σ-LFSR 在序列密码算法 ABC 中的应用

曾 光 韩文报 范淑琴
(解放军信息工程大学信息工程学院 郑州 450002)

摘 要：σ-LFSR 是一种基于字的，在安全性和效率上达到较好折衷的反馈移位寄存器。该文利用一个 σ-LFSR 替代序列密码算法 ABC 中的 LFSR，使得 ABC 的周期由 $2^{32}(2^{127}-1)$ 变为 $2^{32}(2^{128}-1)$，且其二元域上等价 LFSR 反馈多项式的 Hamming 重量由 3 增加到 65，恰好等于次数的一半。此改进增强了 ABC 抵抗快速相关攻击的能力，同时改进后的软件实现效率与原来相当。

关键词：序列密码；σ-线性反馈移位寄存器；ABC；快速软件加密
中图分类号：TP309.7 **文献标识码**：A **文章编号**：1009-5896(2009)03-0727-04

Application of σ-LFSR in Stream Cipher ABC

Zeng Guang Han Wen-bao Fan Shu-qin
(*Information Engineering Institute*, *Information Engineering University*, *Zhengzhou* 450002, *China*)

Abstract: σ-LFSR is a word-oriented feedback shift register with a good tradeoff between security and efficiency. As an example, using σ-LFSR in ABC increases its period from $2^{32}(2^{127}-1)$ to $2^{32}(2^{128}-1)$ and, more important, its Hamming weight with the feedback polynomial of equivalent LFSR over binary field from 3 to 65, which is just half of the degree 128. Consequently, its resistance to fast correlation attack is consolidated while the guaranteed efficiency in software is almost the same.

Key words: Stream cipher; σ-LFSR; ABC; Fast software encryption

1 引言

近年来，出现了许多适合软件实现的序列密码算法，例如 ABC[1,2]，Turing[3]，SSC2[4]，Sober[5]，Snow[6,7]等。可以发现上述序列密码的设计方式较以往有着明显的不同。最大的不同就是传统的设计是基于比特的，而现代的序列密码的设计是以字(例如 32bit 或 64bit)为基本操作，从而达到软件实现的高效。σ-LFSR 是本文作者提出的一种反馈移位寄存器[8,9]，它以处理器的字长为基本运算单元，利用少数计算机的基本指令即可构造出具有良好密码学性质的最大周期序列，可以作为适合软件实现的序列密码的驱动部件。

ABC[1,2]是 ECRYPT[10]征集到的面向软件设计的序列密码算法。它是所有面向软件实现的序列密码算法中速度最快的一个，在 Intel Pentium IV CPU 上可以达到 3.7 cycles/byte。ABC 共有 3 个版本，结构基本相同。输出序列与移存器序列的相关性是 ABC 的最大弱点，本文利用 σ-LFSR 代替 ABCv3 中的 LFSR，增强了其抵抗快速相关攻击能力，同时在软件实现效率上几乎没有损失。

2 σ-LFSR 介绍

σ-LFSR 是一类面向软件的反馈移位寄存器，它以字结构作为基本运算单元，能够充分利用现代处理器提供的操作，具有结构简单、实现快速的特点。σ-LFSR 是字 LFSR 的推广，SOBER 和 SNOW 中的字 LFSR 都可以看成是 σ-LFSR 的特例。设 $\mathrm{GF}(2^m)$ 表示由 2^m 个元素组成的有限域，令 $\beta_0, \beta_1,\cdots,\beta_{m-1}$ 为 $\mathrm{GF}(2^m)$ 在二元域 $\mathrm{GF}(2)$ 上的一组基且 $\alpha\in\mathrm{GF}(2^m)$，则存在 $\mathrm{GF}(2)$ 上 m 维向量 $(\boldsymbol{a}_0, \boldsymbol{a}_1,\cdots,\boldsymbol{a}_{m-1})$ 使得 $\alpha=\boldsymbol{a}_0\beta_0+\boldsymbol{a}_1\beta_1+\cdots+\boldsymbol{a}_{m-1}\beta_{m-1}$。设 $M_m(\mathbf{F}_2)$ 表示 $\mathrm{GF}(2)$ 上的 $m\times m$ 阶矩阵构成的矩阵环，对于 $M_m(\mathbf{F}_2)$ 中的矩阵 $\boldsymbol{M}$ 可以诱导出一个线性变换 $\mathcal{M}(\alpha)=(\boldsymbol{a}_0, \boldsymbol{a}_1,\cdots,\boldsymbol{a}_{m-1})\cdot\boldsymbol{M}$。为了方便描述，将 $\mathcal{M}(\alpha)$ 简记为 $\alpha\boldsymbol{M}$。

定义 1[8] 设 $f(x)=x^n+\boldsymbol{C}_{n-1}x^{n-1}+\cdots+\boldsymbol{C}_1x+\boldsymbol{C}_0$，其中 $\boldsymbol{C}_0,\cdots,\boldsymbol{C}_{n-1}\in M_m(\mathbf{F}_2)$，对于 $\mathrm{GF}(2^m)$ 上的序列 $\underline{s}=(s_t)_{t\geq 0}$，若其满足关系式 $s_{i+n}=s_i\boldsymbol{C}_0+s_{i+1}\boldsymbol{C}_1+\cdots+s_{i+n-1}\boldsymbol{C}_{n-1}$ 对 $i=0,1,2,\cdots$ 都成立，则称此系统为 $\mathrm{GF}(2^m)$ 上由 $f(x)$ 生成的 n 级 $\sigma-\mathrm{LFSR}$，矩阵多项式 $f(x)$ 称为 σ-多项式。

定义 2[8] 如果 $\mathrm{GF}(2^m)$ 上由 $f(x)$ 定义的 n 级 σ-LFSR 生成的序列达到最大周期 $2^{mn}-1$，则称此为本原 σ-LFSR 且对应的 σ-多项式称为本原 σ-多项式。

定理 1[8] 设 σ-LFSR 以 $f(x)\in M_m(\mathbf{F}_2)[x]$ 为 σ-多项式，那么 σ-LFSR 为本原的充要条件是矩阵多项式的行列式 $|f(x)|$ 为 $\mathrm{GF}(2)$ 上 mn 次本原多项式。

定理 2[8] 设 $f(x)\in M_m(\mathbf{F}_2)[x]$ 是一个 n 次首一本原 σ-多项式，那么其输出序列的每个分量序列都是二元域 $\mathrm{GF}(2)$ 上的 m-序列，且极小多项式为 $|f(x)|$。

2007-10-26 收到，2008-04-04 改回
国家自然科学基金(90704003)，国家 863 计划项目(2006AA01Z425)和国家 973 计划项目(2007CB807902)资助课题

本原 σ-LFSR 可以从理论上保证输出序列达到最大周期、输出序列是平衡的、具有良好的伪随机性、分量序列都是 m-序列、拥有广阔的选择空间等。由于 GF(2^m) 上 n 级 σ-LFSR 可由 GF(2) 上一个 mn 级 LFSR 生成，因此 GF(2) 上与 σ-LFSR 等价的 LFSR 反馈多项式的 Hamming 重量对安全性至关重要，能够达到次数一半是较为理想的选择。另外，σ-LFSR 的设计要涉及到矩阵乘向量的运算，这个操作在现代 CPU 上是较为费时的操作。但如果合适地选取矩阵，则能用简单的操作实现，从而提高效率。本文选取了 4 种特殊矩阵如下：

$$\wedge_\gamma=\begin{pmatrix} c_0 & 0 & \cdots & 0 \\ 0 & c_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{m-1} \end{pmatrix}_{m\times m},\ \boldsymbol{\sigma}=\begin{pmatrix} 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}_{m\times m},$$

$$\boldsymbol{L}=\begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}_{m\times m} \tag{1}$$

(1)与运算 $\wedge_\gamma$：令 $\gamma=\sum_{i=0}^{m-1}c_i\beta_i$，其中对 $i=0,1,\cdots,m-1$ 有 $c_i\in$GF(2)。对上述 $\alpha\in$GF(2^m)，定义与运算为 $\wedge_\gamma(\alpha)=a_0c_0\beta_0+\cdots+a_{m-1}c_{m-1}\beta_{m-1}$。事实上，$\wedge_\gamma$ 恰好是对操作数 γ 和 α 做与运算，其对应的矩阵为式(1)中左侧的矩阵。

(2)循环移位运算 $\boldsymbol{\sigma}$：$\boldsymbol{\sigma}(\alpha)=\sigma(a_0\beta_0+a_1\beta_1+\cdots+a_{m-1}\cdot\beta_{m-1})=a_{m-1}\beta_0+a_0\beta_1+\cdots+a_{m-2}\beta_{m-1}$。事实上，在具体实现时是一个循环右移操作。其对应的矩阵为式(1)中间的矩阵。

(3)左移(右移)运算 $\boldsymbol{L}$ (R)：定义 $\boldsymbol{L}(\alpha)=a_1\beta_0+a_2\beta_1+\cdots+a_{m-1}\beta_{m-2}$。在实现时即为逻辑左移运算，对应的矩阵为式(1)右侧矩阵。同理 $\boldsymbol{L}$ 的矩阵转置为右移运算矩阵 $\boldsymbol{R}$。

(4)左右移混合运算 $\sqcup_{s,t}$：设 s 和 t 都是正整数且满足 $0<s,t<m$，定义 $\sqcup_{s,t}=\boldsymbol{L}^s+\boldsymbol{R}^t$。这个线性运算为左移和右移运算的合成。

3 利用 σ-LFSR 改进 ABC 序列密码算法

3.1 ABC 序列密码算法

ABCv3 由 3 部分组成：A，B 和 C，如图 1 所示。其中 A 是一个逻辑运算类型的 LFSR，B 是一个 T-函数，C 是一个与密钥相关的 S 盒。ABCv1 的结构与 ABCv3 基本相同，不同之处是 ABCv1 采用的是 63bit 的 LFSR 和 B 部分的略微差别。

组件 A 为基于比特的 LFSR，其特征多项式是 GF(2) 上的本原多项式 $g(x)=x^{127}+x^{63}+1$。将 A 中的寄存器记为 $(\overline{z}_3,\overline{z}_2,\overline{z}_1,\overline{z}_0)$，每个 $\overline{z}_i$ 为 32bit 长。组件 A 每一拍进行一次

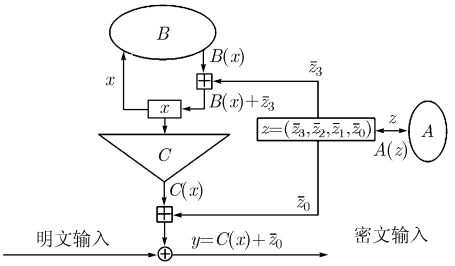

图 1 序列密码 ABC 的算法框架

如下更新：

$$\left.\begin{aligned} &\xi=(\overline{z}_2\oplus(\overline{z}_1\lll 31)\oplus(\overline{z}_0\ggg 1))\mod 2^{32} \\ &\overline{z}_0=\overline{z}_1,\quad \overline{z}_1=\overline{z}_2,\quad \overline{z}_2=\overline{z}_3,\quad \overline{z}_3=\xi \end{aligned}\right\} \tag{2}$$

组件 B,C 以及 ABCv3 的初始化这里不再描述，本文只关心组件 A 中 LFSR 的生成过程。

ABCv1 被 Berbain 和 Gilbert[11]用分割攻击攻破，他们主要利用了 LFSR 的较短周期。为抵御这类攻击，ABC 设计者改进了组件 A 和 B 给出 ABCv2。但基于相同的原因，ABCv2 依然不能抵御快速相关攻击[12]，攻击计算量为 $O(2^{46.34})$。在 ABCv3 中，组件 C 的选择是为了避免 ABCv2 中的弱密钥，但组件 A 未改变，快速相关攻击仍现实可行[13]，攻击计算量为 $O(2^{50.56})$。其组件 A 的少项式和组件 C 的不平衡是 ABC 被攻击的根本原因。σ-LFSR 具有良好密码学性质和快速的软件效率，我们可以将其应用在 ABC 中，在获得高效的同时还能够提供良好的密码学性质。

3.2 改进的 ABC 序列密码算法

为用 σ-LFSR 代替 ABCv3 中的 LFSR，需要寻找指令尽可能少的 GF(2^{32}) 上 4 级本原 σ-LFSR。注意到 ABCv3 中的 LFSR 在进行如式(2)的更新时只用到 4 个机器指令：两个异或，两个移位。所以应首先寻找由 3 个或 4 个指令即可实现的本原 σ-LFSR，但实际中并没有找到这样的本原 σ-LFSR。经过大量实验，虽然目前还不能给出证明，但我们认为不存在 GF(2^{32}) 上只由 3 个或 4 个机器指令构成的本原 σ-LFSR。进而寻找由 5 个或 6 个基本指令构造的 σ-LFSR，从而得到了大量的本原 σ-LFSR，它们的结构有以下 3 种。

在图 2 中，2(b)和 2(c)的 σ-LFSR 包含 5 个运算，图 2(a)包含 6 个运算。为选取可替代 ABCv3 中 LFSR 的本原 σ-LFSR，在表 1，表 2 中列出一些实验结果。

表 1 和表 2 列出了本原 σ-LFSR 和与其等价的二元 LFSR 反馈多项式的重量，其中与系数用十六进制表示。从表中可以看出，图 2(a)的 σ-LFSR 大部分都达到了最大周期 $2^{128}-1$，而且它们反馈多项式的重量接近次数的一半。但

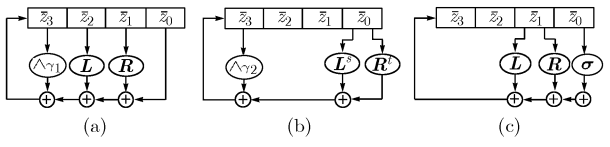

图 2 三类备选的本原 σ-LFSR 结构图

表 1 图 2(a)模型的部分本原 σ-LFSR 的 σ-多项式

(a)模型 σ-多项式	重量	(a)模型 $\sigma-$多项式	重量
$x^4+\wedge$"0x4a"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	71	$x^4+\wedge$"0x54"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	65
$x^4+\wedge$"0x5d"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	59	$x^4+\wedge$"0xad"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	51
$x^4+\wedge$"0x108"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	59	$x^4+\wedge$"0x133"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	57
$x^4+\wedge$"0x1b4"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	71	$x^4+\wedge$"0x1f8"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	63
$x^4+\wedge$"0x2a5"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	65	$x^4+\wedge$"0x14f9"$x^3+\boldsymbol{L}x^2+\boldsymbol{R}x+1$	63

表 2 图 2(b), 2(c)模型的部分本原 σ-LFSR 的 σ-多项式

(b)模型 σ-多项式	重量	(b)模型 σ-多项式	重量	(c)模型 σ-多项式	重量
$x^4+\wedge$"0xfff1c37"$x^3+\sqcup_{7,1}$	23	$x^4+\wedge$"0x6b"$x^3+\sqcup_{1,1}$	39	$x^4+\sqcup_{27,27}\ x^3+\sigma$	9
$x^4+\wedge$"0x3fd"$x^3+\sqcup_{3,1}$	33	$x^4+\wedge$"0x1ff"$x^3+\sqcup_{1,1}$	41	$x^4+\sqcup_{25,25}\ x^3+\sigma$	17
$x^4+\wedge$"0x1c162"$x^3+\sqcup_{9,7}$	13	$x^4+\wedge$"0x2ae"$x^3+\sqcup_{1,1}$	51	$x^4+\sqcup_{13,13}\ x^3+\sigma^5$	43
$x^4+\wedge$"0xfff29d8"$x^3+\sqcup_{5,3}$	31	$x^4+\wedge$"0x4ee"$x^3+\sqcup_{1,1}$	65	$x^4+\sqcup_{15,15}\ x^3+\sigma^{15}$	31
$x^4+\wedge$"0xfff3008"$x^3+\sqcup_{3,1}$	65	$x^4+\wedge$"0xc83"$x^3+\sqcup_{1,1}$	47	$x^4+\sqcup_{25,25}\ x^3+\sigma^{31}$	17

它们在更新时要用到 6 个基本运算，所以不采用此类 σ-LFSR。而图 2(c)的模型虽然只用了 5 个基本运算，但它的反馈多项式重量远远偏离了次数的一半。最后考查图 2(b)的模型，得到如下的 σ-多项式：

$$\theta(x)=x^4+\wedge 0\text{xfff3008}x^3+\sqcup_{3,1} \tag{3}$$

由 $\theta(x)$ 定义的 σ-LFSR 的周期为 $2^{128}-1$，且与其等价的 LFSR 反馈多项式为 GF(2) 上重量为 65 的 128 次本原多项式 $p(x)$=$x^{128}+x^{127}+x^{126}+x^{125}+x^{124}+x^{123}+x^{122}+x^{121}+x^{120}+x^{119}+x^{118}+x^{117}+x^{116}+x^{115}+x^{114}+x^{113}+x^{112}+x^{111}+x^{110}+x^{109}+x^{104}+x^{103}+x^{102}+x^{101}+x^{96}+x^{95}+x^{94}+x^{92}+x^{90}+x^{89}+x^{85}+x^{83}+x^{80}+x^{78}+x^{76}+x^{74}+x^{73}+x^{71}+x^{69}+x^{68}+x^{66}+x^{65}+x^{64}+x^{62}+x^{59}+x^{58}+x^{56}+x^{54}+x^{53}+x^{52}+x^{51}+x^{50}+x^{46}+x^{43}+x^{40}+x^{36}+x^{32}+x^{30}+x^{29}+x^{28}+x^{27}+x^{26}+x^{15}+x^{12}+1$。将此 σ-LFSR 放到 ABC 中，组件 A 的更新过程变为

$$\left.\begin{array}{l}\zeta=((\overline{z}_3\wedge''0\text{xfff3008}'')\oplus(\overline{z}_0\ll 3)\oplus(\overline{z}_0\gg 1))\bmod 2^{32}\\ \overline{z}_0=\overline{z}_1,\quad \overline{z}_1=\overline{z}_2,\quad \overline{z}_2=\overline{z}_3,\quad \overline{z}_3=\zeta\end{array}\right\} \tag{4}$$

由于此 σ-LFSR 与 ABCv3 中原始的 LFSR 性质相似，所以改进后的 ABC 同原来的 ABCv3 具有相同的分布和线性复杂度。从 ABCv3 中的 LFSR 变为由式(3)定义的本原 σ-LFSR，周期可以从 $2^{32}(2^{127}-1)$ 增大到 $2^{32}(2^{128}-1)$。

利用 eSTREAM 的测试平台[14,15]对 ABCv3 和改进后的 ABC 进行比较。测试在主频为 2.66GHz 的 Intel Pentium 630 CPU，内存 2G，操作系统为 Linux AS4 的环境下完成。测试平台 estreambench 允许测试者在 10 种模式下评测算法速度。本文对这 10 种模式都进行了测试，具体结果见表 3。

表 3 中的比例是指改进后的 ABC 相比 ABCv3 速度慢的百分比。10 种比例的均值为 1.183%，可见改进的 ABC 与原来的 ABCv3 有几乎相同的效率。

3.3 改进的 ABC 算法抵御相关攻击的能力

下面将说明改进后的 ABC 可提高对快速相关攻击的抵御能力。

在对 ABC 实施快速相关攻击时[12,13]，都是通过对 $g(x)$ 连续进行 5 次平方找到 32bit 字的线性关系如下：

$$g^{2^5}(x)=x^{127\times 32}+x^{63\times 32}+1 \tag{5}$$

其中 $g(x)$ 是 ABCv3 中组件 A 的特征多项式。但 σ-LFSR 对应的 σ-多项式 $\theta(x)$ 的系数属于矩阵环 $M_m(F_2)$，它是一个非交换环。特别有 $\wedge\upsilon\cdot\sqcup_{3,1}\neq\sqcup_{3,1}\cdot\wedge\upsilon$，其中 $\upsilon=0\text{xfff3008}$，所以无法利用连续方幂的方法得到等式

$$\theta^{2^k}(x)=x^{4\times 2^k}+\wedge 0\text{xfff3008}x^{3\times 2^k}+\sqcup_{3,1}^{2^k} \tag{6}$$

建立 σ-LFSR 的 32bit 字的线性关系。除此之外，由于 $\theta(x)$ 是一个矩阵多项式，如何寻找 32bit 序列的线性表达式，目前无一般方法。

要得到由式(3)定义的 σ-LFSR 的 32bit 字的线性关系，只能利用其在 GF(2) 上由 $p(x)$ 定义的 LFSR 的线性递归关系。同时由于组件 C 未改变，仍可以利用文献[12,13]的方式建立 σ-LFSR 的输出序列和密钥流序列的线性关系式，下面来讨论此种情形下快速相关攻击的复杂度。采用的快速相关攻击方法是 Meier 和 Staffelbach[16]提出的算法 A。令 $(\overline{z}_3,\overline{z}_2,\overline{z}_1,\overline{z}_0)$ 表示式(3)定义的 σ-LFSR 的状态，$\overline{z}_l^i$ 表示

表 3 改进的 ABC 和原来 ABC 的性能对比(cycle/byte)

模式	1	2	3	4	5	6	7	8	9	10
改进 ABC	4.41	3.65	3.82	3.78	5.89	6.13	5.97	10.56	10.53	149.29
ABC-v3	4.3	3.51	3.70	3.70	5.83	6.13	5.97	10.88	10.39	148.60
比例 (%)	2.56	3.99	3.24	2.16	1.02	0	0	-2.94	1.34	0.46

128bit 寄存器第 i 拍第 l 个 32bit 状态，其中 $0 \le l \le 3$，y_i 是第 i 拍输出的 32bit 密钥流。假设已建立关系式

$$\Pr(f_L(\overline{z}_0^i,\overline{z}_1^i,\overline{z}_2^i,\overline{z}_3^i) \oplus f_K(y_i) = 0) = p > 0.5 \quad (7)$$

其中 f_L,f_K 分别是 σ-LFSR 和密钥流的线性关系。例如，在文献[13]中 $f_L = \delta_n(\overline{z}_0^i) \oplus \delta_{n-1}(\overline{z}_0^i)$，$f_K = \delta_n(y_i)$，其中 $\delta_i(x)$ 是第 i 个比特的选择函数，如果在此 f_L,f_K 中取 $n = 16$ 就得到文献[12]建立的线性关系。故可令 $u_i = f_L(\overline{z}_0^i,\overline{z}_1^i,\overline{z}_2^i,\overline{z}_3^i)$，$w_i = f_K(y_i)$，通过对 $p(x)$ 反复平方，可得到关于 u_i 的多个线性关系式。

$$u_i \oplus u_{i+12\cdot 2^j} \oplus u_{i+15\cdot 2^j} \oplus \cdots \oplus u_{i+127\cdot 2^j} \oplus u_{i+128\cdot 2^j} = 0 \quad (8)$$

对 $j \ge 0$ 成立。根据式(7)和式(8)，可得

$$s = \Pr(w_i \oplus w_{i+12\cdot 2^j} \oplus w_{i+15\cdot 2^j} \oplus \cdots \oplus w_{i+127\cdot 2^j} \oplus w_{i+128\cdot 2^j} = 0 \mid u_i = w_i) \quad (9)$$

其中每个 j 均表示关于 w_i 的一个关系式。每个 w_i 平均有 $m = m(N,\ k,\ t) = \log_2(N/2k)(t+1)$ 个关系，这里 N 是输出比特个数，$k = 128$ (LFSR 长度)，$t = 64$ (σ-LFSR 反馈端个数)。

根据文献[16]中的算法 A，w_i 在 m 个关系式中满足至少 h 个的概率为

$$Q(h) = \sum_{i=h}^{m} C_m^i (ps^i(1-s)^{m-i} + (1-p)(1-s)^i s^{m-i}) \quad (10)$$

如果 $u_i = w_i$，则 w_i 在 m 个关系式中满足 h 个的概率为

$$R(h) = \sum_{i=h}^{m} C_m^i p s^i (1-s)^{m-i} \quad (11)$$

在攻击中会有 $NQ(h)$ 个 u_i 被预测，有 $NR(h)$ 个会被正确预测。若错误位数量较少，即 $r = NQ(h) - NR(h)$ 远小于 1，则算法 A 的计算复杂度可忽略不计。但对上述选定的 σ-LFSR，如假设 $N = 6400,\ p = 0.75$，那么每个 w_i 平均有 302 个关系。对于 $h = 168$，在 183.657bit 中有 137.743bit 被正确预测，于是 $r = 45.9144 > 1$。这个假设参数已经对攻击者相当有利，在实际攻击中很难得到这么理想得相关性。经过计算，对于所有参数均有 $r \ge 1$，所以算法 A 的复杂度为 $O(2^{ck})$，其中 $k = 128$，c 是 0 到 1 的常数。由文献[16]可知 c 的渐近值是 0.811，可得算法 A 的计算复杂度为 $O(2^{103})$。这个复杂度将使得快速相关攻击不再现实可行，通过上述改进提高了 ABC 序列密码算法抵抗快速相关攻击的能力。

4 结束语

研究 σ-LFSR 的主要思想就是针对现代 CPU 的特点设计适合其高效实现且具有良好密码学性质的字 LFSR。改进后的 ABC 保持了原有的分布特性、复杂性并将周期从 $2^{32}(2^{127}-1)$ 增大到 $2^{32}(2^{128}-1)$。更为重要的是 LFSR 的重量由 3 增加到 65，从而增加了 ABC 抵抗快速相关的能力。利用 σ-LFSR 改进 ABC 的主要目的是寻找 σ-LFSR 的一个的应用，而不是增强序列密码算法 ABC 的安全性，因为组件 C 也需要变动。从改进的 ABC 算法可以看出，具有良好性质的本原 σ-LFSR 可以作为适合快速软件实现的序列密码算法基本组件。

参 考 文 献

[1] Anashin V, Bogdanov A, and Kizhvatov I, *et al.*. ABC: A new fast flexible stream cipher. eSTREAM, ECRYPT stream cipher project. Report 2005/001, http://www.ecrypt.eu.org/stream, 2005, 04.

[2] Anashin V, Bogdanov A, and Kizhvatov I, *et al.*. ABC: A new fast flexible stream cipher, Specification Version 2. http://crypto.rsuh.ru/papers/abc-spec-v2.pdf, 2005.

[3] Hawkes P and Rose G. Turing: A fast stream cipher [C]. Fast Software Encryption 2003 Workshop, LNCS, 2003, Vol 2887: 290–306.

[4] Zhang M, Carroll C, and Chan A. The software-oriented stream cipher SSC2 [C]. Fast Software Encryption 2000 Workshop, LNCS, 2001, Vol. 1978: 31–48.

[5] Hawkes P and Rose G. Primitive specification and supporting documentation for SOBER-t submission to NESSIE. https://www.cosic.esat.kuleuven.be/nessie/workshop/submissions.html, 2000, 11.

[6] Ekdahl P and Johansson T. SNOW-a new stream cipher. https://www.cosic.esat.kuleuven.be/nessie/workshop/submissions.html, 2000, 11.

[7] Ekdahl P and Johansson T. A new version of the stream cipher SNOW [C]. Selected Areas in Cryptography 2002 Workshop, LNCS, 2003, Vol. 2595: 47–61.

[8] Zeng Guang, Han Wenbao, and He Kaicheng. High efficiency feedback shift register: σ-LFSR. Cryptology ePrint Archive, Report 2007/114, http://eprint.iacr.org/, 2007, 03.

[9] Zeng Guang, He Kaicheng, and Han Wenbao. A trinomial type of σ-LFSR oriented toward software implementation [J]. *Science in China Series F-Information Sciences*, 2007, 50(3): 359–372.

[10] ECRYPT, eSTREAM: ECRYPT stream cipher project, IST-2002-507932, Available at http://www.ecrypt.eu.org/stream /. 2004, 5

[11] Berbain C and Gilbert H. Cryptanalysis of ABC. eSTREAM, ECRYPT stream cipher project. Report 2005/048, http://www.ecrypt.eu.org/stream, 2005, 07.

[12] Wu Hongjun and Preneel B. Cryptanalysis of ABC v2. eSTREAM, ECRYPT stream cipher project. Report 2006/029, http://www.ecrypt.eu.org/stream, 2006, 06.

[13] Zhang H N, Lin L, and Wang X Y. Fast correlation attack on stream cipher ABC v3. eSTREAM, ECRYPT stream cipher project. Report 2006/049, http://www.ecrypt.eu.org/stream, 2006, 08.

[14] Daniel J. Some more recent performance results. http://cr.yp.to/streamciphers.html, 2006.4.

[15] ECRYPT NoE. eSTREAM optimized code HOWTO. http://www. ecrypt.eu.org/stream/perf, 2005, 11.

[16] Meier W and Staffelbach O. Fast correlation attacks on certain stream ciphers [J]. *Journal of Cryptology*, 1989, 1(3): 159–176.

曾 光： 男，1980 年生，博士，研究方向为序列密码.

韩文报： 男，1963 年生，教授，博士生导师，主要研究领域为信息安全.

范淑琴： 女，1978 年生，教授，主要研究领域为密码学及其应用.