流密码 Rabbit 的安全性分析

张振广 胡予濮 王 璐
(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 通过分析流密码算法 Rabbit 的设计弱点,提出了一种针对 Rabbit 密钥流生成器的密钥恢复攻击。攻击分3个阶段分别猜测96bits、96bits和5bits依次恢复状态变量、计数器变量以及密钥种子。结果表明,整个过程的预计算复杂度为 $O(2^{96})$,时间复杂度为 $O(2^{97})$,所需存储空间为 $O(2^{95.81})$。与已有的攻击算法相比,其增加了预计算复杂度和存储空间,但降低了时间复杂度。

关键词 Rabbit,流密码,密钥恢复攻击,存储空间
中图法分类号 TN918.1 **文献标识码** A

Cryptanalysis of Rabbit

ZHANG Zhen-guang HU Yu-pu WANG Lu
(Key Laboratory of Computer Network and Information Security of Ministry of Education, Xidian University, Xi'an 710071, China)

Abstract By analyzing the weakness in design of the stream cipher Rabbit, we presented key-recovery attack on it. After guessing 96bits, 96bits and 5bits in turn, we can obtain the internal variables, the counter variables and the secret keys in sequence. The result shows that precomputation complexity of whole process is $O(2^{96})$, time complexity is $O(2^{97})$, request memory space is $O(2^{95.81})$. Comparing with existing attack algorithm, it adds precomputation complexity and memory space, but reduces time complexity.

Keywords Rabbit, Stream cipher, Key-recovery attack, Memory space

作为 ECRYPT 工程的最终算法之一,流密码 Rabbit[1] 包含 128bits 的密钥种子和 64bits 的初始向量 IV,经过初始化扩充为 513bits 的内部状态。Cryptico A/S 对其安全性进行了深入的分析,其中包括代数分析[2]、设置密钥种子的安全性[3]、周期性[4]、二阶逼近[5]及设置 IV 的安全性[6]等。有两篇文献对该算法进行了攻击。在文献[7]中,作者通过对密钥流比特偏差的研究设计了复杂度为 $O(2^{247})$ 的区分攻击。文献[8]对文献[7]做了进一步改进,用快速傅里叶变换解出密钥流比特的偏差,实施了复杂度更低的区分攻击,此外作者还提出了一种基于多帧的密钥恢复攻击,其预计算复杂度为 $O(2^{32})$,时间复杂度为 $O(2^{97.5})$,所需存储空间为 $O(2^{32})$。

g 函数是算法中的一个重要部件,用于产生下一轮的状态变量。本文首先对其性质进行考察,得到如下结论:当 2^n+1 含平方因子时,函数不是双射,降低了算法的安全性。其次,研究发现当状态更新函数的输入之间存在(2 的幂次方)倍数关系时,输出之间存在 9 种可能的线性关系,根据上述关系进行分类并存储每个类中的(输入)最小的一个。当攻击者截获多个密钥流子块时,就可以分 3 个阶段对算法实施密钥恢复攻击:第一阶段猜测 96bits 由密钥流子块 S_5 恢复状态变量 $x_{j,5}$,第二阶段同样猜测 96bits 由密钥流子块 S_6 恢复计数器变量 $c_{j,6}$,第三阶段猜测 5bits 恢复密钥种子。整个过程的预计算复杂度为 $O(2^{96})$,时间复杂度为 $O(2^{97})$,所需存储空间为 $O(2^{95.81})$。与文献[8]中的密钥恢复攻击相比,时间复杂度略低,其他两项较高,可见两种攻击方法各有利弊。

1 算法描述

1.1 符号说明

本文中用到下面符号:⊕表示异或,≪、≫表示逐比特移位,⋘、⋙表示逐比特循环移位,‖表示级联。

1.2 算法描述

本部分将对 Rabbit 算法进行详细的描述。为了简便起见,初始化过程不使用初始向量 IV。

内部状态由 513bits 组成,其中包括 8 个 32bits 的状态变量、8 个 32bits 的计数器变量和 1bit 进位变量。$x_{j,t}$ 表示第 t 轮的状态变量,$c_{j,t}$ 表示对应的计数器变量,$\phi_{7,t}$ 表示进位变量,被存储在两轮迭代之间,初始值为 0。状态变量和计数器变量由密钥种子初始化得到。

(1)初始化。将密钥种子 $K^{[127\cdots0]}$ 分成 8 个子密钥:$k_0=K^{[15\cdots0]}$,$k_1=K^{[31\cdots16]}$,…,$k_7=K^{[127\cdots112]}$。按如下方法初始化状态变量和计数器变量:

$$x_{j,0}=\begin{cases}k_{(j+1 \bmod 8)} \| k_j, & \text{当 } j \text{ 为偶数时}\\ k_{(j+5 \bmod 8)} \| k_{(j+4 \bmod 8)}, & \text{当 } j \text{ 为奇数时}\end{cases}$$

$$c_{j,0}=\begin{cases}k_{(j+4 \bmod 8)} \| k_{(j+5 \bmod 8)}, & \text{当 } j \text{ 为偶数时}\\ k_j \| k_{(j+1 \bmod 8)}, & \text{当 } j \text{ 为奇数时}\end{cases}$$

系统将按照随后的状态更新函数和计数器更新函数迭代四轮,然后按下式再次初始化计数器变量:

到稿日期:2010-03-29 返修日期:2010-08-18 本文受国家自然科学基金(60833008)和国家 973 计划(2007CB311201)资助。
张振广(1986—),男,硕士生,主要研究方向为流密码的分析与设计,E-mail:zgzhang1986@163.com;胡予濮 教授,博士生导师;王 璐 硕士生。

$c_{j,4}=c_{j,4}\oplus x_{(j+4 \bmod 8),4}$

这样设计是为了避免攻击者用计数器变量迅速恢复密钥种子。

(2)状态更新函数。Rabbit 算法的核心部件是如下定义的状态更新函数：

当 j 为偶数时，

$$x_{j,t+1}=g_{j,t}+(g_{j-1 \bmod 8,t}\lll 16)+(g_{j-2 \bmod 8,t}\lll 16$$

当 j 为奇数时，

$$x_{j,t+1}=g_{j,t}+(g_{j-1 \bmod 8,t}\lll 8)+g_{j-2 \bmod 8,t}$$

这里所有的加法都是模 2^{32}，其中

$$g_{j,t}=(x_{j,t}+c_{j,t+1})^2\oplus((x_{j,t}+c_{j,t+1})^2\gg 32) \bmod 2^{32}。$$

(3)计数器更新函数。计数器的更新如下：

$$c_{j,t+1}=\begin{cases}c_{0,t}+a_0+\phi_{7,t} \bmod 2^{32}, & 当\ j=0\ 时\\ c_{j,t}+a_j+\phi_{j-1,t+1} \bmod 2^{32}, & 当\ j>0\ 时\end{cases}$$

这里的 $\phi_{j,t}$ 由如下方法给出：

$$\phi_{j,t+1}=\begin{cases}1, & c_{0,t}+a_0+\phi_{7,t}\geqslant 2^{32}\ 且\ j=0\\ 1, & c_{j,t}+a_j+\phi_{j-1,t+1}\geqslant 2^{32}\ 且\ j>0\\ 0, & 否则\end{cases}$$

式中，a_j 为常值，

$a_0=a_3=a_6=$0x4D34D34D

$a_1=a_4=a_7=$0xD34D34D3

$a_2=a_5=a_8=$0x34D34D34

(4)生成密钥流。从第五轮开始每次迭代后按如下方式产生一个 128bits 的密钥流子块：

$$s_t^{[15\cdots 0]}=x_{0,t}^{[15\cdots 0]}\oplus x_{5,t}^{[31\cdots 16]},s_t^{[31\cdots 16]}=x_{0,t}^{[31\cdots 16]}\oplus x_{3,t}^{[15\cdots 0]}$$

$$s_t^{[47\cdots 32]}=x_{2,t}^{[15\cdots 0]}\oplus x_{7,t}^{[31\cdots 16]},s_t^{[63\cdots 48]}=x_{2,t}^{[31\cdots 16]}\oplus x_{5,t}^{[15\cdots 0]}$$

$$s_t^{[79\cdots 64]}=x_{4,t}^{[15\cdots 0]}\oplus x_{1,t}^{[31\cdots 16]},s_t^{[95\cdots 80]}=x_{4,t}^{[31\cdots 16]}\oplus x_{7,t}^{[15\cdots 0]}$$

$$s_t^{[111\cdots 96]}=x_{6,t}^{[15\cdots 0]}\oplus x_{3,t}^{[31\cdots 16]},s_t^{[127\cdots 112]}=x_{6,t}^{[31\cdots 16]}\oplus x_{1,t}^{[15\cdots 0]}$$

式中，s_t 表示第 t 轮的密钥流子块。有关 Rabbit 密钥流生成器的更详细的描述可参考文献[1]。

2 g 函数

Rabbit 算法的一个重要部件就是 g 函数：

$$g(y)=(y^2\oplus(y^2\gg n)) \bmod 2^n$$

式中，n 是字长，该算法中 $n=32$。为了表述的方便，令 $g_n(y)=g(y)=(y^2\oplus(y^2\gg n)) \bmod 2^n$。下面给出 $g_n(y)$ 的一些性质，并做简要证明。

性质 1 对于任意的 $n\geqslant 2$，如果 2^n+1 含平方因子，则 $g_n(y)$ 不是双射。

证明：因为 2^n+1 含平方因子，不妨假设 $2^n+1=st^2$，其中 $(s,t)=1$ 且 $t\neq 1$。任取两个正整数 m_1,m_2，满足

$$m_1\leqslant 2^n t/(2^n+1),m_2\leqslant 2^n t/(2^n+1),m_1\neq m_2$$

则 $(m_1 st)^2=(m_1)^2 sst^2=(m_1)^2 s(2^n+1)$

$(m_2 st)^2=(m_2)^2 sst^2=(m_2)^2 s(2^n+1)$

而 $g_n(m_1 st)=(m_1 st)^2\oplus((m_1 st)^2\gg n) \bmod 2^n$

$=(m_1)^2 s2^n\oplus(m_1)^2 s \bmod 2^n$

$=0$

$g_n(m_2 st)=(m_2 st)^2\oplus((m_2 st)^2\gg n) \bmod 2^n$

$=(m_2)^2 s2^n\oplus(m_2)^2 s \bmod 2^n$

$=0$，

进而有 $g_n(m_1 st)=g_n(m_2 st)=0$。

因为 $m_1 st\neq m_2 st$，所以 $g_n(y)$ 不是双射。这导致不同的密钥种子可能产生相同的密钥流，从而给抵抗攻击带来困难。

性质 2 对于任意的 $n\geqslant 2$，如果 $y<\lceil 2^{n/2}\rceil$，则 $g_n(y)=y^2$，其中 $\lceil 2^{n/2}\rceil$ 表示不小于 $2^{\frac{n}{2}}$ 的整数。

证明：因为 $y<\lceil 2^{n/2}\rceil\Rightarrow y^2<2^n$，所以 $g_n(y)=y^2\oplus(y^2\gg n)=y^2$，即 $g_n(y)=y^2$。

性质 3 对于任意的偶数 $n\geqslant 2$ 都有

$$g_n(2^s-1)=\begin{cases}2^n-1, & 当\ s=n\ 时\\ \sum_{k=1}^{n-3}2^k, & 当\ s=n-1\ 时\\ \sum_{k=1}^{2s-n-1}2^k+\sum_{k=s+1}^{n-1}2^k, & 当\ n-1>s>n/2\ 时\end{cases}$$

证明：

$$g_n(2^s-1)=(2^{2s}-2^{s+1}+1)\oplus((2^{2s}-2^{s+1}+1)\gg n) \bmod 2^n$$

$$=(1+\sum_{k=s+1}^{2s-1}2^k)\oplus((1+\sum_{k=s+1}^{2s-1}2^k)\gg n) \bmod 2^n$$

当 $s=n$，$g_n(2^n-1)=1\oplus\sum_{k=1}^{n-1}2^k=\sum_{k=0}^{n-1}2^k=2^n-1$；

当 $s=n-1$，$g_n(2^s-1)=\sum_{k=1}^{n-3}2^k$；

当 $n-1>s>n/2$，$g_n(2^s-1)=\sum_{k=1}^{2s-n-1}2^k+\sum_{k=s+1}^{n-1}2^k$。

性质 4 对于任意的偶数 $n\geqslant 4$，如果

$y=\sum_{k=s}^{n-1}2^k$，则

$$g_n(y)=\begin{cases}2^{2s-n}+\sum_{k=s+1}^{n-1}2^k, & 当\ n-1>s\geqslant n/2\\ \sum_{k=s+1}^{2s-1}2^k+\sum_{k=2s+1}^{n-1}2^k, & 当\ n/2>s>1\end{cases}$$

证明：易得 $y^2=2^{2s}+2^{2n}-2^{s+n+1}$，然后有

当 $n-1>s\geqslant n/2$，

$$g_n(y)=2^{2s-n}+2^n-2^{s+1}=2^{2s-n}+\sum_{k=s+1}^{n-1}2^k$$

当 $n/2>s>1$，

$$g_n(y)=(2^{2s})\oplus(2^n-2^{s+1})=\sum_{k=s+1}^{2s-1}2^k+\sum_{k=2s+1}^{n-1}2^k$$

引理 1[8] 长为 $n$bits 的 X,Y 和 Z，对任意的 $1\leqslant l\leqslant n$，都有

$$(X\lll l)+(Y\lll l)+(Z\lll l)=((X+Y+Z)\lll l)+\delta$$

式中，加法是模 2^n，$\delta=u\times 2^l-v \bmod 2^n$，$u,v\in\{0,1,2\}$。

证明略。

3 对 Rabbit 的密钥恢复攻击

下面我们将引理应用到状态更新函数，试说明如果

$$(y'_{j,t},\ y'_{j-1 \bmod 8,t},\ y'_{j-2 \bmod 8,t})=2^k\ (y_{j,t},\ y_{j-1 \bmod 8,t},\ y_{j-2 \bmod 8,t})$$

则 $x'_{j,t+1}=(x_{j,t+1}\lll 2k)+\delta$，其中 $y_{j,t}=x_{j,t}+c_{j,t+1}$，表示第 t 轮的第 j 个状态变量与第 $t+1$ 轮的第 j 个计算器变量之和。$y'_{j,t}=x'_{j,t}+c'_{j,t+1}$，$j=0,2,4,6$。

证明：因为

$$g_{32}(y_{j,t})=y_{j,t}^2\oplus(y_{j,t}^2\gg 32) \bmod 2^{32}$$

$$g_{32}(2^k y_{j,t})=2^{2k}y_{j,t}^2\oplus(2^{2k}y_{j,t}^2\gg 32) \bmod 2^{32}$$

$$=[y_{j,t}^2\oplus(y_{j,t}^2\gg 32)]\lll 2k \bmod 2^{32}$$

g_{32} 表示该算法的字长是 32，对于下一轮的状态变量 $x_{j,t+1}$ 和 $x'_{j,t+1}$，我们有

$$x_{j,t+1}=g_{32}(y_{j,t})+g_{32}(y_{j-1 \bmod 8,t})\lll 16+g_{32}(y_{j-2 \bmod 8,t})\lll 16$$

$$=(y_{j,t}^2\oplus(y_{j,t}^2\gg32))+((y_{j-1 \bmod 8,t}^2\oplus(y_{j-1 \bmod 8,t}^2\gg32))\lll16)+((y_{j-2 \bmod 8,t}^2\oplus(y_{j-2 \bmod 8,t}^2\gg32))\lll16)$$

$$x'_{j,t+1}=g_{32}(2^k y_{j,t})+g_{32}(2^k y_{j-1 \bmod 8,t})\lll16+g_{32}(2^k y_{j-2 \bmod 8,t})\lll16$$

$$=(y_{j,t}{}^2\oplus(y_{j,t}{}^2\gg32))\lll2k+((y_{j-1 \bmod 8,t}^2\oplus(y_{j-1 \bmod 8,t}^2\gg32))\lll16)\lll2k+((y_{j-2 \bmod 8,t}^2\oplus(y_{j-2 \bmod 8,t}^2\gg32))\lll16)\lll2k$$

利用引理得到

$$x'_{j,t+1}=(x_{j,t+1}\lll2k)+\delta$$

其中加法是模 2^{32}，$\delta=u\times2^{2k}-v \bmod 2^{32}$，$u,v\in\{0,1,2\}$。

可见 $x_{j,t+1}$ 和 $x'_{j,t+1}$ 之间存在 3^2 种可能的线性关系。根据问题的对称性，同样可以得到当 $j=1,3,5,7$ 时 $x_{j,t+1}$ 和 $x'_{j,t+1}$ 之间也存在 3^2 种可能的线性关系。

下面根据上述关系进行分类。分类之前要先对状态更新函数进行预计算，时间复杂度为 $O(2^{96})$。将输入之间存在(2的幂次方)倍数关系的划分为一类，例如

$$((y_{j,t},y_{j-1 \bmod 8,t},y_{j-2 \bmod 8,t}),x_{j,t+1})$$

和

$$((2^k y_{j,t},2^k y_{j-1 \bmod 8,t},2^k y_{j-2 \bmod 8,t}),x'_{j,t+1})$$

属于同一类，这样划分可以得到 7×2^{93} 个类。现将每一类中最小(这里指输入)的一个存储起来，需要的存储空间为 $O(2^{95.81})$。当知道 $x'_{j,t+1}$ 时，就可以将它与存储的 7×2^{93} 个 $x_{j,t+1}$ 进行匹配，一定能找到多组候选值

$$((y_{j,t},y_{j-1 \bmod 8,t},y_{j-2 \bmod 8,t}),x_{j,t-1})$$

满足 $x'_{j,t+1}=(x_{j,t+1}\lll2k)+\delta$。在实际攻击过程中还将对这些候选值做进一步筛选，一旦确定了唯一的候选值也就确定了 $x'_{j,t+1}$ 对应的唯一输入

$$(y'_{j,t},y'_{j-1 \bmod 8,t},y'_{j-2 \bmod 8,t})$$

$$=(2^k y_{j,t},2^k y_{j-1 \bmod 8,t},2^k y_{j-2 \bmod 8,t})$$

下面叙述攻击过程，假设已经截获了多个密钥流子块。

攻击步骤如下：

第一阶段　恢复 $x_{j,5}$

猜测 96bits 的 $x_{1,5}^{[31\cdots16]}$，$x_{3,5}^{[15\cdots0]}$，$x_{5,5}$ 和 $x_{7,5}$。根据密钥流的生成规则由已知的密钥流子块 s_5 求出 $x_{0,5}$，$x_{2,5}$ 和 $x_{4,5}$，共得到 $x_{0,5}$，$x_{2,5}$，$x_{4,5}$，$x_{5,5}$ 和 $x_{7,5}$，将它们分别与存储的 7×2^{93} 个 $x_{j,t+1}$ 进行匹配，得到多组候选值

$$((y_{0,4},y_{7,4},y_{6,4}),x_{0,5})$$

$$((y_{2,4},y_{1,4},y_{0,4}),x_{2,5})$$

$$((y_{4,4},y_{3,4},y_{2,4}),x_{4,5})$$

$$((y_{5,4},y_{4,4},y_{3,4}),x_{5,5})$$

$$((y_{7,4},y_{6,4},y_{5,4}),x_{7,5})$$

这些组候选值中相同变量的值必须是相等的。抛弃那些不等的，得到一组

$$(y_{0,4},y_{1,4},y_{2,4},y_{3,4},y_{4,4},y_{5,4},y_{6,4},y_{7,4})$$

由它计算出 32bits 密钥流 $s_5^{[127\cdots96]'}$，通过核对 $s_5^{[127\cdots96]}$ 来确定正确的猜测，从而恢复 $y_{j,4}$，$j=0,\cdots,7$，进而也得到了 $x_{j,5}$，$j=0,\cdots,7$。本阶段时间复杂度为 $O(2^{96})$。

第二阶段　恢复 $c_{j,6}$

用第一阶段的相同方法由密钥流子块 s_6 恢复 $y_{j,5}$，$j=0,\cdots,7$。由第一阶段得到的 $x_{j,5}$ 根据 $y_{j,5}=x_{j,5}+c_{j,6}$ 求出计数器变量 $c_{j,6}$，$j=0,\cdots,7$。本阶段时间复杂度为 $O(2^{96})$。

第三阶段　恢复 $K^{[127\cdots0]}$

第一步　猜测 2bits 的 $\phi_{7,5}$ 和 $\phi_{7,4}$，反推得到 $c_{j,5}$ 和 $c_{j,4}$，这里的 $c_{j,4}$ 是经历再次初始化后的。由第一阶段得到的 $y_{j,4}$，再根据 $y_{j,4}=x_{j,4}+c_{j,5}$ 得到 $x_{j,4}$。根据 $c_{j,4}=c_{j,4}\oplus x_{(j+4 \bmod 8),4}$ 得到未经历再次初始化的 $c_{j,4}$；

第二步　猜测 3bits 的 $\phi_{7,3}$，$\phi_{7,2}$ 和 $\phi_{7,1}$，反推得到 $c_{j,3}$，$c_{j,2}$ 和 $c_{j,1}$，又由已知的 $\phi_{7,0}=0$ 得到 $c_{j,0}$。根据初始化规则知道 $c_{j,0}$，$j=0,\cdots,7$ 是高度相关的，通过此相关性来确定正确的猜测，从而恢复了 $K^{[127\cdots0]}$。本阶段时间复杂度为 $O(2^5)$。

整个过程需要存储空间为 $O(2^{95.81})$，预计算复杂度为 $O(2^{96})$，时间复杂度为 $O(2^{96})+O(2^{96})+O(2^5)\approx O(2^{97})$。

下面将结果与文献[8]中的密钥恢复攻击相比，如图 1 所示。

攻击算法	存储空间	预计算复杂度	时间复杂度
本文	$O(2^{95.81})$	$O(2^{96})$	$O(2^{97})$
文献[8]	$O(2^{32})$	$O(2^{32})$	$O(2^{97.5})$

图 1

可见存储空间和预计算复杂度稍大，但时间复杂度方面优于文献[8]。

结束语　通过对 Rabbit 密钥流生成器的安全性分析，发现了其核心部件 g 函数的多个弱点。比如当 2^n+1 含平方因子时 g 函数不是双射，即不同的输入可能产生相同的输出。还有当 $y<\lceil 2^{n/2}\rceil$ 时，$g(y)=y^2$ 等。这些弱点直接降低了算法的安全性。并且根据状态更新函数的性质提出了一种基于猜测确定的密钥恢复攻击。攻击过程分为 3 个阶段：依次恢复状态变量 $x_{j,5}$、计数器变量 $c_{j,6}$ 和密钥种子。整个过程的预计算复杂度为 $O(2^{96})$，时间复杂度为 $O(2^{97})$，所需存储空间为 $O(2^{95.81})$，与文献[8]中的密钥恢复攻击相比，预计算复杂度高于它的 $O(2^{32})$，时间复杂度低于它的 $O(2^{97.5})$，需要存储空间多于它的 $O(2^{32})$，可见各有利弊。当然，这种攻击方法仍然还不至于对 Rabbit 的安全性造成现实的威胁。如何进一步挖掘 Rabbit 密钥流生成器的设计弱点以实施更有效的攻击，还需进一步研究。本文的研究结果说明 Rabbit 算法还有漏洞，所以在使用之前还需要做进一步的安全性分析。

参 考 文 献

[1] Boesgaard M, Vesterager M, Pedersen T, et al. Rabbit: A new high-performance stream cipher [C]// Johansson T, ed. FSE 2003. LNCS, vol. 2887, 2003: 307-329

[2] Cryptico A/S. Algebraic analysis of rabbit [DB/OL]. White paper. http://www.crytico.com, 2003

[3] CrypticoA/S. Analysisofthekeysetupfunctionsinrabbit[DB/OL]. White paper. http://www.crytico.com, 2003

[4] Cryptico A/S. Periodic properties of rabbit [DB/OL]. White paper. http://www.crytico.com, 2003

[5] Cryptico A/S. Second degree approximations of the g-function [DB/OL]. White paper. http://www.crytico.com, 2003

[6] Cryptico A/S. Security analysis of the IV-setup for rabbit [DB/OL]. White paper. http://www.crytico.com, 2003

[7] Aumasson J-P. On a bias of Rabbit [DB/OL]. http://www.ecrypt.eu.org/stream/, 2009

[8] Lu Yi, Wang Huaxiong, Ling San. Cryptanalysis of Rabbit [C]// ISC. 2008, LNCS 5222, 2008: 204-214