

基于字的流密码 Dragon 的分析

李 媛, 仵丽花, 胡予濮

(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, 西安 710071)

摘 要: 研究一种新型的流密码——Dragon。Dragon 使用了非线性反馈移位寄存器(NLFSR)和 S 盒, 密钥长度是可变的 128 bit 或 256 bit。探讨了 Dragon 的设计原理, 从内部结构角度分析讨论其安全性, 指出 Dragon 对暴力攻击和 TMD 攻击是安全的, 同时构造了 Dragon 的线性逼近式, 给算法提了 2 点建议。

关键词: 流密码; 线性分析; Dragon 算法

Cryptanalysis of Word Based Stream Cipher Dragon

LI Yuan, WU Li-hua, HU Yu-pu

(Key Lab of Computer Network and Information Security, Ministry of Education, Xidian University, Xi'an 710071)

【Abstract】 This paper studies a new tube stream cipher, Dragon. Dragon uses non-linear feedback shift registers and S boxes. It operates on key sizes of 128 bit and 256 bit. Its mechanism, performance and design principles are studied, and the security against the well-known cryptanalysis is discussed. It is pointed out that Dragon is secure against the brute and TMD attack and also the linear approximations are presented. Some suggestions for Dragon are proposed.

【Key words】 stream ciphers; linear analysis; Dragon algorithm

2004 年开启的 ECRYPT(European Network of Excellence for Cryptology)^[1] 是为期 4 年的信息安全项目。它的主要目标是发展安全快速的流密码。根据 4 个征集原则一共征集到了 34 个流密码算法。按类别可分为: 基于线性反馈移位寄存器(LFSR)的设计, 基于非线性反馈移位寄存器(NFSR)的设计, 基于表驱动的设计, 利用分组密码部件构造流密码。本文介绍的 Dragon 是基于分组密码部件构造的流密码。

1 Dragon 算法

Dragon^[2] 是基于字的流密码, 该算法由一个基于 64 bit 的寄存器 M 表示的线性部分和一个由 1 024 bit 的 F 表示的非线性部分组成。用可变 128 bit 或 256 bit 的密钥长度和初始化变量每轮产生 64 bit 的密钥流。Dragon 的核心是 2 个高度非线性的 8×32 的 S 盒。

非线性部分 F 结构如图 1 所示。它的作用有 2 个: 一个是密钥建立; 另一个是密钥流生成。 F 是由 6 个 32 bit 的字输入产生 6 个 32 bit 字的输出。

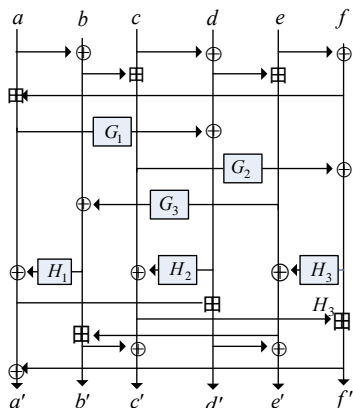


图 1 F 的结构

从图 1 可以看到, 输入的字用 a, b, c, d, e, f 表示, 输出的字用 a', b', c', d', e', f' 表示, \oplus 表示的是二进制的异或, \boxplus 表示的是模 2^{32} 加, F 是一个可逆映射, G 和 H 具有高非线性性, G 由 3 个 S_1 、1 个 S_2 组成, H 由 1 个 S_1 、3 个 S_2 组成, S_1, S_2 是 8×32 的 S 盒。输入的 32 bit 被分为 4 个 8 bit $x = (x_0 \ x_1 \ x_2 \ x_3)$ 作为 S 盒的输入, 可把它分为 3 个部分: 混淆前, 置换, 混淆后。

密钥初始化过程中 1 024 bit 的内部状态是密钥和初始向量的级联, 这 1 024 bit 被分为 8 个 128 bit 的字, 标记为 $w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7$ 。密钥产生的过程中非线性移位寄存器的 1 024 bit 的内部状态用 32 个 32 bit 的字表示, 记为 $B_i, 0 \leq i \leq 31$ 。在每轮中, 内部状态的 6 个字被选作 F 的输入, 它们的抽头系数是 0, 9, 16, 19, 30, 31。

密钥初始化过程:

输入 = $\{K, IV\}$ (256 bit), 输入 = $\{k, iv\}$ (128 bit)

(1) $W_0 \parallel W_1 \parallel \dots \parallel W_7 = K \parallel K \oplus IV \parallel K \oplus IV \parallel IV$ (256 bit),

$W_0 \parallel W_1 \parallel \dots \parallel W_7 = k \parallel k' \oplus iv \parallel k \oplus iv \parallel k \oplus iv \parallel k' \parallel k \oplus iv \parallel iv \parallel k' \oplus iv$ (128 bit)

(2) $M = 0X0000447261676F6E$

(3) $a \parallel b \parallel c \parallel d = (W_0 \oplus W_6 \oplus W_7)$

(4) $e \parallel f = M$

(5) $\{a', b', c', d', e', f'\} = F(a, b, c, d, e, f)$

(6) $W_0 = (a' \parallel b' \parallel c' \parallel d') \oplus W_4$

(7) $W_i = W_{i-1} (1 < i < 7)$

(8) $M = e' \parallel f'$

输出 = $\{W_0 \parallel W_1 \parallel \dots \parallel W_7\}$

基金项目: 国家自然科学基金资助项目(60473029); 国家部委预研基金资助项目

作者简介: 李 媛(1982 -), 女, 硕士研究生, 主研方向: 流密码; 仵丽花, 硕士研究生; 胡予濮, 教授、博士生导师

收稿日期: 2007-11-06 **E-mail:** zsyf08@163.com

密钥流产生过程如下：

输入 = $\{B_0 \parallel B_1 \parallel \dots \parallel B_{31}, M\}$

(1) $(M_L \parallel M_R) = M$

(2) $a = B_0, b = B_3, c = B_{16}, d = B_{19}, e = B_{30} \oplus M_L, f = B_{31} \oplus M_R$

(3) $(a', b', c', d', e', f') = F(a, b, c, d, e, f)$

(4) $B_0 = b', B_1 = c'$

(5) $B_i = B_{i-2}, 2 \leq i \leq 31$

(6) $M = M + 1$

(7) $k = a' \parallel e'$

输出 = $\{k, B_0 \parallel B_1 \parallel \dots \parallel B_{31}, M\}$

2 原理分析

基于字的流密码是以字为单位进行密钥加解密，而传统的流密码是以bit为单位的，所以基于字的流密码比传统的流密码的每个时钟产生的密钥流要多，不用以牺牲效率来换取更多的密钥流，所以基于字的流密码可以解决安全与效率之间的折中问题。例如基于 8 bit 字的 RC4 和基于 32 bit 字的 Turing，以新的加密标准来衡量的话，在软件上比分组密码还快。Dragon 不仅密钥流产生快，密钥流的生成也很有效，这使 Dragon 可以应用到移动通信和无线通信领域。Dragon 克服了分组密码的 OFB 模式的缺点，即 OFB 模式要求密钥块不能碰撞^[3]。

Dragon 用 2 个 8×32 的 S 盒构成 G 和 H，实现高度的非线性性，S 盒的设计满足以下性质：

(1) 理想的非线性度是 116；

(2) 根据布尔函数代数次数和相关免疫阶之间的制约关系，得到最合适的代数次数是 6 或者 7；

(3) 低的自相关函数。

用符号 (n, t, d, x, y) 来描述布尔性质。其中， n 代表变量的个数； t 为弹性系数； d 代数次数； x 为非线性度； y 为最大自相关函数。当 S_1 达到最高的非线性性，它的布尔性质满足 $(8, 1, 16, 116, y)$ ， $32 \leq y \leq 48$ ，当 S_2 达到最低自相关度时， S_2 的布尔性质为 $(8, 0, 7, 116, 24)$ 。G 和 H 的非线性可达到 116，比现在流行的 SBOX/MIXCOL (非线性性是 112) 的非线性还要高。密钥建立和密钥流产生都是由 F 生成的，但密钥建立过程中的 F 和密钥流生成时的 F 的设计是不同的，如表 1 所示。

表 1 密钥建立和密钥流生成不同部件的不同作用

	密钥建立	密钥流生成
64 位 M 的作用	寄存器	计数器
反馈的个数	128 bit	64 bit
抽头系数	{0, 4, 6, 7}	{0, 9, 16, 19, 30, 31}

3 安全性分析

假设 Dragon 的 1 024 bit 的内部状态是伪随机的，理想的周期是 2^{512} 。每一轮的 Dragon 都要受到 64 位寄存器的影响，寄存器的周期为 2^{64} ，所以总体考虑 Dragon 的周期为 $2^{512} \times 2^{64} = 2^{576}$ 。

伪随机序列生成器是很多流密码密钥流生成器中不可缺少的一部分。一个好的流密码设计要求密钥流序列拥有稳定的或者说足够长的周期。Dragon 的周期是一个长的周期，长的周期为 Dragon 提供更好的安全性，可以使 Dragon 更好地抗击未知攻击。

一般说来分组与密钥长度越长意味着安全性越高，但实现的计算复杂性也越高，速度越低。分组与密钥长度可变的分组密码算法能够满足不同的安全性要求，同时又能保证算法的速度。Dragon 就是这样的可变密钥长度，可以根据需要

来选择密钥流的长度是 128 bit 还是 256 bit。而且 Dragon-128 和 Dragon-256 有相同的初始化过程，以至于有相同的速度，但是它们的密钥和初始向量是 128 bit 或 256 bit，这可以保证没有任何一对 256 bit 的 (K, IV) 与任何一对的 128 bit 的 (k, iv) 有相同的初始状态，以此避免暴力攻击。

TMD (Time-Memory Tradeoff Attack) 是一种选择明文攻击方法^[4]，它由穷尽密钥搜索攻击和查表攻击 2 种方法混合而成，在选择明文攻击中以时间换取空间。Dragon 的内部状态为 1 088 bit (包括 64 位的寄存器)，暴力攻击的时间 $T = O(2^{576})$ ，数据 $D = O(2^{64})$ ，由公式 $T \times M^2 \times D^2 = S^2$ 可得到 $M = O(2^{896}) > \text{Dragon 的 } M$ ，所以 Dragon 可以抗击 TMD 攻击。

Dragon 每产生 2^{64} bit 密钥流时重置一次密钥，而 Dragon 的周期是 2^{576} ，就是说还没有到达一个周期已经开始下一轮，由此，Dragon 可以抗击未知攻击，所以总体来说 Dragon 是安全的。

当寄存器的内部状态已知时，可以预知寄存器 M 的变化，增加了攻击的可能，为了更好地抗击攻击，可以把 M 设计为不可预知的。

4 线性逼近

线性逼近的问题可以作这样的描述：设 ϕ 是定义在某个确定空间 A 上的函数集，f 是定义在 A 上的函数。判断能否得到一个接近函数 f 的线性组合，也就是用一个线性表达式去逼近一个非线性表达式。这里有一个问题要解决，就是线性组合和函数 f 之间的偏差的度量，也叫偏移量。

Dragon 的线性逼近以及偏移量可分为 3 步来计算：

(1) 通过构造 S 盒的线性式计算 G 和 H 线性式以及它们的偏移量；

(2) 计算把田近似为 \oplus 的偏移量；

(3) 找到 f 的最佳近似表达，并计算偏移量。

对各变量进行说明：

$x = (x_{n-1}, x_n, \dots, x_0)$, $y = (y_{n-1}, y_n, \dots, y_0)$, $x, y \in GH(2^n)$

$x \cdot y = x_{n-1}y_{n-1} \oplus x_ny_n \oplus \dots \oplus x_0y_0$

其中，n 表示非负整数。

函数 $f: (0, 1)^m \rightarrow (0, 1)^n$, m, n 是正整数

引理 1^[5] 把 $A \cdot f(x) = B \cdot x$ 的近似表达的偏移量定义为

$\epsilon_f(A, B) = 2^{-n} (\#(A \cdot f(x) \oplus B \cdot x = 0) - \#(A \cdot f(x) \oplus B \cdot x = 1))$

其中，A, B 表示向量。

由于 G, H 是由 S_1, S_2 构成的，因此线性近似 G, H 可以看作近似 S_1, S_2 ，对引理 1 中的不同的 $f(x)$, B 进行讨论得到表 2。

表 2 近似表达式与偏移量对照表

近似表达式	偏移量
$A \cdot G_1(x) = A \cdot x$	$\in G_1(A, A)$
$A \cdot G_2(x) = A \cdot x$	$\in G_2(A, A)$
$A \cdot G_3(x) = A \cdot x$	$\in G_3(A, A)$
$A \cdot H_1(x) = 0$	$\in H_1(A, 0)$
$A \cdot H_2(x) = 0$	$\in H_2(A, 0)$
$A \cdot H_3(x) = 0$	$\in H_3(A, 0)$

因为 $x = x_0 \parallel x_1 \parallel x_2 \parallel x_3$ ，所以 $A \cdot H_1(x) = 0$ 可以表示为

$A \cdot H_1(x) = A \cdot S_2(x_0) \oplus A \cdot S_2(x_1) \oplus A \cdot S_2(x_2) \oplus A \cdot S_1(x_3) = 0$ (1)

$A \cdot H_1(x) = 0$ 可以表示为

偏移量 $\in H_1(A, 0) = \in S_2(A, 0)^3 \times \in S_1(A, 0) \in S_1(A, 0)$ 是 $A \cdot$

$S_i(x_j)=0$ 的近似的偏移量, $A \cdot H_3, A \cdot H_2$ 和 $A \cdot H_1$ 是等价的。所以 $\in H_1(A,0)=\in H_2(A,0)=\in H_3(A,0)$, 计算 G 的偏移量假设 $A=A_0 \parallel A_1 \parallel A_2 \parallel A_3$, 有

$$\begin{aligned} A \cdot (G_1(x) \oplus x) &= (A \cdot S_1(x_0) \oplus A_0 \cdot x_0) \oplus (A \cdot S_1(x_1) \oplus A_1 \cdot x_1) \oplus \\ &\quad (A \cdot S_1(x_2) \oplus A_2 \cdot x_2) \oplus (A \cdot S_1(x_3) \oplus A_3 \cdot x_3) = 0 \\ \in G(A,A) \text{ 的偏移量可以表示如下:} \\ \in G(A,A) &= \in S_1(x_0)(A, A_0) \times \in S_1(x_1)(A, A_1) \times \in S_1(x_2) \\ &\quad (A, A_2) \times \in S_1(x_3)(A, A_3) \end{aligned} \quad (2)$$

设田到 \oplus 的近似偏移量记作 $\in_+(A,A)$, 有

$$Pr[A \cdot (x \boxplus y) = A \cdot (x \oplus y)] = \frac{1}{2}(1 + \in_+(A,A)) \quad (3)$$

引理 2 m, n 是正整数 $A = (a_{n-1}, a_{n-2}, \dots, a_0), a_i \in \{0,1\}$, m 是向量 A 的汉明重量, 用 W 表示向量 A 为 1 的位置, $W = (w_{m-1}, w_{m-2}, \dots, w_0)$, $0 \leq w_j \leq n$ 。

当 m 为奇数时,

$$\in_+(A,A) = \sum_{i=0}^{m/2} (w_{2i+1} - w_{2i}) \quad (4)$$

当 m 为偶数时,

$$\in_+(A,A) = \sum_{i=1}^{(m-1)/2} (w_{2i} - w_{2i-1}) + w_0 \quad (5)$$

$$\begin{aligned} a' &= [(a \boxplus (e \oplus f)) \boxplus H_1] \boxplus [(e \oplus f \oplus G_2) \boxplus (H_2 \oplus ((a \oplus b) \boxplus c))] \\ A \cdot a' &= A \cdot [(a \boxplus (e \oplus f)) \boxplus H_1] \boxplus [(e \oplus f \oplus G_2) \boxplus (H_2 \oplus ((a \oplus b) \boxplus c))] \end{aligned}$$

G_2 的输入为 $(a \oplus b) \boxplus c$, 所以 $A \cdot G_2 = A \cdot [(a \oplus b) \boxplus c]$ 的偏移量 $\in G_2(A,A)$ 。

根据式(1)~式(3), $A \cdot a' = A \cdot a$ 的偏移量为

$$\in a'(A,A) = \in_+(A,A)^2 \times \in H_1(A,0) \times \in H_2(A,0) \times \in G_2(A,A)$$

同理: $A \cdot e' = A \cdot e$ 的偏移量为

$$\in e'(A,A) = \in_+(A,A)^2 \times \in H_1(A,0) \times \in H_3(A,0) \times \in G_1(A,A)$$

综合引理 1 和引理 2 对 Dragon 的总的偏移量分析如下: 32 位 a' 可以代表 64 位密钥流的前半部分, e' 代表 64 位密钥流的后半部分, s 代表密钥流, N_a, N_e 分别代表 a 到 a' 和 e 到 e' 的偏移。由 $e^{(i+15)} = a^{(i)} \oplus N_L^{(i+15)}$ 得

$$\begin{aligned} s(t) &= a^{(t)} \oplus e^{(t+15)} = (a^{(t)} \oplus N_a^{(t)}) \oplus (a^{(t)} \oplus M_L^{(t+15)} \oplus \\ &\quad N_e^{(t+15)}) = N_a^{(t)} \oplus N_e^{(t+15)} \oplus M_L^{(t+15)} \end{aligned} \quad (6)$$

如果可以正确猜出 M 的初始状态, 在正确猜对 M 的条件

(上接第 145 页)

是一次性的, 有效地防止了中间人的攻击, 保证了私有信息的保密性和完整性。

(4)安全的交易

1)客户端对服务器端的验证。客户端并没有服务器端的对称密钥, 但客户端通过验证 $h_s = H(x,k)$, 确信与其通信的服务器端拥有客户端的对称密钥 k , 这样也就间接地验证了服务器端的身份。

2)服务器端对客户端的验证。每次认证过程中, 客户端都要根据服务器发送过来的随机数和图像口令的特征值计算出一次性密钥。如果客户端是一个假冒者, 则他提供的一次性密钥必然无法与服务器生成的一次性密钥相匹配, 服务器会拒绝本次访问。

6 结束语

本文介绍了一次性口令和图像口令相结合进行身份认证

下式(6)的偏移量为

$$\begin{aligned} N_a^{(i)} \oplus N_e^{(i+15)} &= a'(A,A) \times \in e'(A,A) = \in_+(A,A)^4 \times \in H_1(A,0)^2 \times \\ &\quad \in H_2(A,0) \times \in H_3(A,0) \times \in G_1(A,A) \times \in G_2(A,A) \end{aligned}$$

由于 $\in_+(A,A)$ 的大小和 A 的汉明重量有关, 但并不是线性变化的, 当 A 的汉明重量增加时, $\in_+(A,A)$ 的偏移量随指数减小, 因此当 A 的汉明重量比较小时, 就有可能达到比较大的偏移量。为了减小未知内部状态的偏移量, 需要猜测 M_L 的前 27 bit 和 M_R 的 32 bit。因此要存储 $2^{27} + 2^{32} = 2^{59}$ bit 可能的内部状态值。

5 结束语

最后给算法提出 2 点建议:

(1)密钥生成过程的第(4)步应该和第(5)步交换一下, 原算法是先给 $B_0 = b', B_1 = c'$ 赋值, 再执行 $B_i = B_{i-2}, 2 \leq i \leq 31$, $B_0 = B_2 = b', B_1 = B_3 = c', B_i = B_{i-2} (4 \leq i \leq 31)$ 。如果把第(4)步和第(5)步交换一下, 就得到 $B_0 = b', B_1 = c', B_i = B_{i-2} (2 \leq i \leq 31)$ 。

如果是原文中那样的话, 前 4 个字中有 2 个是重复的, 这样不安全, 而改进的算法可以克服这个缺点。

(2)在密钥生成过程中 $M(64 \text{ bit})$ 是一个计数器, 也有可能超出 M 的取值范围, 最好在第(6)步保证模上 $2^{64} - 1$ 。

下一步的工作是分析 S 盒的特性, 找到 S 盒的缺陷, 即找到 Dragon 算法的缺陷。

参考文献

- [1] Chen K, Henricksen M, Millan W. Dragon: A Fast Word Based Stream Cipher[EB/OL]. (2007-01-10). <http://www.ecrypt.eu.org/stream/>.
- [2] Englund H, Maximoy A. Attack the Dragon[EB/OL]. (2007-01-10). <http://www.ecrypt.eu.org/stream/dragonp3.html>.
- [3] Cho J K, Pieprzyk J. A Linear Distinguisher for Dragon[EB/OL]. (2007-01-10). <http://www.ecrypt.eu.org/stream/dragonp3.html>.
- [4] 冯登国. 密码分析学[M]. 北京: 清华大学出版社, 2000-08.
- [5] Cho J Y, Pieprzyk J. Multiple Modular Addition and Crossword Puzzle Attack on NLSv2[EB/OL]. (2006-12-15). <http://www.ecrypt.eu.org/stream/nlsp2.html>.

参考文献

- [1] Haller N M. The S/Key One-time Password System[C]//Proceedings of the Internet Society Symposium on Network and Distributed System Security. San Diego, CA, USA: [s. n.], 1994.
- [2] Blonder G. Graphical Passwords: United States, 5559961[P]. 1996.
- [3] Perrig A. Hash Visualization: A New Technique to Improve Real-world Security[C]//Proceedings of CryTEC'99. Hong Kong, China: [s. n.], 1999.
- [4] Ralph N H. How We Remember What We See[Z]. 1970.