

基于正交拟群的流密码算法

张晓燕, 徐允庆*

(宁波大学 理学院, 浙江 宁波 315211)

摘要: 利用正交拟群的特点, 克服 Edon80 的弱点, 设计了称为 Double40 的二进制加法同步流密码算法. 它基于一对 8 阶相互正交的自正交拟群, 使得 Johansson 和 Hell 的密钥恢复攻击对 Double40 无法奏效.

关键词: 流密码; 拟群; 正交拉丁方

中图分类号: O157

文献标志码: A

文章编号: 1001-5132 (2016) 02-0089-05

Edon80^[1]是 Gligoroski 等设计的面向硬件的流密码算法, 于 2005 年被提交到 eSTREAM 计划. Hong^[2]研究发现 Edon80 存在一些周期很小的弱密钥. 而 Edon80 的设计者后来对此做了进一步的研究证明这些弱密钥不会影响 Edon80 的安全性^[3]. 但是 Johansson 等^[4]提出的密钥恢复攻击, 使得 Edon80 在第 3 轮筛选中被淘汰, 恢复密钥的具体步骤参见文献[1]. 如果仅仅在 Edon80 的主链上增加一些拟群运算并不能抵抗这种攻击, 除非当 e-变换的个数达到 160 时是可以的, 但是这将会导致硬件成本的增加.

笔者设计了一个新的流密码, 它基于一对 8 阶相互正交的拟群, 并由此对正交拟群可得到 8 对正交拟群. 每个序列经 e-变换作用后长度都至少增加一倍. 由于正交拟群的特点, 使得密钥流序列的数据结构完全不同于 Edon80, 这样 Double40 便可抵抗密钥恢复攻击.

1 定义和定理

目前, 对于拟群运算、拟群的字符串处理和周期^[5-7], 以及 Double40 所使用到的正交拟群的字符串处理相关的定义、定理有:

定义 1 设 Q 是一个有限集, $(Q, \circ)$ 和 $(Q, *)$ 是一对正交拟群. $a \in Q$ 是一个固定的元素, 令 $\alpha = a_1 a_2 \cdots a_m \in Q^+$, 则定义基于运算 $(\circ, *)$ 的变换 $E_{a, (\circ, *)}$:

$Q^+ \rightarrow Q^+, E_{a, (\circ, *)}(\alpha) = b_1 b_2 \cdots b_{2m}$, 其中

$$\begin{cases} b_1 = a \circ a_1, b_2 = a * a_1, \\ b_{2i+1} = b_{2i} \circ a_{i+1}, b_{2i+2} = b_{2i} * a_{i+1}, \\ i = 1, 2, \dots, m-1, \end{cases}$$

称函数 $E_{a, (\circ, *)}$ 为基于运算 $(\circ, *)$ 和引导元 a 的 Q^+ 的 e-变换, 其图形如图 1 所示.

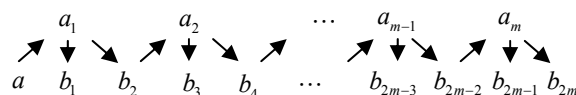


图 1 映射 $E_{a, (\circ, *)}$ 的形状

定义 2 设 Q 是一个 n 元集, σ 是 Q 上的置换. 对任意的 $x \in Q$, 称序列 $x\sigma(x)\sigma^2(x)\cdots\sigma^i(x)\cdots$ 的周期为 σ 在 x 处的周期因子, 并记为 $f_\sigma(x)$.

定义 3^[7] 设 $Q = \{0, 1, 2, \dots, n\}$, $\sigma_1, \sigma_2, \dots, \sigma_n \in Q^+$ 是拟群 $(Q, *)$ 的 n 个列置换. $\alpha = a_1 a_2 \cdots a_m \in Q^+$ 是一个周期为 p 的序列. e-变换 $E_{a, (\circ, *)}$ 如定义 1, $\beta = b_1 b_2 \cdots b_{2m} = E_{a, (\circ, *)}(\alpha_1 a_2 \cdots a_m)$. 若 a 在置换 $\sigma = \sigma_{a_p} \sigma_{a_{p-1}} \cdots \sigma_{a_1}$ 一个长度为 k 的循环中, 则序列 α 的周期为 $2kp$.

定义 4^[7] 设 Q 是一个 n 元集, $\sigma_1, \sigma_2, \dots, \sigma_n$ 是 $(Q, *)$ 的 n 个列置换, $S = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$. 记 S 生成的置换群为 $\{\tau_1, \tau_2, \dots, \tau_v\} = \langle S \rangle$, 置换 τ_i 的型 $1^{i_1} 2^{i_2} \cdots n^{i_n}$ (j^{i_j} 表示有 i_j 个长度为 j ($1 \leq j \leq n$) 的循环. 假设 Q 中元素的分布是均匀的, 即 $\{P(x) = 1/n: x \in Q\}$, 则对任意的正整数 p , 如果多重集 $S^{(p)} = \{\sigma_{i_p} \sigma_{i_{p-1}} \cdots \sigma_{i_1}: 1 \leq i_1, i_2, \dots, i_p \leq n\} = \{n_1^{(p)} \cdot \tau_1, n_2^{(p)} \cdot \tau_2, \dots, n_v^{(p)} \cdot \tau_v\}$, 那么

收稿日期: 2015-03-07.

宁波大学学报(理工版)网址: <http://journal.lg.nbu.edu.cn/>

基金项目: 国家自然科学基金(61373007); 浙江省自然科学基金(LY13F020039).

第一作者: 张晓燕(1989-), 女, 河南南阳人, 在读硕士研究生, 主要研究方向: 组合数学与密码学. E-mail: 787251165@qq.com

*通信作者: 徐允庆(1959-), 男, 河南南阳人, 教授, 主要研究方向: 组合数学与密码学. E-mail: xuyunqing@nbu.edu.cn

$(Q, *)$ 的 p 阶周期因子 $f_*^{(p)}$ 的概率分布为:

$$\left(\frac{1}{n^{p+1}} \sum_{i=1}^v n_i^{(p)} \lambda_{i1}, \frac{2}{n^{p+1}} \sum_{i=1}^v n_i^{(p)} \lambda_{i2}, \dots, \frac{n}{n^{p+1}} \sum_{i=1}^v n_i^{(p)} \lambda_{in} \right).$$

定义 5 设 $(Q, *)$ 是一个拟群, $G = \{\tau_1, \tau_2, \dots, \tau_v\}$ 是 Q 上的置换群, 设 τ_i 的型为 $1^{\lambda_{i1}} 2^{\lambda_{i2}} \dots n^{\lambda_{in}}$ ($i=1, 2, \dots, v$), 称概率分布

$$\left(\frac{1}{n \cdot v} \sum_{i=1}^v \lambda_{i1}, \frac{1}{n \cdot v} \sum_{i=1}^v 2\lambda_{i2}, \dots, \frac{1}{n \cdot v} \sum_{i=1}^v j\lambda_{ij}, \dots, \frac{1}{n \cdot v} \sum_{i=1}^v n\lambda_{in} \right)$$

为置换群 G 的轮换长度的概率分布.

推论 1 在定义 4 中, 如果置换 $\tau_i (i=1, 2, \dots, v)$ 在多重集 $S_*^{(p)}$ 中趋于均匀分布, 即 $\lim_{p \rightarrow \infty} (n_i^{(p)} / n^p) = 1/v$, 那么拟群 $(Q, *)$ 的周期因子趋近于随机变量 $f = \lim_{p \rightarrow \infty} f_*^{(p)}$, 并且它的概率分布和置换群 $\langle S_* \rangle$ 的轮换长度的概率分布是相同的.

2 Double40 的算法描述

2.1 KeySetup 模式

令 $Q = \{0, 1, 2, \dots, 7\}$, $(Q, \odot_0)$ 和 $(Q, \odot_1)$ 是一对相互正交的拟群(表 1). 设 L_1 和 L_2 分别是 $(Q, \odot_0)$ 和 $(Q, \odot_1)$ 所对应的拉丁方. 在集合 Q 上定义 2 个分别与 L_1 和 L_2 的转置相对应的拟群运算 $\odot_2$ 和 $\odot_3$, 如(1)式:

$$\begin{aligned} x \odot_2 y = z &\Leftrightarrow y \odot_0 x = z, \\ x \odot_3 y = z &\Leftrightarrow y \odot_1 x = z, \quad \forall (x, y) \in Q^2. \end{aligned} \quad (1)$$

这样, 我们就得到了 Q 上的 4 个拟群 $(Q, \odot_0)$, $(Q, \odot_1)$, $(Q, \odot_2)$ 和 $(Q, \odot_3)$.

表 1 流密码 Double40 所用到的拟群

元	$\odot_0$								$\odot_1$							
素	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	3	7	1	4	6	0	5	2	4	0	6	3	1	7	2	5
1	4	0	6	3	1	7	2	5	6	2	4	1	3	5	0	7
2	5	1	7	2	0	6	3	4	5	1	7	2	0	6	3	4
3	1	5	3	6	4	2	7	0	2	6	0	5	7	1	4	3
4	7	3	5	0	2	4	1	6	3	7	1	4	6	0	5	2
5	6	2	4	1	3	5	0	7	0	4	2	7	5	3	6	1
6	2	6	0	5	7	1	4	3	7	3	5	0	2	4	1	6
7	0	4	2	7	5	3	6	1	1	5	3	6	4	2	7	0

给出一个 120 比特的向量 $K_{\text{key}} = K_1 K_2 \dots K_{40}$, 其

中 $0 \leq K_i \leq 7$, $i=1, 2, \dots, 40$. 设 $\bullet_i = \odot_{K_i \bmod 4}$, $i=1, 2, \dots, 40$, 记 $e_{40} = e_{K_1, \bullet_{40}} \odot e_{K_2, \bullet_{39}} \odot \dots \odot e_{K_{40}, \bullet_1}$ 为 $e_{K_i, \bullet_{40-i}}$ 的复合. 设 $t_{40,1} t_{40,2} \dots t_{40,39} t_{40,40} t_{40,41} \dots t_{40,80} = e_{40}(K_1 K_2 \dots K_{40} K_{40} K_{39} \dots K_1)$, 则这个变换的运算过程如方程(2)和表 2.

$$\begin{cases} t_{1,1} = K_{40} \bullet_1 K_1, \\ t_{1,j} = t_{1,j-1} \bullet_1 K_j, \quad 1 \leq j \leq 40; \\ t_{1,j} = t_{1,j-1} \bullet_1 K_{81-j}, \quad 41 \leq j \leq 80; \\ t_{i,1} = K_{41-i} \bullet_i t_{i-1,1}, \quad t_{i,j} = t_{i,j-1} \bullet_i t_{i-1,j}, \\ 2 \leq i \leq 40, 2 \leq j \leq 80, \end{cases} \quad (2)$$

这里, 我们令

$$k_i = t_{40,i}, a_i = t_{40,40+i}, i=1, 2, \dots, 40, \quad (3)$$

将在 Keystream 模式中用到.

表 2 Keystream 模式中的拟群字符串变换

运算	引导元								序列							
$\bullet_i$	K_{41-i}	K_1	$\dots$	K_{40}	K_{40}	$\dots$	K_1									
$\bullet_1$	K_{40}	$t_{1,1}$	$\dots$	$t_{1,40}$	$t_{1,41}$	$\dots$	$t_{1,80}$									
$\bullet_2$	K_{39}	$t_{2,1}$	$\dots$	$t_{2,40}$	$t_{2,41}$	$\dots$	$t_{2,80}$									
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$									
$\bullet_{40}$	K_1	$t_{40,1}$	$\dots$	$t_{40,40}$	$t_{40,41}$	$\dots$	$t_{40,80}$									

2.2 Keystream 模式

设表 1 中拟群 $(Q, \odot_0)$ 和 $(Q, \odot_1)$ 所对应的拉丁方分别为 L_1 和 L_2 . 易证 L_1 和 L_2 都是自正交的. 拟群 $(Q, \odot_2)$ 和 $(Q, \odot_3)$ 见(1)式. 那么由这 4 个拟群可以得到 8 对不同的正交拟群运算, 分别是 $(\odot_0, \odot_1)$ 、 $(\odot_1, \odot_3)$ 、 $(\odot_2, \odot_3)$ 、 $(\odot_3, \odot_1)$ 、 $(\odot_0, \odot_2)$ 、 $(\odot_1, \odot_0)$ 、 $(\odot_2, \odot_0)$ 和 $(\odot_3, \odot_2)$. 然后在域 $\{0, 1, \dots, 7\}$ 上定义 2 个函数 f 和 g :

$$\begin{aligned} f(i) &= i \bmod 4, \\ g(i) &= (i + (i \bmod 2) + \lfloor i/4 \rfloor + 1) \bmod 4, \\ i &\in \{0, 1, \dots, 7\}, \end{aligned}$$

其中 $\lfloor i/4 \rfloor$ 表示不超过 $i/4$ 的最大整数. 容易证明 $\{(f(i), g(i)) \mid i=0, 1, \dots, 7\} = \{(0, 1), (1, 3), (2, 3), (3, 1), (0, 2), (1, 0), (2, 0), (3, 2)\}$.

并且 $(\odot_{f(i)}, \odot_{g(i)})$ ($i=0, 1, \dots, 7$) 就是上述 8 对不同正交拟群运算.

如(3)式, 令 $k_i = t_{40,i}$, $a_i = t_{40,40+i}$, $i=1, 2, \dots, 40$, 则 Keystream 模式的 e-变换见表 3. 一个周期为 8 的有限字符串 0123456701234567... 经 40 个 e-变换 $E_{a_i, (\odot_{f(k_i)}, \odot_{g(k_i)})}$ ($i=1, 2, \dots, 40$) 作用后得到 Double40 的

密钥序列, 即

$$\text{Keystream} = a_{40,1}a_{40,2}a_{40,3}a_{40,4}\cdots a_{40,2k-1}a_{40,2k}\cdots$$

这些 e-变换的循环方程如(4)式:

$$\begin{cases} a_{1,1} = a_1 \odot_{f(k_1)} 0, a_{1,2} = a_1 \odot_{g(k_1)} 0, \\ a_{1,2j-1} = a_{1,2j-2} \odot_{f(k_1)} (j-1) \pmod{8}, \\ a_{1,2j} = a_{1,2j-2} \odot_{g(k_1)} (j-1) \pmod{8}, j \geq 2, \\ a_{i,1} = a_i \odot_{f(k_i)} a_{i-1,1}, a_{i,2} = a_i \odot_{g(k_i)} a_{i-1,1}, \\ 2 \leq i \leq 40, \\ a_{i,2j-1} = a_{i,2j-2} \odot_{f(k_i)} a_{i-1,j}, a_{i,2j} = \\ a_{i,2j-2} \odot_{g(k_i)} a_{i-1,j}, \\ 2 \leq i \leq 40, j \geq 2. \end{cases} \quad (4)$$

表3 Keystream 模式的 e-变换

运算	引导元	序列
$(\odot_{f(k_1)}, \odot_{g(k_1)})$	a_1	0123456701234567...
$(\odot_{f(k_1)}, \odot_{g(k_1)})$	a_1	$a_{1,1} \quad a_{1,2} \quad \cdots \quad a_{1,32} \quad \cdots$
$(\odot_{f(k_2)}, \odot_{g(k_2)})$	a_2	$a_{2,1} \quad a_{2,2} \quad a_{2,3} \quad a_{2,4} \quad \cdots \quad a_{2,63} \quad a_{2,64} \quad \cdots$
$\vdots$	$\vdots$	$\vdots$
$(\odot_{f(k_{40})}, \odot_{g(k_{40})})$	a_{40}	$a_{40,1} \cdots a_{40,2^{39}} \cdots a_{40,2^{43}+1} \cdots a_{40,2^{44}} \cdots$

3 Double40 的功能描述

二进制加法流密码 Double40 的原理如图 2 所示. 图 3 给出了 Double40 的组件及其相互之间的关系. 其包含 2 个模式: Keysetup 模式和 Keystream 模式, 包含组件如下:

- (1) 1 个 120 比特的寄存器 Key 来存储密钥.
- (2) 1 个 6 比特的计数器(用在 Keysetup 模式).
- (3) 1 个 3 比特的计数器(将在 Keystream 模式中用到).

(4) 2 个 8 阶自正交的拟群 $(Q, \odot_0)$ 和 $(Q, \odot_1)$, 并且这 2 个拟群共占用了 $2 \times 24 = 48$ 字节的存储空间. 给 Double40 一个时间变量 t , 对于每个 t , Double40 都将进行若干 e-变换(从第 d 个开始, 到第 40 个结束), 然后输出密钥序列中的一对值 $a_{40,2t-1}$ 和 $a_{40,2t}$.

Double40 还包含一个大小为 39 比特, 用于存储时间变量 t 的计数器和一个由 81 个 3 比特的存储单元构成的 RAM. 我们用 $x_0, x_{i,1}, x_{i,2}, 1 \leq i \leq 40$, 来表示该 RAM 中的数据, 并且 x_0 的初始值为 7, $x_{i,2} = a_i, 1 \leq i \leq 40$. 首先, 由变量 t 的值可得 d 的取值, 如(5)式:

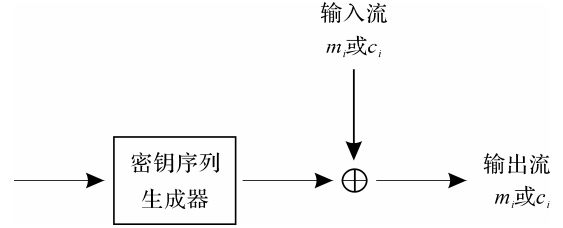


图2 Double40 的原理图

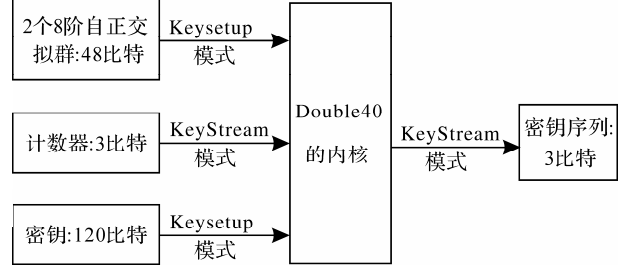


图3 Double40 的组件及其相互关系

$$d = \begin{cases} 1, t \equiv 1 \pmod{2^{39}}, \\ 40 - n, \text{如果存在一个奇数 } t \text{ 使得} \\ \quad t \equiv l \cdot 2^n + 1 \pmod{2^{39}}, \\ 40, t \equiv 0 \pmod{2}, \end{cases} \quad (5)$$

接下来 RAM 中的数据将会按照以下方式更新:

当 $d=1$ 时,

$$\begin{cases} x_0 \leftarrow x_0 + 1 \pmod{8}, \\ x_{1,1} \leftarrow x_{1,2} \odot_{f(k_1)} x_0, x_{1,2} \leftarrow x_{1,2} \odot_{g(k_1)} x_0, \\ x_{i,1} \leftarrow x_{i,2} \odot_{f(k_i)} x_{i-1,1}, \\ x_{i,2} \leftarrow x_{i,2} \odot_{g(k_i)} x_{i-1,1}, \quad 2 \leq i \leq 40, \end{cases}$$

当 $d>1$ 时,

$$\begin{cases} x_{d,1} \leftarrow x_{d,2} \odot_{f(k_d)} x_{d-1,2}, x_{d,2} \leftarrow \\ \quad x_{d,2} \odot_{g(k_d)} x_{d-1,2}, \\ x_{i,1} \leftarrow x_{i,2} \odot_{f(k_i)} x_{i-1,1}, \\ x_{i,2} \leftarrow x_{i,2} \odot_{g(k_i)} x_{i-1,1}, \quad d+1 \leq i \leq 40. \end{cases}$$

从而, 对于每一个 d , 都有一对 RAM 的输出值 $x_{40,1}$ 和 $x_{40,2}$, 并且它们就是密钥序列中的一对值.

4 Double40 的安全性分析

Double40 包含 2 个内部状态 $(\odot_{f(k_i)}, \odot_{g(k_i)})$ 和 $a_i, i=1,2,\cdots,40$, 并且它们的空间大小均为 8^{40} , 从而该流密码的内部空间为 $8^{40} \times 8^{40} = 2^{240}$, 而对密钥进行穷举搜索攻击的复杂度为 2^{120} , 因此如果采取对 Double40 的内部空间搜索的攻击方法将比穷举搜索攻击更复杂.

4.1 相关密钥攻击和猜测试验攻击的安全性分析

关于相关密钥攻击和猜测试验攻击,可参照文献[1],得出其复杂度分别为 2^{120} 和 $64^{40}=2^{240}$. 因此,密钥恢复攻击将优于这 2 种攻击方法.

4.2 Double40 的密钥周期分析

拟群 $(Q, \odot_0)$ 见表 1, 依次记 $(Q, \odot_0)$ 的列置换为 $\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7$. 由于这些列置换都是偶置换, 故可得:

$$\langle \sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7 \rangle \subseteq \text{Alt}(Q). \quad (6)$$

因为 $\sigma_1=(1074352)$, $\sigma_2^2=(610)$, 所以由文献[8]可知:

$$\langle \sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7 \rangle \supseteq \langle \sigma_1, \sigma_2^2 \rangle \supseteq \text{Alt}(Q), \quad (7)$$

由(6)式和(7)式得:

$$\langle \sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7 \rangle = \text{Alt}(Q).$$

容易证明 $\text{Alt}(Q)$ 中置换的型为 $1^8, 1^5 3^1, 1^4 2^2, 1^3 5^1, 1^2 2^1 4^1, 1^2 3^2, 1^1 2^2 3^1, 1^1 7^1, 2^4, 2^1 6^1, 3^1 5^1, 4^2$. 用 $T_i, i=1, 2, \dots, 12$ 来表示以上这些置换的型, t_i 表示型为 T_i 的置换个数. 那么 t_i 的取值见表 4.

表 4 $\text{Alt}(Q)$ 中类型为 T_i 的置换个数

i	1	2	3	4	5	6
t_i	1	112	210	1 344	2 520	1 120
i	7	8	9	10	11	12
t_i	1 680	5 760	105	3 360	2 688	1 260

根据定义 5, $\text{Alt}(Q)$ 的轮换长度的概率分布为:

$$\left(\frac{1}{8 \cdot 8!} \sum_{i=1}^{12} t_i \lambda_{i1}, \frac{2}{8 \cdot 8!} \sum_{i=1}^{12} 2 t_i \lambda_{i2}, \dots, \frac{n}{8 \cdot 8!} \sum_{i=1}^{12} 8 t_i \lambda_{i8} \right) = \left(\frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{4} 0 \right). \quad (8)$$

用 $e_{a_i, \odot_0}$ 对字符串 $e^{(i-1)}(0123456701234567 \dots)$ 作用, 经实验可得 $e_{a_i, \odot_0}$ 的周期因子 R_i 服从(8)式所示的概率分布. 并且拟群 $(Q, \odot_1), (Q, \odot_2)$ 和 $(Q, \odot_3)$ 各自生成的置换群及其周期因子的概率分布都和拟群 $(Q, \odot_0)$ 的相同.

由定理 3 知, $E_{a_i, (\odot_{f(k_i)}, \odot_{g(k_i)})}$ 的周期因子 Y_i 趋向于如下分布:

$$\left(\frac{2}{8} \frac{4}{8} \frac{6}{8} \frac{8}{8} \frac{10}{8} \frac{12}{8} \frac{14}{8} \frac{16}{8} \right) \left(\frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{8} \frac{1}{4} 0 \right). \quad (9)$$

设 $Y=8 \prod_{i=1}^{40} Y_i$, 则根据 Y_i 的概率分布可以得到 Y 的近似概率密度函数:

$$f(x) = \frac{1}{x \sqrt{32.7762\pi}} \cdot \exp \left(-\frac{(\ln x - 82.1606)^2}{32.7762} \right). \quad (10)$$

关于 Double40 的密钥序列的部分周期概率见表 5.

表 5 Double40 的密钥序列的部分周期概率

周期	概率	周期	概率
$<2^{43}$	0	$<2^{120}$	0.599 2
$<2^{100}$	2~10.37	$<2^{130}$	0.975 2
$<2^{110}$	0.072 0	$<2^{140}$	0.999 8

4.3 密钥恢复攻击的安全性分析

Johansson 和 Hell 在分析 Edon80 的密钥序列周期的基础上提出了密钥恢复攻击, 该攻击只需要 2^{72} 次运算就能恢复密钥. 此种攻击的前提是假设已经知道密钥序列, 然后利用 Edon80 以下 2 个性质来恢复密钥.

(1)e-变换所使用的拟群的选取完全直接由密钥 $K_0, K_1, \dots, K_{79}$ 决定. 这样, 如果攻击者知道了第 i 个 e-变换所使用的拟群, 那么就得到了 K_i 的值.

(2)Edon80 中 e-变换的素周期因子只有 2 和 3, 对于很小的 i , 经第 i 个 e-变换作用后生成的新字符串的周期容易被猜出.

而 Double40 中 e-变换所依赖的正交拟群的选取并不是直接由密钥决定, 这在 Keystream 模式中有详细说明. 另一方面, Double40 中 e-变换的素周期因子有 2, 3, 5 和 7, 且周期因子的期望值增大使得密钥序列的周期大大增加. 这样经 e-变换作用后生成的新字符串的周期就很难被猜测到. 由此, Double40 便可以抵制密钥恢复攻击.

5 结语

Double40 是基于一对 8 阶相互正交的自正交拟群的二进制加法流密码, 由此一对拟群可以生成 8 对正交拟群. 由于正交拟群的特殊性, 使得密钥序列的数据结构由原来 Edon80 的“矩阵结构”变为“脉冲结构”. 我们在 Double40 的 Keystream 模式中引进 2 个函数 $f(i)$ 和 $g(i)$, 使得 e-变换所依赖

的正交拟群的选取不直接由密钥决定. 由于以上 2 点的改进, 使得 Double40 将能够抵制 Johansson 和 Hell 提出的密钥恢复攻击.

参考文献:

- [1] GLIGOROSKI D, MARKOVSKI S, KNAPSKOG S J. The stream cipher Edon80[M]. Berlin: Springer, 2008: 152-169.
- [2] HONG J. Period of stream cipher Edon80[M]. Berlin: Springer, 2005:23-34.
- [3] GLIGOROSKI D, MARKOVSKI S, KNAPSKOG S J. On periods of Edon-(2m,2k) family of stream cipher[J]. SASC Stream Cipher Revisited, 2006, 216:216-227.
- [4] JOHANSSON T, HELL M. A key recovery attack on Edon80[M]. Berlin: Springer, 2007:568-581.
- [5] DENES J, KEEDWELL A D. Latin squares and their applications[M]. New York: Academic Press, 1974:63-125.
- [6] MARKOVSKI S, GLIGOROSKI D, BAKEVA V. Quasigroup string processing: Part 1[J]. Proc of Maced Acad of Sci and Arts for Math and Tech Scien, 1999, 13(2):1-2; 13-28.
- [7] XU Y. On the key-stream periods probability of Edon80[C]//Pre-proceedings of The 9th China International Conference on Information Security and Cryptology, Berlin: Springer International Publishing, 2014:50-64.
- [8] DIXON J D, MORTIMER B. Permutation groups[M]. New York: Springer-Verlag, 1996:3-4.

Stream Cipher Based on Mutually Orthogonal Quasigroups

ZHANG Xiao-yan, XU Yun-qing *

(Faculty of Science, Ningbo University, Ningbo 315211, China)

Abstract: Edon80 is a hardware binary additive synchronous stream cipher submitted to the last phase of the eSTREAM project. The core of the cipher consists of quasigroup string e-transformations and employs four quasigroups of order 4. The best attack on Edon80 is the key recovery attack given by Johansson and Hell, which can recover the key with complexity around 2^{72} . In this paper, using the characteristics of orthogonal quasigroup to overcome the shorting of Edon80, a binary additive synchronous stream cipher named Double40 is designed. It is based on a pair of mutually orthogonal quasigroups of order 8 which are self-orthogonal. The developed algorithm disables the key recovery attack of Johansson and Hell on Double40.

Key words: stream cipher; quasigroup; orthogonal Latin square

(责任编辑 史小丽)