doi:10.3969/j.issn.1000-436x.2012.11.024

基于 E_0 算法的猜测决定攻击

詹英杰，丁林，关杰
(信息工程大学 电子技术学院，河南 郑州 450004)

摘 要：对短距离无线蓝牙技术中使用的 E_0 序列密码算法进行了猜测决定攻击，攻击中利用线性逼近的方法做出了一个巧妙的攻击假设，降低了攻击所需的猜测量，并且通过一个检验方程降低了候选状态的数量，攻击的计算复杂度为 $O(2^{76})$，需要约 988bit 密钥流,属于短密钥流攻击。相对于长密钥流攻击，短密钥流攻击所需的密钥流不超过 2 745bit，对 E_0 的安全性更具威胁。与目前已有的针对 E_0 的短密钥流攻击相比，所提出猜测决定的攻击结果是最好的。

关键词：密码分析；E_0 算法；猜测决定攻击；检验方程
中图分类号：TN918.1 **文献标识码**：A **文章编号**：1000-436X(2012)11-0185-06

Guess and determine attack on E_0 algorithm

ZHAN Ying-jie, DING Lin, GUAN Jie
(Electronic Technology Institute, Information Engineering University, Zhengzhou 450004, China)

Abstract: A guess and determine attack on E_0 was presented, the stream cipher that was used in the short-range wireless technology Bluetooth. In the attack, an ingenious assumption by linear approximation to diminish the number of values that have to be guessed was made. Futher more, a test equation to reduce the number of the guessed candidates was used. The guess and determine attack on E_0 with time complexity of $O(2^{76})$ required about 988 keystream bit. Compared with long keystream attacks, short keystream attacks were more threatening to E_0 cipher which require less than 2 745 bit. This attack which belongs to short keystream attack was the fastest state recovery attack on E_0 so far compared with any other existed short keystream attacks.

Key words: cryptanalysis; E_0 algorithm; guess and determine attack; test equation

1 引言

蓝牙[1]（Bluetooth）是一种低成本、低功率、短距离的无线通信技术标准，包括硬件规范和软件体系结构，其目的是取代现有的 PC 机、打印机、传真机和移动电话等设备的有线接口。蓝牙在财务处理、汽车应用、工业控制等诸多领域中有着广泛的应用前景。在蓝牙网络规范中，只是在无线链路上提供蓝牙安全机制。蓝牙基带标准推荐了 4 个认证函数 E_0、E_1、E_2 和 E_3，它们是在蓝牙设备中产生各种密钥的函数，其中，E_0 用于数据加密；E_1 用于设备认证；E_2 分成 E_{21} 和 E_{22} 2 部分，其中，E_{21} 函数用于生成链路密钥；E_{22} 函数用于生成初始密钥，E_3 用于产生加密密钥。本文将具体分析蓝牙链路层所采用的加密算法 E_0 的安全性。

E_0 算法是一个二进制序列密码，使用 128bit 的加密密钥 K_c、48bit 的主设备蓝牙地址 BD_ADDR 和 26bit 的主设备时钟 Clock 作为 E_0 的输入参数，

收稿日期：2011-08-22；**修回日期**：2012-02-08
基金项目：国家自然科学基金资助项目（61202491）；全军军事学研究生课题基金资助项目(2010JY0263-149)
Foundation Items: The National Natural Science Foundation of China(61202491); The Military Science Graduate Research Foundation of Army(2010JY0263-149)

产生二进制密钥流，该密钥流与数据流进行异或运算后发送到空中接口，完成加密过程。

由于 E_0 具有广泛的应用背景并被写入蓝牙网络规范，对它的安全性分析一直都是一个热点问题。蓝牙加密系统使用频繁的自同步机制来保证同步，这使得在实际应用中 E_0 的一个初始状态最多只能产生 2 745bit 的密钥流，也就是说，要产生多于 2 745bit 的密钥流，需要更换内部状态后再使用 E_0 加密产生。根据所需密钥流长度是否多于 2 745bit，通常将针对 E_0 的攻击分为 2 种类型：长密钥流攻击和短密钥流攻击。针对 E_0 的长密钥流攻击主要有代数攻击[2,3]、快速相关攻击[4]和条件相关攻击[5]，其中，最好的长密钥流攻击是由 Yi Lu 等在 CRYPTO 2005 年会上提出的针对 E_0 的条件相关攻击[5]，需要 $2^{28.4}$bit 的密钥流，$O(2^{38})$的预计算量，计算复杂度和存储复杂度分别为 $O(2^{38})$和 $O(2^{33})$。虽然长密钥流攻击的计算复杂度已经足够低了，但由于所需的密钥流多于 2 745bit，大多只具有理论意义，不能对 E_0 构成实际性的威胁。

相对于长密钥流攻击，短密钥流攻击所需的密钥流不超过 2 745bit，因而对 E_0 的安全性更具威胁，遗憾的是，目前已有的短密钥流攻击的计算复杂度都较高。在文献[6]中，Bleichenbacher 提出了一个针对 E_0 的状态恢复攻击，需要 132bit 密钥流，计算复杂度为 $O(2^{100})$。Fluhrer 和 Lucks[7]使用优化的回溯法恢复了 E_0 的初始密钥，需要 132bit 密钥流，计算复杂度降为 $O(2^{84})$。在文献[8]中，Krause 将二元决策图（binary decision diagrams）的思想引入到流密码分析中，对 E_0 进行了分析，需要 128bit 密钥流，所需计算复杂度和存储复杂度分别为 $O(2^{81})$和 $O(2^{77})$。Levy 和 Wool 在文献[9]中对文献[6]、文献[7]的分析方法进行了深入分析，恢复全部 132bit 内部状态需要 128bit 密钥流和 $O(2^{86})$的计算量。Yaniv 和 Avishai[10]对 Krause 的攻击方法进行了优化，仍需要 128bit 密钥流，所需计算复杂度和存储复杂度分别为 $O(2^{87})$和 $O(2^{23})$。目前最好的短密钥流攻击是由郭峰和庄弈琪[11]对 E_0 的状态恢复攻击，需要 1 146bit 密钥流，计算复杂度为 $O(2^{83})$。

猜测决定攻击（guess and determine attack）是一种常见的流密码分析方法，从原理上讲是一种“分别征服（divide and conquer）”的方法，其基本思想是先猜测一部分内状态，运用输出密钥流和内部状态之间的关系（密钥流生成器的输出变换）以及内部状态之间的关系（内部状态的刷新变换）来决定其他的内部状态，进而恢复出全部的内部状态。一般而言，猜测决定攻击主要包括猜测过程和决定过程 2 个方面，计算复杂度主要由猜测量决定；事实上，有时仅仅通过猜测和决定会使得猜测决定攻击的计算复杂度很高，此时做出适当的攻击假设，可以有效地降低猜测量，但代价是需增加一定的数据量以使得攻击假设能够成立，此时猜测决定攻击的计算复杂度由猜测量和数据量 2 个方面决定，是两者的综合。

本文对 E_0 序列密码算法进行了猜测决定攻击，攻击中利用线性逼近的方法做出了一个巧妙的假设，降低了攻击所需的猜测量，并用一个检验方程来对候选状态进行筛选，攻击的计算复杂度为 $O(2^{76})$，需要约 988bit 密钥流。与目前已有的针对 E_0 的短密钥流攻击相比，本文的攻击结果是最好的。

2 E_0 序列密码

序列密码通常包括 2 部分：初始化过程和密钥流生成过程。因本文对 E_0 的猜测决定攻击只涉及 E_0 的密钥流生成过程，故本文不再详细介绍 E_0 的初始化过程。

E_0 的密钥流生成器的结构如图 1 所示。图中 E_0 的密钥流生成器主要由 2 部分组成：4 个线性移位寄存器和 1 个有限状态机（FSM, finite state machine）。4 个线性移位寄存器的长度分别为 25、31、33 和 39（共 128bit），其反馈多项式都是本原多项式，分别为

$$f_1(x) = x^{25} + x^{20} + x^{12} + x^8 + 1$$

$$f_2(x) = x^{31} + x^{24} + x^{16} + x^{12} + 1$$

$$f_3(x) = x^{33} + x^{28} + x^{24} + x^4 + 1$$

$$f_4(x) = x^{39} + x^{36} + x^{28} + x^4 + 1$$

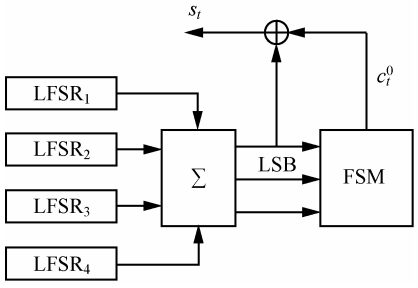

图 1 E_0 的密钥流生成器

FSM 中包含有 4bit 的内部记忆单元，记为 $C_t=(c_t,c_{t-1})$，其中，t 表示时钟，$c_t=(c_t^1,c_t^0)$，c_t^0 和 c_t^1 都表示 1bit 记忆单元。在 FSM 的状态刷新变换中还用到 2 个中间比特，表示为 $s_t=(s_t^1,s_t^0)$。$\{z_t\}_{t\geq 0}$ 表示 E_0 的输出密钥流序列。则 FSM 的状态刷新变换和 E_0 的密钥流输出变换可表示为

$$(s_{t+1}^1,s_{t+1}^0)=\left\lfloor\frac{x_t^1+x_t^2+x_t^3+x_t^4+2c_t^1+c_t^0}{2}\right\rfloor \tag{1}$$

$$c_{t+1}^0=s_{t+1}^0\oplus c_t^0\oplus c_{t-1}^0\oplus c_{t-1}^1 \tag{2}$$

$$c_{t+1}^1=s_{t+1}^1\oplus c_t^1\oplus c_{t-1}^0 \tag{3}$$

$$z_t=x_t^1\oplus x_t^2\oplus x_t^3\oplus x_t^4\oplus c_t^0 \tag{4}$$

除了以上 4 个等式，N.Rajesh Pillai 等在文献[12]中还给出了一个由 4 个连续时刻的内部状态建立的方程，该方程不依赖于记忆位 C_t。在进行猜测决定攻击时，可以用这个方程对候选状态进行筛选，以达到减少候选状态的数量。方程如下(证明过程见附录)

$$\begin{aligned}0=&\,z_t\oplus z_{t+1}\oplus z_{t+2}\oplus z_{t+3}\\&\oplus\prod_{t+1}^1(z_{t+1}z_t\oplus z_{t+1}z_{t+2}\oplus z_{t+1}z_{t+3}\oplus z_t\oplus z_{t+1}\oplus z_{t+2}\\&\oplus z_{t+3})\oplus\prod_{t+1}^2(z_t\oplus z_{t+1}\oplus z_{t+2}\oplus z_{t+3})\oplus\prod_{t+1}^3 z_{t+1}\\&\oplus\prod_{t+1}^4\oplus\prod_t^1\oplus\prod_{t+1}^1\prod_t^1(1\oplus z_{t+1})\oplus\prod_{t+1}^2\prod_t^1\\&\oplus\prod_{t+2}^1 z_{t+2}\oplus\prod_{t+1}^1\prod_{t+2}^1 z_{t+2}(1\oplus z_{t+1})\oplus\prod_{t+1}^2\prod_{t+2}^1 z_{t+2}\\&\oplus\prod_{t+2}^2\oplus\prod_{t+2}^2\prod_{t+1}^1(1\oplus z_{t+1})\oplus\prod_{t+1}^2\prod_{t+2}^2\\&\oplus\prod_{t+3}^1\oplus\prod_{t+3}^1\prod_{t+1}^1(1\oplus z_{t+1})\oplus\prod_{t+3}^1\prod_{t+1}^2\end{aligned} \tag{5}$$

其中，

$$\prod_t^1=x_t^1\oplus x_t^2\oplus x_t^3\oplus x_t^4$$

$$\prod_t^2=x_t^1x_t^2\oplus x_t^1x_t^3\oplus x_t^1x_t^4\oplus x_t^2x_t^3\oplus x_t^2x_t^4\oplus x_t^3x_t^4$$

$$\prod_t^3=x_t^1x_t^2x_t^3\oplus x_t^1x_t^2x_t^4\oplus x_t^1x_t^3x_t^4\oplus x_t^2x_t^3x_t^4$$

$$\prod_t^4=x_t^1x_t^2x_t^3x_t^4$$

$$\prod_{t+1}^1=x_{t+1}^1\oplus x_{t+1}^2\oplus x_{t+1}^3\oplus x_{t+1}^4$$

$$\begin{aligned}\prod_{t+1}^2=&\,x_{t+1}^1x_{t+1}^2\oplus x_{t+1}^1x_{t+1}^3\oplus x_{t+1}^1x_{t+1}^4\\&\oplus x_{t+1}^2x_{t+1}^3\oplus x_{t+1}^2x_{t+1}^4\oplus x_{t+1}^3x_{t+1}^4\end{aligned}$$

$$\begin{aligned}\prod_{t+1}^3=&\,x_{t+1}^1x_{t+1}^2x_{t+1}^3\oplus x_{t+1}^1x_{t+1}^2x_{t+1}^4\\&\oplus x_{t+1}^1x_{t+1}^3x_{t+1}^4\oplus x_{t+1}^2x_{t+1}^3x_{t+1}^4\end{aligned}$$

$$\prod_{t+1}^4=x_{t+1}^1x_{t+1}^2x_{t+1}^3x_{t+1}^4$$

$$\prod_{t+2}^1=x_{t+2}^1\oplus x_{t+2}^2\oplus x_{t+2}^3\oplus x_{t+2}^4$$

$$\begin{aligned}\prod_{t+2}^2=&\,x_{t+2}^1x_{t+2}^2\oplus x_{t+2}^1x_{t+2}^3\oplus x_{t+2}^1x_{t+2}^4\\&\oplus x_{t+2}^2x_{t+2}^3\oplus x_{t+2}^2x_{t+2}^4\oplus x_{t+2}^3x_{t+2}^4\end{aligned}$$

$$\prod_{t+3}^1=x_{t+3}^1\oplus x_{t+3}^2\oplus x_{t+3}^3\oplus x_{t+3}^4$$

根据以上描述可以看出，E_0 的内部状态规模为 132bit，其中，4 个线性移位寄存器共 128bit 和 FSM 中的内部记忆单元共 4bit，E_0 的初始化使用 128bit 的加密密钥 K_c、48bit 的主设备蓝牙地址 BD_ADDR 和 26bit 的主设备时钟 Clock 作为输入参数，经过 239 个时钟，完成 E_0 的初始化过程，进入密钥流生成过程，产生密钥流序列用于对数据流的加密[1]。

3　攻击方法

在对 E_0 进行猜测决定攻击之前，首先对 E_0 中使用的非线性变换进行线性逼近，以得到想要的攻击假设。从第 2 节中对 E_0 的描述中可以看出，E_0 的密钥流生成器中唯一的非线性函数就是变换(1')，即刷新变换

$$(s_{t+1}^0,s_{t+1}^1)=\left\lfloor\frac{x_t^1+x_t^2+x_t^3+x_t^4+2\cdot c_t^1+c_t^0}{2}\right\rfloor \tag{1'}$$

该变换是一个 6bit 输入、2bit 输出的非线性函数。以下将通过线性逼近的方法考察变换(1')中输入与输出之间的线性相关性。

设 $w=(w_0,\cdots,w_5)$ 是一个 6bit 非零系数，$u=(u_0,u_1)$ 是一个 2bit 非零系数，令 $x=(x_0,\cdots,x_5)=(x_t^1,x_t^2,x_t^3,x_t^4,c_t^1,c_t^0)$，$y=(y_0,y_1)=(s_{t+1}^0,s_{t+1}^1)$，则变换(1)中输入与输出之间的线性相关性可用如下概率来描述。

$$p(w,u)=\frac{\#\{x\mid wx\oplus uy=0\}}{64}$$

其中，$wx=\bigoplus_{i=0}^{5}w_ix_i$ 和 $uy=\bigoplus_{j=0}^{1}u_jy_j$ 表示点积。通过编程穷举的方法，得到了概率 $p(w,u)$ 的具体分布情况，结果表明概率最大为

$$p(000\,010,01)=\frac{52}{64}$$

这意味着变换(1)中输入与输出之间存在着线性逼近式 $s_{t+1}^1\oplus c_t^1=0$，该线性逼近式成立的概率为 $52/64$。利用该线性逼近式，可以做出一个攻击假设，降低攻击所需的猜测量。

假设攻击者已经得到长度为 N 的密钥流

$\{z_t\}, t=1,\cdots,N$，其中，$N \leqslant 2\,745$。为降低猜测量，做出如下假设

假设：$s^1_{t+1+i} \oplus c^1_{t+i} = 0$，$i = 0,1,\cdots,32$

式(1)中，如果 FSM$(x^1_{t+i}, x^2_{t+i}, x^3_{t+i}, x^4_{t+i}, c^1_{t+i}, c^0_{t+i})$的输入比特是均匀分布的，则假设成立的概率为

$$\left(\frac{52}{64}\right)^{33} \approx 2^{-9.89}$$

因此，要找到满足上面假设的内部状态，需要尝试 $2^{9.89}$ 个 t 的取值。如果假设是错误的，则下面所做的猜测就无法得到算法正确的内部状态，所以就无法得到正确的密钥流序列。

当假设满足时，由式(3)可以得到

$$c^1_{t+1+i} = c^0_{t-1+i},\ i = 0,1,\cdots,32 \qquad (3')$$

本文针对 E_0 的攻击可以分为 4 个步骤。

步骤 1 在时刻 t，猜测 $c^0_{t-1}, c^0_t, c^0_{t+1}, x^1_t \oplus x^2_t, c^1_{t-1}, c^1_t$ 这 6 个比特，这样就可以决定以下的内部状态比特

Step 0.a $\quad c^0_{t-1} \xrightarrow{(3')} c^1_{t+1}$

Step 0.b $\quad c^1_t \xrightarrow{\text{假设}} s^1_{t+1}$

Step 0.c $\quad c^0_{t+1}, c^0_t, c^0_{t-1}, c^1_{t-1} \xrightarrow{(2)} s^0_{t+1}$

Step 0.d $\quad z_t, x^1_t \oplus x^2_t, c^0_t \xrightarrow{(4)} x^3_t \oplus x^4_t$

令 $a = x^1_t \oplus x^2_t$，$b = x^3_t \oplus x^4_t$，由此，式(1)可以表示为

$$(s^0_{t+1}, s^1_{t+1}) = \left\lfloor \frac{x^1_t + (x^1_t \oplus a) + x^3_t + (x^3_t \oplus b) + 2 \cdot c^1_t + c^0_t}{2} \right\rfloor \qquad (6)$$

其中，x^1_t 和 x^3_t 为未知变量，很容易证明此时方程解个数的期望为 $\frac{104}{64}$。

在时刻 $t+i(i=1,2)$，通过猜测 c^0_{t+1+i} 和 $x^1_{t+i} \oplus x^2_{t+i}$，可以得到以下的内部状态比特：

Step i.a $\quad c^0_{t-1+i} \xrightarrow{(3')} c^1_{t+1+i}$

Step i.b $\quad c^1_{t+i} \xrightarrow{\text{假设}} s^1_{t+1+i}$

Step i.c $\quad c^0_{t+1+i}, c^0_{t+i}, c^0_{t-1+i}, c^1_{t-1+i} \xrightarrow{(2)} s^0_{t+1+i}$

Step i.d $\quad z_{t+i}, x^1_{t+i} \oplus x^2_{t+i}, c^0_{t+i} \xrightarrow{(4)} x^3_{t+i} \oplus x^4_{t+i}$

同样将式(6)中的 t 替换成 $t+i$，可以得到关于 x^1_{t+i} 和 x^3_{t+i} 的方程，解个数的期望为 $\frac{104}{64}$。

在时刻 $t+i(i=3,\cdots,24)$ 时，先采用相同的方法确定内部状态。然后将 $t+i$ 以及前 3 个时刻所确定的 $(x^1_t, x^2_t, x^3_t, x^4_t)$ 代入式(5)进行检验，通过检验的概率约为 $\frac{11}{13}$。

步骤 1 中共猜测了 6+2×24=54 个比特，所以候选状态的个数为

$$2^6 \times \frac{104}{64} \times \left(2^2 \times \frac{104}{64}\right)^2 \times \left(2^2 \times \frac{104}{64} \times \frac{11}{13}\right)^{22} \approx 2^{66.20}$$

此时已经恢复了 $x^1_t, \cdots, x^1_{t+24}$，可以通过 LFSR$_1$ 的反馈多项式确定 x^1_{t+i} $(i \geqslant 25)$ 的值。

步骤 2 在时刻 $t+i(i=25,\cdots,30)$，需要猜测 c^0_{t+1+i} 的值来确定以下的比特：

Step i.a $\quad c^0_{t-1+i} \xrightarrow{(3')} c^1_{t+1+i}$

Step i.b $\quad c^1_{t+i} \xrightarrow{\text{假设}} s^1_{t+1+i}$

Step i.c $\quad c^0_{t+1+i}, c^0_{t+i}, c^0_{t-1+i}, c^1_{t-1+i} \xrightarrow{(2)} s^0_{t+1+i}$

Step i.d $\quad z_{t+i}, x^1_{t+i}, c^0_{t+i} \xrightarrow{(4)} x^2_{t+i} \oplus x^3_{t+i} \oplus x^4_{t+i}$

令 $c = x^1_{t+i} \oplus c^0_{t+i} \oplus z_{t+i}$，可以得到非线性方程

$$(s^0_{t+1+i}, s^1_{t+1+i}) = \left\lfloor \frac{x^1_{t+i} + x^2_{t+i} + x^3_{t+i} + (x^2_{t+i} \oplus x^3_{t+i} \oplus c) + 2 \cdot c^1_{t+i} + c^0_{t+i}}{2} \right\rfloor \qquad (7)$$

其中，x^2_{t+i} 和 x^3_{t+i} 为未知，同样方程解个数的期望也是 $\frac{104}{64}$，代入式(5)检验，通过概率为 $\frac{124}{169}$。步骤 2 过后，候选状态的个数为

$$2^{66.20} \times \left(2 \times \frac{104}{64} \times \frac{124}{169}\right)^6 \approx 2^{70.78}$$

步骤 2 完成后，已经恢复出 $x^2_t, \cdots, x^2_{t+30}$，因此可以根据 LFSR$_2$ 的反馈多项式决定 x^2_{t+i} $(i \geqslant 31)$ 的取值。

步骤 3 在时刻 $t+i(i=31,32)$，此前已经猜测了 c^0_{t+31} 且决定了 c^1_{t+31}，因此在本步骤中不需要猜测任何比特，就可以决定以下内部比特的取值

Step i.a $\quad c^0_{t-1+i} \xrightarrow{(3')} c^1_{t+1+i}$

Step i.b $\quad c^1_{t+i} \xrightarrow{\text{假设}} s^1_{t+1+i}$

Step i.c $\quad z_{t+i}, x^1_{t+i}, x^2_{t+i}, c^0_{t+i} \xrightarrow{(4)} x^3_{t+i} \oplus x^4_{t+i}$

此时，令 $d = x^1_{t+i} \oplus x^2_{t+i} \oplus c^0_{t+i} \oplus z_{t+i}$，即得到

$$(s^0_{t+1+i}, s^1_{t+1+i}) = \left\lfloor \frac{x^1_{t+i} + x^2_{t+i} + x^3_{t+i} + (x^3_{t+i} \oplus d) + 2 \cdot c^1_{t+i} + c^0_{t+i}}{2} \right\rfloor \qquad (8)$$

式(8)中的未知变量为 x^3_{t+i} 和 s^0_{t+1+i}，此时方程解个数的期望为 $\frac{52}{64}$，代入式(5)进行检验，通过的概率为 $\frac{11}{16}$。并且由

表 1　　**对 E_0 的猜测决定攻击过程**

步骤	时刻 $t+i$	假设	猜测比特	决定比特	候选
1	$i=0,1,\cdots,24$	$s^1_{t+1+i}\oplus c^1_{t+i}=0$	$c^0_{t-1},c^0_t,\cdots,c^0_{t+25},c^1_{t-1},c^1_t,\ x^1_t\oplus x^2_t,x^1_{t+1}\oplus x^2_{t+1},\cdots,x^1_{t+24}\oplus x^2_{t+24}$	$x^1_{t+i},x^2_{t+i},x^3_{t+i},x^4_{t+i}$	$2^{66.20}$
2	$i=25,\cdots,30$	$s^1_{t+1+i}\oplus c^1_{t+i}=0$	$c^0_{t+26},\cdots,c^0_{t+31}$	$x^2_{t+i},x^3_{t+i},x^4_{t+i}$	$2^{70.78}$
3	$i=31,32$	$s^1_{t+1+i}\oplus c^1_{t+i}=0$	…	x^3_{t+i},x^4_{t+i}	$2^{69.10}$
4	$i=33,\cdots,38$	…	…	x^4_{t+i}	$2^{65.86}$

Step i.d　$s^0_{t+1+i},c^0_{t+i},c^0_{t-1+i},c^1_{t-1+i}\xrightarrow{(2)}c^0_{t+1+i}$

可以决定 c^0_{t+1+i} 的取值，将 x^3_{t+i} 代入式(4)中可以求出 x^4_{t+i} $(i=31,32)$。

在步骤 3 中，恢复了 $x^3_t,\cdots,x^3_{t+32}$ 的取值，当 $i\geqslant 33$ 时，根据 $LFSR_3$ 的反馈多项式就可以决定 x^3_{t+i} 的值。此时候选状态的个数为

$$2^{70.78}\times\left(\frac{52}{64}\times\frac{11}{16}\right)^2\approx 2^{69.10}$$

步骤 4　因为已经决定了 c^0_{t+33} 和 c^1_{t+33} 的取值，所以在时刻 $t+i(i=33,\cdots,38)$ 时，可以直接确定以下比特

Step i.a　$z_{t+i},x^1_{t+i},x^2_{t+i},x^3_{t+i},c^0_{t+i}\xrightarrow{(4)}x^4_{t+i}$

Step i.b　$x^1_{t+i},x^2_{t+i},x^3_{t+i},x^4_{t+i},c^1_{t+i},c^0_{t+i}\xrightarrow{(1)}s^0_{t+1+i},s^1_{t+1+i}$

Step i.c　$s^0_{t+1+i},c^0_{t+i},c^0_{t-1+i},c^1_{t-1+i}\xrightarrow{(2)}c^0_{t+1+i}$

Step i.d　$s^1_{t+1+i},c^1_{t+i},c^0_{t-1+i}\xrightarrow{(3)}c^1_{t+1+i}$

因此可以获得 x^4_{t+i} $(i=33,\cdots,38)$ 的取值，代入式(5)检验，通过的概率为 $\frac{11}{16}$。候选状态的个数为

$$2^{69.10}\times\left(\frac{11}{16}\right)^6\approx 2^{65.86}$$

至此，已经恢复出 E_0 所有 132bit 内部比特。攻击的过程总结如表 1 所示。

4　计算复杂度分析

在攻击开始时所作的假设条件为

$$s^1_{t+1+i}\oplus c^1_{t+i}=0\quad(i=0,1,\cdots,32)$$

如果 $FSM(x^1_{t+i},x^2_{t+i},x^3_{t+i},x^4_{t+i},c^1_{t+i},c^0_{t+i})$ 的输入比特是均匀分布的，则假设成立的概率为 $2^{-9.89}$。本文进行了 100 000 次实验，每次实验随机选取 LFSR 与 FSM 的内部状态比特，结果显示满足假设的实验次数为 113，则假设正确的概率为 $113/100\,000\approx 2^{-9.79}$，非常接近 $2^{-9.89}$，证明本文中的攻击方法是正确的。

在对 E_0 的猜测决定攻击中，攻击者需要尝试大约 $2^{9.89}$ 个时刻 t 才能找到满足攻击假设的时刻；同时，在攻击过程中，4 个阶段共需要 39bit 密钥流 $(z_t,\cdots,z_{t+38})$。因此，对 E_0 的猜测决定攻击所需的数据量约为 $2^{9.89}+39\approx 988$ bit 密钥流。由于该密钥流长度未超过 2 745bit，因而，本文对 E_0 的猜测决定攻击是一种实际可行的短密钥流攻击。

在完成 4 个攻击步骤后所剩下的候选状态中，用它们来生成密钥流序列，与正确的密钥流进行比较来检验其正确性。如果所有候选状态均不正确，则需要增加 t 的取值来重新开展攻击。

本文中针对 E_0 的猜测决定攻击计算复杂度为

$$2^{65.86}\times 2^{9.89}\approx 2^{76}$$

因为每个候选状态是依次进行尝试的，所以攻击的存储复杂度可以忽略不计。

本文的攻击结果与目前已有的针对 E_0 的短密钥流攻击结果对比如表 2 所示。从表中可以看出，本文的攻击明显优于目前已有的所有短密钥流攻击。

表 2　　**E_0 的短密钥流攻击结果**

攻击方法	所需密钥流比特	计算复杂度	存储复杂度
Bleichenbacher[6]	132	2^{100}	—
Fluhrer-Lucks[7]	132	2^{84}	—
Levy-Wool[8]	128	2^{86}	—
Yaniv-Wool[9]	128	2^{87}	2^{23}
Krause[10]	128	2^{81}	2^{77}
郭锋，庄奕琪[11]	1146	2^{83}	—
本文	988	2^{76}	—

5　结束语

序列密码算法 E_0 是蓝牙链路层所采用的加密算法，其密钥规模为 128bit，内部状态规模为 132bit。本文对 E_0 密码算法进行了猜测决定攻击，攻击中利用线性逼近的方法作出了一个巧妙的假设，降低了攻击所需的猜测量，并用一个不依赖于记忆位的检验方程降低了候选状态的数量，攻击的计算复杂度为 $O(2^{76})$，需要约 988bit 密钥流。与目前已有的针对 E_0 的短密钥流攻击相比，本文的攻击效果是最好

的。对猜测决定攻击而言，本文所使用的利用线性逼近的方法做攻击假设的思想具有一定的普适性，可以考虑将其应用于其他的序列密码算法。

附录：式(5)证明[12,13]

式(5)最先由 F Armknecht 等在文献[13]中提出，但 N Rajesh Pillai 等在文献[12]中指出了其存在的细微错误，并给出了正确的方程，但没有给出具体的证明过程。这一节将给出详细的证明过程。

由 $(s_{t+1}^1, s_{t+1}^0) = \left\lfloor \dfrac{x_t^1 + x_t^2 + x_t^3 + x_t^4 + 2c_t^1 + c_t^0}{2} \right\rfloor$，定义 2 个布尔函数:

$$s_{t+1}^i = f_i(x_t^1, x_t^2, x_t^3, x_t^4, c_t^1, c_t^0), i \in \{0,1\} \tag{9}$$

通过观察 f_0, f_1 的真值表，可以得到它们的代数标准形为

$$f_1 = \Pi_4(t) \oplus \Pi_3(t)c_t^0 \oplus \Pi_2(t)c_t^1 \oplus \Pi_1(t)c_t^0 c_t^1 \tag{10}$$

$$f_0 = \Pi_2(t) \oplus \Pi_1(t)c_t^0 \oplus c_t^1 \tag{11}$$

将式(10)、式(11)代入式(2)、式(3)可以得

$$c_{t+1}^0 = \Pi_2(t) \oplus \Pi_1(t)c_t^0 \oplus c_t^1 \oplus c_{t-1}^1 \oplus c_t^0 \oplus c_{t-1}^0 \tag{12}$$

$$c_{t+1}^1 = \Pi_4(t) \oplus \Pi_3(t)c_t^0 \oplus \Pi_2(t)c_t^1 \oplus \Pi_1(t)c_t^0 c_t^1 \oplus c_t^1 \oplus c_{t-1}^0 \tag{13}$$

令

$$a(t) = \Pi_4(t) \oplus \Pi_3(t)c_t^0 \oplus c_{t-1}^0$$

$$b(t) = \Pi_2(t) \oplus \Pi_1(t)c_t^0 \oplus 1$$

则式(12)和式(13)可以表示为

$$c_{t+1}^0 = b(t) \oplus 1 \oplus c_{t-1}^0 \oplus c_t^0 \oplus c_t^1 \oplus c_{t-1}^1 \tag{14}$$

$$c_{t+1}^1 = a(t) \oplus b(t)c_t^1 \tag{15}$$

将式(15)乘以 $b(t)$ 可以得到式(16)

$$0 = b(t)(a(t) \oplus c_t^1 \oplus c_{t+1}^1) \tag{16}$$

将式(14)中的 t 用 $t+1$ 代替，得到

$$c_{t+1}^1 \oplus c_t^1 = b(t+1) \oplus 1 \oplus c_t^0 \oplus c_{t+1}^0 \oplus c_{t+2}^0 \tag{17}$$

将式(17)代入式(16)得

$$0 = b(t)(a(t) \oplus b(t+1) \oplus 1 \oplus c_t^0 \oplus c_{t+1}^0 \oplus c_{t+2}^0)$$

即

$$\begin{aligned}
0 &= z_t \oplus z_{t+1} \oplus z_{t+2} \oplus z_{t+3} \\
&\oplus \Pi_{t+1}^1(z_{t+1}z_t \oplus z_{t+1}z_{t+2} \oplus z_{t+1}z_{t+3} \oplus z_t \oplus z_{t+1} \oplus z_{t+2} \oplus z_{t+3}) \\
&\oplus \Pi_{t+1}^2(z_t \oplus z_{t+1} \oplus z_{t+2} \oplus z_{t+3}) \oplus \Pi_{t+1}^3 z_{t+1} \oplus \Pi_{t+1}^4 \\
&\oplus \Pi_t^1 \oplus \Pi_{t+1}^1 \Pi_t^1(1 \oplus z_{t+1}) \oplus \Pi_{t+1}^2 \Pi_t^1 \\
&\oplus \Pi_{t+2}^1 z_{t+2} \oplus \Pi_{t+1}^1 \Pi_{t+2}^1 z_{t+2}(1 \oplus z_{t+1}) \oplus \Pi_{t+1}^2 \Pi_{t+2}^1 z_{t+2} \\
&\oplus \Pi_{t+2}^2 \oplus \Pi_{t+2}^2 \Pi_{t+1}^1(1 \oplus z_{t+1}) \oplus \Pi_{t+1}^2 \Pi_{t+2}^2 \\
&\oplus \Pi_{t+3}^1 \oplus \Pi_{t+3}^1 \Pi_{t+1}^1(1 \oplus z_{t+1}) \oplus \Pi_{t+3}^1 \Pi_{t+1}^2
\end{aligned}$$

得证。

参考文献：

[1] Bluetooth TM, Bluetooth specification, version 1.2[EB/OL]. http://www.bluetooth.org, 2003.

[2] ARMKNECHT F, KRAUSE M. Algebraic attacks on combiners with memory[A]. Cryptology CRYPTO 2003, Lecture Notes in Computer Science[C]. California, America, 2003.162-175.

[3] COURTOIS N T. Fast algebraic attacks on stream ciphers with linear feedback[A]. Cryptology-CRYPTO 2003, Lecture Notes in Computer Science[C]. California America, 2003. 176-194.

[4] LU Y, VAUDENAY S. Faster correlation attack on Bluetooth key-stream generator E_0[A]. Cryptology-CRYPTO 2004, Lecture Notes in Computer Science[C]. California, America, 2004.407-425.

[5] LU Y, MEIER W, VAUDENAY S. The conditional correlation attack: a practical attack on Bluetooth encryption[A]. Cryptology CRYPTO'05[C]. California, America, 2005. 97-117.

[6] BLEICHENBACHER D. Security weaknesses in Bluetooth[A]. Proc RSA Security Conf Cryptographer's Track[C]. San Francisco, America, 2001. 176-191.

[7] FLUHRER S, LUCKS S. Analysis of the E_0 encryption system[A]. Selected Areas in Cryptography, Lecture Notes in Computer Science[C]. Toronto, Canada, 2001. 38-48.

[8] KRAUSE M. BDD-based cryptanalysis of keystream generators[A]. Advances in Cryptology-EUROCRYPT 2002[C]. Amsterdam: The Netherlands, 2002. 222-237.

[9] OPHIR L, AVISHAI W. A uniform framework for cryptanalysis of the Bluetooth E_0 cipher[A]. Proc of 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)[C]. Athens, Greece, 2005. 365-373.

[10] SHAKED Y, WOOL A. Cryptanalysis of the Bluetooth E_0 cipher using OBDD's[EB/OL]. http://eprint.iacr.org/2006/072.pdf, 2006.

[11] 郭峰, 庄弈琪. 蓝牙 E_0 加密算法安全分析[J]. 电子科技大学学报, 2006,35(2):160-164.
GUO F, ZHANG Y Q. Analysis of the E_0 encryption system in Bluetooth[J]. Journal of UEST of China, 2006,35(2):160-164.

[12] PILLAI N R, BEDI S S, KUMAR S, *et al*. Relation for algebraic attack on E0 combiner[EB/OL]. http://eprint.iacr.org/2010/129, 2010.

[13] ARMKNECHT F, KRAUSE M. Algebraic attacks on combiners with memory[A]. Lecture Notes in Computer Science[C]. California, America, 2003.162-175.

作者简介：

詹英杰（1985-），男，湖北武汉人，信息工程大学硕士生，主要研究方向为序列密码的分析与设计。

丁林（1987-），男，河南息县人，信息工程大学硕士生，主要研究方向为序列密码的分析与设计。

关杰（1974-），女，河南郑州人，博士，信息工程大学副教授、硕士生导师，主要研究方向为密码学与信息安全。