ 和 $\underline{z}$ 之间的汉明距离的期望值为 $(1-p)\cdot N$。如果 $d=N/k$ 很小，则也许有 $\underline{a}$ 的不同的状态具有距离小于或等于 $(1-p)\cdot N$，也就是说，对相关问题有多个解。在这种情况下，算法 A 也许选择了 $\underline{a}$ 的一个错误状态。

随着抽头数 t 的增加，$c=c(p,t,N/k)$ 收敛于 $H(p)$，这是由于当 t 趋于无穷时，由式(3.2.9)可知，函数 $s(p,t)$ 接近 1/2。再者，由式(3.2.15)和式(3.2.16)可知，当 $s=1/2$ 时，

$$T(p,m,h)=\frac{R(p,m,h)}{Q(p,m,h)}=p$$

这意味着 $\theta=r/k$ 收敛于 $1-p$。因此，$c(p,\infty,N/k)=H(1-p)=H(p)$。这个极限 $c=H(p)$对相关分析的密码学意义是：如果在所有的状态上进行穷举搜索被修改成从最大可能的错误模式开始搜索(见算法 3.2.1 的第 4 步)，则它的计算复杂度是 $O(2^{ck})$而不是 $O(2^k)$。当 $p=0.75$ 时，$c=0.81$。

通过计算可以发现以下一些事实，这里值得注意的是本章思考题 6 是这些事实的基础。当 $t=2$，$p\geqslant 0.6$ 时，算法 3.2.1 比穷举搜索有很大的改进，使用该算法甚至可分析长度为 1000 或更长的 LFSR。当 $d=N/k=10^6$，$t=2$，$p>0.67$ 时，所有的 c 都小于 0.0005。当 $t<10$ 时，随着 $d=N/k$ 的增加，算法 3.2.1 有一个实质性的改进。例如，当 $d=N/k=10^9$，$p>0.57$，$t=2$ 时，$c=0.408$，$H(0.57)=0.986$。当 $t\geqslant 10$，$p\leqslant 0.75$ 时，c 十分接近渐近值 $H(p)$，算法 3.2.1 与(修改的)穷举搜索攻击相比没有本质上的优势，这一事实对可能发生在实际应用中的所有 $d=N/k$ 都成立。

3.2.3 算法 B 的基本思想及其描述

提出算法 B 的动因是如下这样一个事实：如果一个数字仅满足较少的关系式，则条件概率 p^* 是很小的。这就导致了修正(也称校正)满足不超过一定数量关系式的数字的方法。在合适的条件下，可以期望“正确的”的序列是与 LFSR 序列 $\underline{a}$ 有较少不同数字的序列，重复这个过程直到恢复 LFSR 序列 $\underline{a}$。

算法 B 的基本思想是：考虑所有的数字以及它们是正确的数字的概率。开始时我们已经知道 $\underline{z}$ 与 $\underline{a}$ 对应的数字相等的概率是 p，通过考察等式成立的个数，给 $\underline{z}$ 的每个数字赋予一个新的概率 p^*，即 $z_n=a_n$ 的概率。实质上，p^* 是 p 和等式的个数的函数。可以将新的可变的概率 p^* 作为每一轮的输入，迭代地进行上述过程。经过若干轮后，$\underline{z}$ 的所有具有比某一门限值低的概率 p^* 的数字都被修正了。在适当的条件下，我们期望不正确的数字的个数能降低。在这种情况下，重做整个过程若干次后，用新的序列代替 $\underline{z}$，直到找到 LFSR 序列 $\underline{a}$ 为止。

m 个关系式中至多有 h 个关系式成立的概率可按如下公式计算：

$$U(p,m,h)=\sum_{i=0}^{h}C_m^i(ps^i(1-s)^{m-i}+(1-p)(1-s)^is^{m-i}) \tag{3.2.19}$$

再者，$z_n=a_n$ 且 m 个关系式中至多有 h 个关系式成立的概率可由如下公式给出：

$$V(p,m,h)=\sum_{i=0}^{h}C_m^ips^i(1-s)^{m-i} \tag{3.2.20}$$

类似地，$z_n\neq a_n$ 且 m 个关系式中至多有 h 个关系式成立的概率为

$$W(p,m,h)=\sum_{i=0}^{h}C_m^i(1-p)(1-s)^is^{m-i} \tag{3.2.21}$$

因此，$U(p,m,h)\cdot N$ 是满足至多 h 个关系式的 $\underline{z}$ 中数字的期望数。如果这些数字被修正，则 $W(p,m,h)\cdot N$ 是被正确地改变的数字的个数，$V(p,m,h)\cdot N$ 是被错误地改变的数字的个数。正确数字的增量是 $W(p,m,h)\cdot N-V(p,m,h)\cdot N$。定义相对增量如下：

$$I(p,m,h)=W(p,m,h)-V(p,m,h) \tag{3.2.22}$$

这样，对给定的 p 和 m，最佳方式是选择使得 $I(p,m,h)$达到最大值的 $h_{\max}$作为 h。

为了达到最大的修正效果(correction effect，也称校正作用)，取门限值 P_{thr}为

$$P_{\text{thr}}=\frac{1}{2}(p^*(p,m,h_{\max})+p^*(p,m,h_{\max}+1)) \tag{3.2.23}$$

因此，概率为 $p^*<P_{\text{thr}}$ 的数字的个数 N_w 的期望数为

$$N_{\text{thr}}=U(p,m,h_{\max})\cdot N \tag{3.2.24}$$

如果只有少数数字的概率 $p^*<P_{\text{thr}}$，那么新的概率分配需要被迭代执行。

算法 B 的整个攻击过程交替执行如下两个阶段：

(1) 计算阶段。对 $\underline{z}$ 的每个数字分配新的概率 p^*。

(2) 校正阶段。修正概率为 $p^*<P_{\text{thr}}$ 的数字，并将每个数字的概率重置为原来的值 p。

对计算阶段可进行迭代。为此，需要将式(3.2.9)一般化。t 个数字可能有不同的概率，分别记为 $p_1,p_2,\cdots,p_t$，则

$$\begin{cases}s(p_1,p_2,\cdots,p_t,t)=p_t s(p_1,p_2,\cdots,p_{t-1},t-1)\\ \qquad\qquad +(1-p_t)(1-s(p_1,p_2,\cdots,p_{t-1},t-1))\\ s(p_1,1)=p_1\end{cases} \tag{3.2.25}$$

这个一般化也引发了其他所有公式的一般化，特别是关于 p^* 的式(3.2.11)的一般化。

大量的实验表明，在这种迭代中，假定迭代数 α 是一个很有限的数是合理的。在很多情况下，$\alpha=5$ 是一个合适的选择。

算法 3.2.2 算法 B

第 1 步：根据式(3.2.17)确定 m。

第 2 步：找到使得 $I(p,m,h)$ 达到最大值的 $h=h_{\max}$。如果 $I_{\max}=I(p,m,h_{\max})\leqslant 0$，则在计算阶段没有修正效果，即攻击失败；如果 $I_{\max}>0$，则根据式(3.2.23)和式(3.2.24)计算 P_{thr} 和 N_{thr}。

第 3 步：初始化迭代计数器 $i=0$。

第 4 步：对 $\underline{z}$ 的每个数字，利用其所满足的关系式的个数计算新的概率 p^*(计算阶段，利用一般化了的式(3.2.11)和式(3.2.25))。确定概率为 $p^*<P_{\text{thr}}$ 的数字的个数 N_w。

第 5 步：如果 $N_w<N_{\text{thr}}$ 或 $i<\alpha$(其中 α 表示事先限定的迭代数)，则给 i 加 1 并转向第 4 步。

第 6 步：修正概率为 $p^*<P_{\text{thr}}$ 的 $\underline{z}$ 的数字并将每个数字的概率重置为原来的值 p(校正阶段)。

第 7 步：如果有 $\underline{z}$ 的数字不满足基本的反馈关系式(3.2.1)，则转向第 3 步。

第 8 步：在 $\underline{a}=\underline{z}$ 时终止。

随后的讨论基于应用算法 B 进行的模拟，该算法的外循环(第 3 步至第 7 步)称作轮(round)，内循环(第 4 步和第 5 步)称作迭代(iteration)。在第 1 轮，可观察到理论上期望 $N_w\approx N_{\text{thr}}$，因此，只有第 4 步和第 5 步进行一次迭代也许是必要的。在较高次的轮中，错误($z_n\neq a_n$)不再独立于关系式，这样，统计模型现在不能严格地应用。这通过在较高轮中观察到 $N_w\ll N_{\text{thr}}$ 被反映出来。由于这个原因，迭代分配新的概率直至有足够多的数字具有概率 $p^*<P_{\text{thr}}$。然而在一些迭代之后，可观察到概率 p^* 或者很接近 0 或者很接近 1 的数字表现出强烈的两极分化。除去一些数字，这个两极分化变得稳定，意味着不需要再迭代。这证实了迭代有限的次数 α 之后一轮终止。

在执行算法 B 时，通过例 3.2.3 观察到，具有 $\underline{a}=\underline{z}$ 的一些轮终止后获得了一个稳定的

修正效果。为了解释这种现象，需要考虑不同轮之间的统计独立性。事实上，我们也不能解释除第一次迭代外算法 B 为什么能够成功，也不清楚为什么算法 B 能够在若干轮后导致一个解。

可对算法 B 做一些修改。例如，在第 6 步，根据在每一轮之后错误的期望数降低的事实，可将其概率重置为高于原来的值 p。然而，模拟结果表明，这样做没有导致算法 B 的效果的改进。

为了估计算法 B 的修正效果，不得不对给定的 p、t、N 和 k，计算 $I_{\max}=I(p,m,h_{\max})$（第 2 步）。首先由式(3.2.17)可知，m 是 t 和 $d=N/k$ 的函数，而 $h_{\max}$ 是 p 和 m 的函数，因此，$I_{\max}$ 是 p、t 和 $d=N/k$ 的函数，即 $I_{\max}=I_{\max}(p,t,d)$。在一次迭代中，被修正的数字的期望数可按如下公式计算：

$$N_c=I_{\max}(p,t,d)\cdot N \tag{3.2.26}$$

为方便起见，可将 N_c 表示为 $N_c=F(p,t,d)\cdot k$，其中

$$F(p,t,d)=I_{\max}(p,t,d)\cdot d \tag{3.2.27}$$

$F(p,t,d)$是一个独立于 k 的修正因子。如果 $F(p,t,d)\leqslant 0$，则没有修正效果，攻击失败；如果 $F(p,t,d)\geqslant 0.5$，大多数实验结果表明，算法 B 看起来好像是很成功的。对固定的 t 和 d，计算使得 $F(p^*,t,d)\geqslant 0.5$（$p^*\geqslant p$）最小的相关概率，见表 3.2.1。

表 3.2.1　满足 $F(p,t,d)=0.5$ 的 p

d	t								
	2	4	6	8	10	12	14	16	18
10	0.761	0.880	0.980	0.980	0.980	0.980	0.980	0.980	0.980
10^2	0.959	0.754	0.824	0.863	0.889	0.905	0.917	0.926	0.934
10^3	0.553	0.708	0.787	0.832	0.861	0.882	0.897	0.908	0.918
10^4	0.533	0.679	0.763	0.812	0.844	0.867	0.883	0.896	0.906
10^5	0.525	0.663	0.748	0.800	0.833	0.857	0.875	0.889	0.900
10^6	0.519	0.650	0.737	0.789	0.825	0.849	0.868	0.883	0.894
10^7	0.515	0.641	0.727	0.781	0.817	0.843	0.862	0.877	0.890
10^8	0.514	0.634	0.720	0.774	0.812	0.838	0.858	0.874	0.886
10^9	0.512	0.628	0.714	0.770	0.807	0.833	0.854	0.870	0.882
10^{10}	0.510	0.621	0.709	0.764	0.802	0.830	0.850	0.866	0.879

从表 3.2.1 可以看出，当 $t<8$ 时，一个成功攻击必需的相关概率界是在相关的实际值范围内。特别地，当 $t=8$ 时，概率越来越接近 0.5。在这些情况下，算法 B 对很长的 LFSR 是成功的。例 3.2.3 也说明了这一点。

例 3.2.3　由表 3.2.1 可知，满足 $F(p,4,100)=0.5$ 的 $p=0.754$。现在考虑下列情形：$N=10^4$，$k=100$，$t=4$，$p=0.75$（而不是 0.754），则 $d=100$，$F(p,t,d)=0.392$，而不是 0.5。可计算出算法 B 中的参数 $p_{\text{thr}}=0.524$，$N_{\text{thr}}=448$。这样，在第 1 次迭代中期望有 448 个数字被改变，导致减少 39 个错误数字。