对288轮Trivium算法的线性分析

魏长河,李俊志,张少武
(解放军信息工程大学,河南郑州 450001)

摘 要: 此前对288轮Trivium算法线性分析的文章中,均将密钥视为随机且变化的值,这样对算法进行分析是存在问题的,攻击者实际上无法将得到的线性偏差用于对算法实施攻击. 本文在选择IV(Initialization Vector)攻击条件下,重新对288轮Trivium算法进行了线性分析. 由于将密钥比特作为未知的定值,因而由密钥比特组成的非线性项是定值,不会产生线性偏差,在选取10个特殊IV后,得到一个线性偏差为1.9E-6的线性逼近式.

关键词: 密码分析;线性分析;Trivium算法;线性偏差
中图分类号: TN918.1 **文献标识码:** A **文章编号:** 0372-2112 (2017)06-1456-06
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.06.025

Linear Cryptanalysis of 288-Round Trivium

WEI Chang-he, LI Jun-zhi, ZHANG Shao-wu
(*The PLA Information Engineering University, Zhengzhou, Henan* 450001, *China*)

Abstract: In the previous linear cryptanalysis of 288-round Trivium, it is problematic to treat the key as a random and changing value in the process of analysis. In this way the attackers actually cannot attack the cipher with the inaccurate linear bias. For the problem above, we present the linear cryptanalysis of 288-round Trivium afresh under chosen initialization vector (IV) condition. Because the key bits are fixed, the nonlinear term which consists of key bits should be constant and does not produce a linear bias, and we find a linear approximation with the linear bias of 1.9E-6 on the condition that 10 bits of the IV are fixed.

Key words: cryptanalysis; linear cryptanalysis; Trivium; linear bias

1 引言

线性密码分析方法[1,2]是Matsui等人提出的一种已知明文攻击方法,并成功的运用于迭代型分组密码算法,在对流密码算法的分析[3~5]中也取得了很好的效果,其核心思想是寻找密码算法中明文、密文和密钥的线性逼近式来对密码系统进行攻击. 引用Matsui等人的表达方法,设某个密码算法中密钥为K,明文为M,密文为C,分析者掌握足够多使用同一密钥加密的密文及其对应的明文,需要找到具有如下形式的线性表达式:

$$M[i_1,i_2,\cdots,i_a]\oplus C[j_1,j_2,\cdots,j_b]=K[k_1,k_2,\cdots,k_c] \tag{1}$$

其中数据块$A[i,j,\cdots,k]=A[i]\oplus A[j]\oplus\cdots\oplus A[k]$,并且该等式成立的概率$p\neq 1/2$,我们用$|1/2-p|$来刻画这个线性表达式的有效性,称之为线性偏差$\varepsilon$.

在对流密码算法进行线性分析时,设算法输出为Z,初始IV为$(\mathrm{iv}_1,\mathrm{iv}_2,\cdots)$,密钥$K$为$(k_1,k_2,\cdots)$,上述式(1)可化为:

$$Z=K[k_{s_1},k_{s_2},\cdots,k_{s_n}]\oplus \mathrm{iv}_{t_1}\oplus \mathrm{iv}_{t_2}\oplus\cdots\oplus \mathrm{iv}_{t_m},$$

其中$K[k_{s_1},k_{s_2},\cdots,k_{s_n}]$为密钥比特的一个表达式. 与分组密码的线性分析一样,我们依然用线性偏差ε来刻画这个等式的有效性. 通过线性化技术[6],将某些初始IV令为0,可以提高线性偏差ε. 下面我们简述一下对流密码进行线性分析的思路:

若找到这样一个线性逼近式:$Z=K[k_{s_1},k_{s_2},\cdots,k_{s_n}]\oplus \mathrm{iv}_{t_1}\oplus \mathrm{iv}_{t_2}\oplus\cdots\oplus \mathrm{iv}_{t_m}$,逼近式的线性偏差为$\varepsilon$,设某个样本为$(\mathrm{IV},K,Z)$,令$\mathrm{IV}[t_1,t_2,\cdots,t_m]=\mathrm{iv}_{t_1}\oplus \mathrm{iv}_{t_2}\oplus\cdots\oplus \mathrm{iv}_{t_m}$,密钥$K$未知(但固定),因而$K[k_{s_1},k_{s_2},\cdots,k_{s_n}]$是个未知的定值,初始IV和输出$Z$是我们已知的,计算$Z\oplus \mathrm{IV}[t_1,t_2,\cdots,t_m]$的值,它等于$K[k_{s_1},k_{s_2},\cdots,k_{s_n}]$概率应当为$1/2+\varepsilon$,随机选取$N$个样本$(\mathrm{IV}_1,K,Z_1)$,$(\mathrm{IV}_2,K,Z_2)$,$\cdots$,$(\mathrm{IV}_N,K,Z_N)$,对这$N$个$Z_i\oplus \mathrm{IV}_i[t_1,t_2,\cdots,$

收稿日期:2015-07-24;修回日期:2016-08-17;责任编辑:梅志强
基金项目:国家自然科学基金(No.61272041)

t_m]的值进行统计,0 和 1 的个数应当存在一个偏差 ε,这样就可以将输出 Z 与随机序列区分开. 更进一步,能够恢复出 $K[k_{s_1},k_{s_2},\cdots,k_{s_n}]$ 的值,若 N 个 $Z_i \oplus \mathrm{IV}_i[t_1,t_2,\cdots,t_m]$ 的值中"1"比"0"多,说明 $K[k_{s_1},k_{s_2},\cdots,k_{s_n}]=1$,反之,若"0"比"1"多,说明 $K[k_{s_1},k_{s_2},\cdots,k_{s_n}]=0$.

Trivium 算法[7]是 eSTREAM 工程[8]最终获选的 7 个算法中面向硬件实现的算法之一,其设计思想对深入研究基于非线性反馈移位寄存器设计的算法具有重要意义. 目前对 288 轮 Trivium 算法的线性分析结果主要如下:

2006 年,Turan 等人在文献[9]中对 288 轮 Trivium 算法进行线性分析时,将 10 个密钥比特,10 个初始 IV 比特令为 0,找到了 1 个线性偏差为 2^{-31} 的线性逼近式.

2011 年,贾艳艳等人在文献[10]中利用多线性分析理论对文献[9]中的结果进行了改进,在将 10 个密钥比特,13 个初始 IV 比特令为 0 后,找到 2 个线性偏差为 2^{-31} 的线性逼近式.

2014 年,孙文龙等人在文献[11]中提出了一种最佳线性逼近式搜索算法,在将 10 个密钥比特,10 个初始 IV 比特令为 0 后,找到 3 个线性偏差为 2^{-25} 的线性逼近式.

2015 年,李俊志等人在文献[12]中提出一种因式分解消项线性化方法,进一步改进了结果,在将 10 个密钥比特,10 个初始 IV 比特令为 0 后,找到 1 个线性偏差为 2^{-20} 的线性逼近式.

上述分析结果都是在相关密钥条件下获得了,思路基本一致,在对输出 Z 进行线性逼近时,都将密钥作为随机变化且未知的值. 在文献[9]中,Turan 等人对 288 轮 Trivium 算法进行线性分析,给出了分析的两个步骤和分析过程中出现的各类表达式,做了大量的基础工作,并将 10 个密钥比特和 10 个初始 IV 令为 0,消去了一部分非线性项,最后通过计算非线性项个数,使用堆积引理给出了一个理论上的线性偏差. 而实际上,在对 Trivium 算法进行线性分析时,如果在线性逼近的过程中将密钥作为随机变化且未知的值,这样对算法进行线性逼近是存在问题的,攻击者实际上并不能检测到这种理论上的线性偏差,无法用于对算法实施攻击. 随后,文献[10~12]在 Turan 等人的基础上,各自利用不同的方法对分析结果进行了改进,但重点研究的都是:在将尽量少的密钥和 IV 比特令为 0 的前提下,怎样更多的消去线性逼近过程中的非线性项,进而提高这种理论上的线性偏差. 因而,文献[10~12]均未能避免文献[9]中存在的问题.

本文针对此前分析结果中存在的问题,将密钥作为未知的定值,在选择 IV 攻击条件下,重新对 288 轮 Trivium 算法进行了分析,在将 10 个初始 IV 比特令为 0 后,找到一个线性偏差为 2^{-19} 的线性逼近式,当数据复杂度为 2^{38} 时成功率为 97.8%,并能以 $2^{41.46}$ 的计算复杂度恢复出 4 个密钥和 7 个密钥式的值,给出了对 288 轮 Trivium 算法新的线性分析结果.

2　流密码算法选择 IV 条件下的线性分析

2.1　基础知识

引理 1[1]　设 N 是样本量(已知的随机初始 IV 及其对应输出 Z 的数量),p 是线性逼近式成立的概率,线性偏差为 $\varepsilon=|p-0.5|$,那么线性攻击的成功率是:

$$\int_{-2\sqrt{N}\cdot\varepsilon}^{\infty}\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-x^2/2}\mathrm{d}x \tag{2}$$

特别地,当 $N=\varepsilon^{-2}$ 时,线性攻击的成功率为 97.8%.

引理 2(堆积引理)[1]　设 $x_i(i=1,2,\cdots,n)$ 是相互独立的变量,第 i 个变量等于 0 的概率为 p_i,则 $x_1 \oplus x_2 \oplus\cdots\oplus x_n=0$ 的概率为:

$$\frac{1}{2}+2^{n-1}\prod_{i=1}^{n}\left(p_i-\frac{1}{2}\right) \tag{3}$$

2.2　此前对 288 轮 Trivium 算法线性分析中存在的问题

线性密码分析方法最早是对分组密码算法提出的一种已知明文攻击方法,并成功的攻击了 DES 算法,在分析时,攻击者必须掌握足够多的明文和使用同一密钥加密后对应的密文,才能利用明密文之间的线性逼近式对算法进行攻击. 在对流密码算法进行线性分析时,攻击者同样需要掌握足够多的初始 IV 和使用同一密钥 K 迭代后对应的输出 Z. 而文献[9~12]中对 Trivium 算法的线性分析结果都是在相关密钥条件下获得的,存在的问题均为:对输出 Z 进行线性逼近时,将密钥作为随机且变化的值,这样得到的线性逼近式和相应的线性偏差是无法进行检测的. 下面我们举例进行说明:

例 1　若算法的输出 Z 关于初始 IV 和密钥 K 的表达式的线性部分中存在密钥比特. 为了简明直观,我们不妨设此表达式为:$Z=k_6+\mathrm{iv}_3+k_4\cdot k_5+\mathrm{iv}_{13}\cdot\mathrm{iv}_{14}+\mathrm{iv}_9\cdot k_7$.

若在分析时将密钥 K 作为随机且变化的值,在对输出进行线性逼近时,非线性项 $\mathrm{iv}_{13}\cdot\mathrm{iv}_{14}$,$k_4\cdot k_5$ 和 $\mathrm{iv}_9\cdot k_7$ 均会产生线性偏差,由此计算输出 Z 关于线性逼近式 $k_3+\mathrm{iv}_3$ 的线性偏差为 1/16,即:$Z+\mathrm{iv}_3=k_6$ 的概率为 1/2+1/16. 随机选取 N 个样本(IV_1,K_1,Z_1),(IV_2,K_2,Z_2),$\cdots$,(IV_N,K_N,Z_N),由于在各个样本中我们已知的参量是(IV_i,Z_i),密钥 K_i 是随机且未知的,不同的样本使用的密钥也不同,因而在等式 $Z+\mathrm{iv}_3=k_6$ 中,我们仅能得到 $Z+\mathrm{iv}_3$ 的值,而 $K[k_{s_1},k_{s_2},\cdots,k_{s_n}]$(即 k_6)的值

是随机变化且未知的,所以我们无法对这 N 个样本对应的 $Z_i + \mathrm{iv}_3^i + k_6^i$ 的值进行统计,更无法对偏差进行检测. 另外,若对 N 个样本对应的 $Z_i + \mathrm{iv}_3^i$ 的值进行统计时,由于 k_6 的值是随机变化且未知的,因而 $Z_i + \mathrm{iv}_3^i$ 的值也一定是随机的,不存在偏差,无法对算法进行攻击.

例 2 若算法的输出 Z 关于初始 IV 和密钥 K 的表达式的线性部分中不存在密钥比特. 我们不妨设此表达式为:$Z = \mathrm{iv}_3 + k_4 \cdot k_5 + \mathrm{iv}_{13} \cdot \mathrm{iv}_{14} + \mathrm{iv}_9 \cdot k_7$.

若在分析时将密钥 K 当作是随机变化的,输出 Z 关于线性逼近式 iv_3 的线性偏差为 1/16,即:$Z + \mathrm{iv}_3 = 0$ 的概率为 1/2 + 1/16. 随机选取 N 个样本$(\mathrm{IV}_1, K_1, Z_1)$,$(\mathrm{IV}_2, K_2, Z_2)$,…,$(\mathrm{IV}_N, K_N, Z_N)$,在各个样本中我们已知的参量是$(\mathrm{IV}_i, Z_i)$,密钥 K_i 是随机变化且未知的,不同的样本使用的密钥也不同. 但在等式 $Z + \mathrm{iv}_3 = 0$ 中,由于不包含密钥比特,虽然密钥 K_i 未知,但不会影响我们对这 N 个样本对应的 $Z_i + \mathrm{iv}_3^i$ 的值进行统计,因此可以对线性偏差进行检测,进而对算法进行区分攻击,但无法恢复密钥.

由此可知,若在分析时将密钥 K 当作是随机变化的值,只有当输出 Z 关于初始 IV 和密钥 K 的表达式的线性部分中不存在密钥比特时,才能对得到的线性偏差进行检测,进而对算法进行区分攻击. 但对于 288 轮 Trivium 算法,其输出 Z 关于初始 IV 和密钥 K 的表达式的线性部分中,含有 13 个密钥比特,是无法对算法进行攻击的,因此文献[9~12]中关于攻击成功率的分析实际上是不对的. 另外,我们经过计算机推导,随着算法迭代轮数的增加,密钥比特的数量还会继续增多. 因而在对 Trivium 算法进行线性分析时,不能将密钥 K 作为随机变化的值,而应该将密钥作为未知的常量,在选择 IV 条件下对算法进行线性分析.

2.3 流密码算法选择 IV 条件下的线性分析

流密码的线性分析是寻找密码算法中初始 IV、密钥 K 和输出 Z 的线性逼近式. 设某个密码算法初始 IV 为$(\mathrm{iv}_1, \mathrm{iv}_2, \cdots)$,密钥 K 为$(k_1, k_2, \cdots)$,在选择 IV 条件下对流密码进行线性分析时,分析者需要找到具有如下形式的线性逼近式:

$$Z = K[k_{s_1}, k_{s_2}, \cdots, k_{s_n}] \oplus \mathrm{iv}_{t_1} \oplus \mathrm{iv}_{t_2} \oplus \cdots \oplus \mathrm{iv}_{t_m},$$

其中 $K[k_{s_1}, k_{s_2}, \cdots, k_{s_n}]$ 为密钥比特的一个表达式,且表达式 $K[k_{s_1}, k_{s_2}, \cdots, k_{s_n}]$ 中允许存在由密钥比特组成的非线性项. 我们希望该线性逼近式成立的概率 $p \neq 1/2$,用 $|1/2 - p|$ 来刻画它的有效性,称之为线性偏差 ε.

在对流密码算法进行线性分析时,随机选取一个密钥 K(密钥 K 未知但确定),第 1 步得到输出 Z 关于寄存器某一中间时刻内部状态的表达式,进行一次线性逼近,使用堆积引理计算输出关于该表达式中线性部分的线性偏差;第 2 步用密钥 K 和初始 IV 将中间状态表达式中的线性部分表示出来,得到一个输出关于密钥 K 和初始 IV 的表达式,将此表达式分成线性部分和非线性部分,并将非线性部分中的非线性项分成以下 3 类:

Type 0:仅与密钥有关;

Type 1:既与初始 IV 有关,又与密钥有关;

Type 2:仅与初始 IV 有关.

其中,Type 0 型的非线性项只与密钥有关,因而是常量,可以归入线性部分中去,作为线性逼近式中 $K[k_{s_1}, k_{s_2}, \cdots, k_{s_n}]$ 的一部分. Type 2 型的非线性项会产生线性偏差,我们可以通过将某些 IV 比特令为 0,使其个数减少,进而提高线性偏差.

对于 Type 1 型的非线性项,若某个 Type 1 型的非线性项中 IV 比特的个数为 1,会使得我们无法得到确定的线性逼近式;若 IV 比特的个数大于 1,虽然不会影响到线性逼近式,但由于密钥比特的不确定,会使得线性偏差不稳定. 比如:对于非线性项 $k_s \cdot \mathrm{iv}_t$,当密钥比特 $k_s = 1$ 时,$k_s \cdot \mathrm{iv}_t = \mathrm{iv}_t$,是线性项,应当归类于线性逼近式中,而当 $k_s = 0$ 时,$k_s \cdot \mathrm{iv}_t$ 又变为 0,由于密钥的未知,这会使得我们无法得到确定的线性逼近式;对于非线性项 $k_s \cdot \mathrm{iv}_t \cdot \mathrm{iv}_r$,当密钥比特 $k_s = 1$ 时,$k_s \cdot \mathrm{iv}_t \cdot \mathrm{iv}_r = \mathrm{iv}_t \cdot \mathrm{iv}_r$,应当归类于 Type 2 型的非线性项中,而当 $k_s = 0$ 时,$k_s \cdot \mathrm{iv}_t \cdot \mathrm{iv}_r$ 又变为 0,这会使得逼近式的线性偏差不稳定. 因此,我们需要选取特殊 IV,使得 Type 1 型的非线性项化为 0 或定值,以消除这种不稳定. 因为要选择特殊 IV,我们对流密码进行线性分析时,应当在选择 IV 攻击条件下进行.

在对流密码算法进行线性分析的过程中,进行线性逼近时需要注意以下两方面问题:

(1)使用堆积引理计算线性偏差时,不应该将由密钥组成的非线性项计算进去. 例如:设算法输出关于初始状态的表达式为:$Z = k_6 + \mathrm{iv}_3 + k_4 \cdot k_5 + \mathrm{iv}_{13} \cdot \mathrm{iv}_{14}$,在计算线性偏差时,Type 0 型非线性项 $k_4 \cdot k_5$ 实际上是定值,因此应当归于线性逼近式中的 $K[k_{s_1}, k_{s_2}, \cdots, k_{s_n}]$ 中.

(2)某些寄存器中间时刻的内部状态只跟密钥有关,因此是定值,在进行线性逼近时,应将由这种内部状态组成的非线性项归于线性逼近式中的 $K[k_{s_1}, k_{s_2}, \cdots, k_{s_n}]$ 中.

3 对 288 轮简化版 Trivium 算法的线性分析

3.1 算法描述

Trivium 算法是面向硬件实现的同步流密码算法,其初始化向量和密钥都是 80 比特,采用了三个非线性反馈移位寄存器,长度分别为 93,84 和 111,内部状态总共为 288 比特. 三个寄存器的初始化如下:

$$(s_1, s_2, \cdots, s_{93}) \leftarrow (k_1, k_2, \cdots, k_{80}, 0, \cdots, 0)$$

$$(s_{94},s_{95},\cdots,s_{177})\leftarrow(x_1,x_2,\cdots,x_{80},0,0,0,0)$$
$$(s_{178},s_{179},\cdots,s_{288})\leftarrow(0,0,\cdots,0,1,1,1)$$

状态更新过程如下:

$$t_1=s_{66}+s_{93}$$
$$t_2=s_{162}+s_{177}$$
$$t_3=s_{243}+s_{288}$$
$$Z=t_1+t_2+t_3$$
$$t_1\leftarrow t_1+s_{91}\cdot s_{92}+s_{171}$$
$$t_2\leftarrow t_2+s_{175}\cdot s_{176}+s_{264}$$
$$t_3\leftarrow t_3+s_{286}s_{287}+s_{69}$$
$$(s_1,s_2,\cdots,s_{93})\leftarrow(t_3,s_1,\cdots,s_{92})$$
$$(s_{94},s_{95},\cdots,s_{177})\leftarrow(t_1,s_{94},\cdots,s_{176})$$
$$(s_{178},s_{179},\cdots,s_{288})\leftarrow(t_2,s_{178},\cdots,s_{287})$$

算法分为两个阶段,初始化阶段和密钥流生成阶段,这两个阶段的寄存器更新函数完全一样,只是在初始化阶段不输出密钥流 Z,算法的初始化轮数为 1152.

3.2　分析过程

对于 288 轮简化版 Trivium 算法,设输出 Z 为算法 288 轮迭代后密钥流生成阶段输出的第一比特. 由于直接计算输出 Z 关于寄存器初始状态的表达式太过复杂,我们先计算出输出 Z 关于寄存器第 144 轮内部状态的表达式,在此引用文献[9]中的结果:

$$\begin{aligned}Z=&s_{144}(6)+s_{144}(33)+s_{144}(57)+s_{144}(84)+s_{144}(96)\\&+s_{144}(99)+s_{144}(111)+s_{144}(129)+s_{144}(144)\\&+s_{144}(150)+s_{144}(162)+s_{144}(165)+s_{144}(186)\\&+s_{144}(192)+s_{144}(210)+s_{144}(231)+s_{144}(237)\\&+s_{144}(252)+s_{144}(16)\cdot s_{144}(17)+s_{144}(31)\cdot s_{144}(32)\\&+s_{144}(82)\cdot s_{144}(83)+s_{144}(97)\cdot s_{144}(98)\\&+s_{144}(142)\cdot s_{144}(143)+s_{144}(163)\cdot s_{144}(164)\\&+s_{144}(208)\cdot s_{144}(209)+s_{144}(235)\cdot s_{144}(236)\end{aligned}\tag{2}$$

上述式(2)中的非线性项共有 8 个,涉及 16 个中间状态,但由于 $s_{144}(82)=s_7+s_{181}+s_{226}+s_{224}\cdot s_{225}=k_7$, $s_{144}(83)=s_8+s_{182}+s_{227}+s_{225}\cdot s_{226}=k_8$,这两个中间时刻寄存器状态只跟密钥有关,因此非线性项 $s_{144}(82)\cdot s_{144}(83)=k_7\cdot k_8$ 是未知的常量. 对其它 7 个非线性项,我们使用堆积引理计算出输出 Z 关于下述式(3)的线性偏差 $\varepsilon_1=2^6\cdot(0.25)^7=2^{-8}$:

$$\begin{aligned}A=&s_{144}(6)+s_{144}(33)+s_{144}(57)+s_{144}(84)\\&+s_{144}(96)+s_{144}(99)+s_{144}(111)+s_{144}(129)\\&+s_{144}(144)+s_{144}(150)+s_{144}(162)+s_{144}(165)\\&+s_{144}(186)+s_{144}(192)+s_{144}(210)+s_{144}(231)\\&+s_{144}(237)+s_{144}(252)+k_7\cdot k_8\end{aligned}\tag{3}$$

将式(3)中的第 144 轮寄存器内部状态用寄存器初始状态表示出来:

$$\begin{aligned}A=&1+s_3+s_6+s_{15}+s_{21}+s_{27}+s_{30}+s_{39}+s_{54}+s_{57}\\&+s_{67}+s_{68}+s_{69}+s_{72}+s_{96}+s_{99}+s_{114}+s_{117}+s_{123}\\&+s_{126}+s_{132}+s_{138}+s_{144}+s_{165}+s_{171}+s_4\cdot s_5+s_{13}\cdot s_{14}\\&+s_{13}\cdot s_{41}+s_{13}\cdot s_{119}+s_{14}\cdot s_{40}+s_{14}\cdot s_{118}+s_{16}\cdot s_{17}\\&+s_{19}\cdot s_{20}+s_{19}\cdot s_{47}+s_{19}\cdot s_{125}+s_{20}\cdot s_{46}+s_{20}\cdot s_{124}\\&+s_{22}\cdot s_{23}+s_{25}\cdot s_{26}+s_{28}\cdot s_{29}+s_{34}\cdot s_{35}+s_{37}\cdot s_{38}\\&+s_{37}\cdot s_{65}+s_{37}\cdot s_{143}+s_{38}\cdot s_{64}+s_{38}\cdot s_{142}+s_{39}\cdot s_{40}\\&+s_{40}\cdot s_{119}+s_{41}\cdot s_{118}+s_{43}\cdot s_{44}+s_{45}\cdot s_{46}+s_{46}\cdot s_{125}\\&+s_{47}\cdot s_{124}+s_{49}\cdot s_{50}+s_{52}\cdot s_{53}+s_{58}\cdot s_{59}+s_{58}\cdot s_{164}\\&+s_{59}\cdot s_{163}+s_{61}\cdot s_{62}+s_{63}\cdot s_{64}+s_{64}\cdot s_{65}+s_{64}\cdot s_{143}\\&+s_{64}\cdot s_{170}+s_{65}\cdot s_{142}+s_{65}\cdot s_{169}+s_{67}\cdot s_{68}+s_{70}\cdot s_{71}\\&+s_{103}\cdot s_{104}+s_{106}\cdot s_{107}+s_{118}\cdot s_{119}+s_{124}\cdot s_{125}\\&+s_{127}\cdot s_{128}+s_{130}\cdot s_{131}+s_{133}\cdot s_{149}+s_{134}\cdot s_{148}\\&+s_{142}\cdot s_{143}+s_{147}\cdot s_{148}+s_{151}\cdot s_{152}+s_{154}\cdot s_{155}\\&+s_{160}\cdot s_{161}+s_{163}\cdot s_{164}+s_{166}\cdot s_{167}+s_{13}\cdot s_{39}\cdot s_{40}\\&+s_{14}\cdot s_{38}\cdot s_{39}+s_{19}\cdot s_{45}\cdot s_{46}+s_{20}\cdot s_{44}\cdot s_{45}\\&+s_{37}\cdot s_{63}\cdot s_{64}+s_{38}\cdot s_{39}\cdot s_{40}+s_{38}\cdot s_{39}\cdot s_{41}\\&+s_{38}\cdot s_{39}\cdot s_{119}+s_{38}\cdot s_{62}\cdot s_{63}+s_{39}\cdot s_{40}\cdot s_{118}\\&+s_{44}\cdot s_{45}\cdot s_{46}+s_{44}\cdot s_{45}\cdot s_{47}+s_{44}\cdot s_{45}\cdot s_{125}\\&+s_{45}\cdot s_{46}\cdot s_{124}+s_{62}\cdot s_{63}\cdot s_{64}+s_{62}\cdot s_{63}\cdot s_{65}\\&+s_{62}\cdot s_{63}\cdot s_{143}+s_{63}\cdot s_{64}\cdot s_{142}+s_{133}\cdot s_{147}\cdot s_{148}\\&+s_{134}\cdot s_{146}\cdot s_{147}+s_{146}\cdot s_{147}\cdot s_{148}+s_{146}\cdot s_{147}\cdot s_{149}\\&+k_7\cdot k_8\end{aligned}\tag{4}$$

用密钥 K 和初始 IV 将上述式(4)中的寄存器初始状态表示出来,将得到的表达式分成线性部分和非线性部分,并将非线性部分中的非线性项分成 Type 0, Type 1 和 Type 2 三类:

对于式(5)中的 Type 0 型非线性项,由于此类非线性项是由密钥组成的,因而是定值,应当归于线性部分中.

$$\begin{aligned}A=&(1+k_3+k_6+k_{15}+k_{21}+k_{27}+k_{30}+k_{39}+k_{54}+k_{57}\\&+k_{67}+k_{68}+k_{69}+k_{72}+\mathrm{iv}_3+\mathrm{iv}_6+\mathrm{iv}_{21}+\mathrm{iv}_{24}+\mathrm{iv}_{30}\\&+\mathrm{iv}_{33}+\mathrm{iv}_{39}+\mathrm{iv}_{45}+\mathrm{iv}_{51}+\mathrm{iv}_{72}+\mathrm{iv}_{78})+[(k_4\cdot k_5\\&+k_{13}\cdot k_{14}+k_{13}\cdot k_{41}+k_{16}\cdot k_{17}+k_{14}\cdot k_{40}+k_{19}\cdot k_{20}\\&+k_{19}\cdot k_{47}+k_{20}\cdot k_{46}+k_{22}\cdot k_{23}+k_{25}\cdot k_{26}+k_{28}\cdot k_{29}\\&+k_{34}\cdot k_{35}+k_{37}\cdot k_{38}+k_{37}\cdot k_{65}+k_{38}\cdot k_{64}+k_{39}\cdot k_{40}\\&+k_{43}\cdot k_{44}+k_{45}\cdot k_{46}+k_{49}\cdot k_{50}+k_{52}\cdot k_{53}+k_{58}\cdot k_{59}\\&+k_{61}\cdot k_{62}+k_{63}\cdot k_{64}+k_{64}\cdot k_{65}+k_{67}\cdot k_{68}+k_{70}\cdot k_{71}\\&+k_{13}\cdot k_{39}\cdot k_{40}+k_{14}\cdot k_{38}\cdot k_{39}+k_{19}\cdot k_{45}\cdot k_{46}\\&+k_{20}\cdot k_{44}\cdot k_{45}+k_{37}\cdot k_{63}\cdot k_{64}+k_{38}\cdot k_{39}\cdot k_{40}\\&+k_{38}\cdot k_{39}\cdot k_{41}+k_{38}\cdot k_{62}\cdot k_{63}+k_{44}\cdot k_{45}\cdot k_{46}\\&+k_{44}\cdot k_{45}\cdot k_{47}+k_{62}\cdot k_{63}\cdot k_{64}+k_{62}\cdot k_{63}\cdot k_{65})\\&+k_7\cdot k_8]+[(k_{14}+k_{41}+k_{39}\cdot k_{40})\cdot \mathrm{iv}_{25}+(k_{13}\\&+k_{40}+k_{38}\cdot k_{39})\cdot \mathrm{iv}_{26}+(k_{20}+k_{47}+k_{45}\cdot k_{46})\cdot \mathrm{iv}_{31}\\&+(k_{19}+k_{46}+k_{44}\cdot k_{45})\cdot \mathrm{iv}_{32}+(k_{38}+k_{65}+k_{63}\cdot k_{64})\cdot \mathrm{iv}_{49}\\&+(k_{37}+k_{64}+k_{62}\cdot k_{63})\cdot \mathrm{iv}_{50}+k_{59}\cdot \mathrm{iv}_{70}+k_{58}\cdot \mathrm{iv}_{71}\\&+k_{65}\cdot \mathrm{iv}_{76}+k_{64}\cdot \mathrm{iv}_{77}]+(\mathrm{iv}_{10}\cdot \mathrm{iv}_{11}+\mathrm{iv}_{13}\cdot \mathrm{iv}_{14}\\&+\mathrm{iv}_{25}\cdot \mathrm{iv}_{26}+\mathrm{iv}_{31}\cdot \mathrm{iv}_{32}+\mathrm{iv}_{34}\cdot \mathrm{iv}_{35}+\mathrm{iv}_{37}\cdot \mathrm{iv}_{38}\\&+\mathrm{iv}_{40}\cdot \mathrm{iv}_{56}+\mathrm{iv}_{41}\cdot \mathrm{iv}_{55}+\mathrm{iv}_{49}\cdot \mathrm{iv}_{50}+\mathrm{iv}_{54}\cdot \mathrm{iv}_{55}\end{aligned}$$

$$+iv_{58}\cdot iv_{59}+iv_{61}\cdot iv_{62}+iv_{67}\cdot iv_{68}+iv_{70}\cdot iv_{71}$$
$$+iv_{73}\cdot iv_{74}+iv_{40}\cdot iv_{54}\cdot iv_{55}+iv_{41}\cdot iv_{53}\cdot iv_{54}$$
$$+iv_{53}\cdot iv_{54}\cdot iv_{55}+iv_{53}\cdot iv_{54}\cdot iv_{56}) \quad (5)$$

而对于 Type 1 型的非线性项,我们将如下 10 个初始 IV 比特令为 0:

$$iv_{25}=0, iv_{26}=0, iv_{31}=0, iv_{32}=0, iv_{49}=0,$$
$$iv_{50}=0, iv_{70}=0, iv_{71}=0, iv_{76}=0, iv_{77}=0,$$

使 Type 1 型的非线性项全部变为 0,以获得稳定的线性逼近式和线性偏差.

对于 Type 2 型非线性项,将其分为线性无关的若干组,分别穷举各组中所包含的变量,通过真值表法来计算该组非线性项为 0 概率,最后用堆积引理求出所有 Type 2 型非线性项相加为 0 的概率. 通过计算,我们得出式(5)中 Type 2 型非线性项相加为 0 的概率为 $P=0.5+2^{-12}$.

至此,我们将式(5)中的三类非线性项处理完毕,式(5)等于下述式(6)的概率为 $P=0.5+2^{-12}$,线性偏差 $\varepsilon_2=2^{-12}$. 设式(5)中的密钥项之和为 K',则:

$$\begin{aligned}B=&(1+k_3+k_6+k_{15}+k_{21}+k_{27}+k_{30}+k_{39}+k_{54}+k_{57}\\&+k_{67}+k_{68}+k_{69}+k_{72}+k_7\cdot k_8+k_4\cdot k_5+k_{13}\cdot k_{14}\\&+k_{13}\cdot k_{41}+k_{16}\cdot k_{17}+k_{14}\cdot k_{40}+k_{19}\cdot k_{20}+k_{19}\cdot k_{47}\\&+k_{20}\cdot k_{46}+k_{22}\cdot k_{23}+k_{25}\cdot k_{26}+k_{28}\cdot k_{29}+k_{34}\cdot k_{35}\\&+k_{37}\cdot k_{38}+k_{37}\cdot k_{65}+k_{38}\cdot k_{64}+k_{39}\cdot k_{40}+k_{43}\cdot k_{44}\\&+k_{45}\cdot k_{46}+k_{49}\cdot k_{50}+k_{52}\cdot k_{53}+k_{58}\cdot k_{59}+k_{61}\cdot k_{62}\\&+k_{63}\cdot k_{64}+k_{64}\cdot k_{65}+k_{67}\cdot k_{68}+k_{70}\cdot k_{71}\\&+k_{13}\cdot k_{39}\cdot k_{40}+k_{14}\cdot k_{38}\cdot k_{39}+k_{19}\cdot k_{45}\cdot k_{46}\\&+k_{20}\cdot k_{44}\cdot k_{45}+k_{37}\cdot k_{63}\cdot k_{64}+k_{38}\cdot k_{39}\cdot k_{40}\\&+k_{38}\cdot k_{39}\cdot k_{41}+k_{38}\cdot k_{62}\cdot k_{63}+k_{44}\cdot k_{45}\cdot k_{46}\\&+k_{44}\cdot k_{45}\cdot k_{47}+k_{62}\cdot k_{63}\cdot k_{64}+k_{62}\cdot k_{63}\cdot k_{65})\\&+(iv_3+iv_6+iv_{21}+iv_{24}+iv_{30}+iv_{33}+iv_{39}+iv_{45}\\&+iv_{51}+iv_{72}+iv_{78})=1+K'+(iv_3+iv_6+iv_{21}+iv_{24}\\&+iv_{30}+iv_{33}+iv_{39}+iv_{45}+iv_{51}+iv_{72}+iv_{78}) \quad (6)\end{aligned}$$

最后,结合输出 Z 关于式(3)的线性偏差 $\varepsilon_1=2\cdot(0.25)^7=2^{-8}$,得到输出 Z 关于线性式逼近 B(即式(6))的线性偏差 $\varepsilon=2\varepsilon_1\cdot\varepsilon_2=2^{-19}$,为使线性分析的成功率达到 97.8%,所需的样本量为 2^{38},并能恢复出密钥式 K'的值.

表 1 结果对比

攻击条件	选择密钥和 IV 量	偏差	数据量	成功率	来源
相关密钥	(10,10)	2^{-31}	——	——	文献[9]
相关密钥	(10,13)	2^{-31}	——	——	文献[10]
相关密钥	(10,10)	2^{-25}	——	——	文献[11]
相关密钥	(10,10)	2^{-20}	——	——	文献[12]
选择 IV	(0,10)	2^{-19}	2^{38}	0.978	本文

注:文献[9~12]中分析结果存在问题,无法对算法进行攻击,因此成功率和数据量用"——"代替

更进一步,在选择特殊 IV 对 Type 1 型非线性项进行处理时,可令 10 个特殊 IV 中的某一个为 1,其它 9 个为 0,不妨令:

$$iv_{25}=1, iv_{26}=0, iv_{31}=0, iv_{32}=0, iv_{49}=0,$$
$$iv_{50}=0, iv_{70}=0, iv_{71}=0, iv_{76}=0, iv_{77}=0,$$

则所有的 Type 1 型非线性项化简之后等于定值 $k_{14}+k_{41}+k_{39}\cdot k_{40}$,最终我们能恢复出 $K'+k_{14}+k_{41}+k_{39}\cdot k_{40}$的值,结合之前恢复出的 K',能够恢复出 $k_{14}+k_{41}+k_{39}\cdot k_{40}$的值. 同理,分别让这 10 个特殊 IV 中的某一个为 1,其它的为 0,能够恢复出以下 10 个密钥或密钥式的值:

$$k_{58}, k_{59}, k_{64}, k_{65}, k_{13}+k_{40}+k_{38}\cdot k_{39}, k_{14}+k_{41}+k_{39}\cdot k_{40},$$
$$k_{19}+k_{46}+k_{44}\cdot k_{45}, k_{20}+k_{47}+k_{45}\cdot k_{46}, k_{37}+k_{64}+k_{62}\cdot k_{63},$$
$$k_{38}+k_{65}+k_{63}\cdot k_{64}$$

且恢复出 K'和上述 10 个密钥或密钥式的值所需的计算复杂度为 $11\times 2^{38}\approx 2^{41.46}$.

4 结束语

本文在选择 IV 攻击条件下,对初始化轮数为 288 轮的简化版 Trivium 算法进行了线性分析,在选取 10 个特殊 IV 后,得到一个线性偏差为 2^{-19}的线性逼近式,当数据复杂度为 2^{38}时成功率为 97.8%,将计算复杂度提高 11 倍后,能够恢复出 11 个密钥或密钥式的值,更正了此前对 288 轮简化版 Trivium 算法的线性分析结果. 进一步提高线性逼近概率和对更多轮数的 Trivium 算法进行线性分析是需要继续研究的问题.

参考文献

[1] Matsui M. Linear cryptanalysis method for DES cipher [A]. EUROCRYPT 1993 [C]. Germany: Springer-Verlag, 1993. 386 – 397.

[2] Matsui M, Yamagishi A. A new method for known plaintext attack of feal cipher[A]. EUROCRYPT 1992[C]. Germany: Springer-Verlag, 1992. 81 – 91.

[3] Lu Y, Vaudenay S, Meier W. Synthetic linear analysis with applications to CubeHash and Rabbit[J]. Cryptography and Communications, 2012, 4(3-4): 259 – 276.

[4] Muller F, Peyrin T. Linear cryptanalysis of the TSC family of stream ciphers[A]. ASIACRYPT 2005[C]. Germany: Springer-Verlag, 2005. 373 – 394.

[5] Golić J D, Bagini V, Morgari G. Linear cryptanalysis of bluetooth stream cipher[A]. EUROCRYPT 2002[C]. Germany: Springer-Verlag, 2002. 238 – 255.

[6] Segers A J M. Algebraic attacks from a Grobner basis perspective[J]. International Journal of Algebra and Computation, 2004, (9-12): 447 – 459.

[7] Cannière C D. Trivium: A stream cipher construction inspired by block cipher design principles[J]. Lecture Notes

in Computer Science,2006,4176:171 – 186.

[8] ECRYPT. eSTREAM: ECRYPT Stream Cipher Project, IST-2002-507932[EB/OL]. http://www. ecrypt. eu. org/stream,2005-04-12.

[9] Turan M S,Kara O. Linear approximations for 2-round trivium[A]. Security of Information and Networks 2007[C]. USA:Trafford Publishing,2007. 96 – 105.

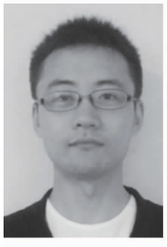
[10] 贾艳艳,胡予濮,杨文峰,高军涛. 2 轮 Trivium 的多线性密码分析[J]. 电子与信息学报,2011,33(1):223 – 227.
Jia Y Y,Hu Y P,Yang W F,Gao J T. Linear cryptanalysis of 2-round trivium with multiple approximations[J]. Journal of Electronics & Information Technology, 2011, 33(1):223 – 227. (in Chinese)

[11] 孙文龙,关杰,刘建东. 针对简化版 Trivium 算法的线性分析[J]. 计算机学报,2012,35(9):1890 – 1896.
Sun W L,Guan J,Liu J D. Linear cryptanalysis of simplified trivium[J]. Chinese Journal of Computers,2012,35(9):1890 – 1896. (in Chinese)

[12] 李俊志,关杰,孙文龙. 一种改进的线性化技术及其应用[J]. 密码学报,2014,(5):491 – 503.
Li J Z,Guan J,Sun W L. A modified linearization technique and its application[J]. Journal of Cryptologic Research,2014,1(5):491 – 504. (in Chinese)

作者简介

魏长河　男,1991 年出生,湖北随州人,解放军信息工程大学硕士研究生,主要研究方向为序列密码的设计与分析.
E-mail:416282490@qq.com



李俊志　男,1990 年出生,河南新乡人,解放军信息工程大学博士研究生,主要研究方向为流密码的设计与分析.
E-mail:lijunzhi1998@163.com

张少武　男,1962 年出生,河南郑州人,解放军信息工程大学教授,硕士生导师,主要研究方向为序列密码的设计与分析.
E-mail:zhangsw37@sina.com