针对简化版 Trivium 算法的线性分析

孙文龙[1)] 关 杰[1)] 刘建东[2)]

[1)](信息工程大学电子技术学院 郑州 450004)

[2)](北京市2857信箱四室 北京 100085)

摘 要 流密码 Trivium 算法是欧洲密码工程 eSTREAM 的7个最终获选算法之一.该文针对初始化为288轮的简化版 Trivium 算法进行了线性分析,更正了 Turan 等人给出的关于密钥、初始化向量和密钥流比特的表达式,并给出了当允许选取特殊的密钥和 ***IV*** 时,搜索最佳线性逼近式的算法.据此算法,找到了3个线性偏差为 2^{-25} 的线性逼近式,改进了 Turan 等人给出的线性偏差为 2^{-31} 的线性分析结果.

关键词 密码分析;线性分析;Trivium;流密码

中图法分类号 TN918 **DOI 号**: 10.3724/SP.J.1016.2012.01890

Linear Cryptanalysis of Simplified Trivium

SUN Wen-Long[1)] GUAN Jie[1)] LIU Jian-Dong[2)]

[1)](*Electronic Technology Institute, Information Engineering University, Zhengzhou* 450004)

[2)](*Beijing* 2857 *Mailbox, Beijing* 100085)

Abstract Stream cipher Trivium is one of the seven finalists of the eSTREAM project. In this paper, we apply linear cryptanalysis to the simplified Trivium with the initialization of 288 rounds. The equation, which involves the key bits, initial vector bits and the first keystream bit in linear approximations for 288-round Trivium of Turan, is corrected. In addition, when special Key bits and ***IV*** bits are allowed to be chosen, the algorithm to search the linear approximations with the biggest linear bias is presented. Based on this algorithm, 3 linear approximations with the same linear bias 2^{-25} are found, which is better than Turan's 2^{-31}.

Keywords cryptanalysis; linear cryptanalysis; Trivium; stream cipher

1 引 言

为了发展安全快速的流密码算法,欧洲 ECRYPT(European Network of Excellence for Cryptology)在2004年启动了 eSTREAM[1] 工程,Trivium[2] 是7个最终获选的算法之一.以 eSTREAM 为代表的流密码评选工程的进行,极大地促进了流密码标准的制定进程.因此,作为流密码的杰出代表,Trivium 算法的设计思想及深入研究对于我们具有重要的意义.

Trivium 是由 Cannière 和 Preneel 基于分组密码设计的面向硬件实现的同步流密码算法,密钥和初始化向量规模均为80 bit,内部状态为288 bit,由3个级数分别为93、84和111的非线性反馈移位寄存器(NFSR)互控更新内部状态.该结构设计简单,便于硬件实现.

目前针对 Trivium 的安全性分析主要有代数攻

收稿日期:2012-05-09;最终修改稿收到日期:2012-07-03.本课题得到国家自然科学基金(61202491)及研究课题(2010JY0263-149)资助.
孙文龙,男,1988年生,硕士研究生,主要研究方向为流密码的设计与分析.E-mail: swl_cipher@163.com. **关 杰**,女,1974年生,博士,副教授,主要研究方向为对称密码的设计与分析. **刘建东**,男,1974年生,高级工程师,主要研究方向为通信保密.

击、滑动攻击、立方攻击和猜测决定攻击等. Maximov 和 Biryukov[3] 对 Trivium 进行状态恢复攻击和统计测试，恢复 Trivium 的内部状态需要的计算量为 $c\cdot 2^{83.5}$，其中常数 c 是求解线性方程组的复杂度. Raddum[4] 通过求解一个线性方程组来恢复288 bit 的初始状态，其计算复杂度为 $O(2^{164})$. Priemuth-Schmid 等人[5]针对 Trivium 算法提出了滑动攻击，指出 Trivium 算法有多于 2^{39} 个滑动对. Dinur 和 Shamir[6]针对 767 个时刻的初始化 Trivium 算法实施了立方攻击，所需的计算复杂度为 2^{45} 比特操作. 孙国平等人[7]通过错误注入的方法改变 Trivium 内部状态中的 52 bit，提出了一种基于选择差分的 Trivium 猜测攻击，攻击的计算复杂度为 2^{45}.

Turan 等人[8]对初始化为 288 轮的简化版的 Trivium 算法进行了线性分析，找到了 1 个线性偏差为 2^{-31} 的线性逼近式. 贾艳艳等人[9]运用多线性密码分析理论，对文献[8]的结果进行了改进，提出了一种更有效的区分攻击算法.

本文针对初始化为 288 轮的简化版 Trivium 算法进行线性分析，更正了 Turan 等人给出的关于密钥、初始化向量和密钥流比特的表达式，并给出了当允许选取特殊的密钥和 ***IV*** 时，搜索最佳线性逼近式的算法. 据此算法，找到了 3 个线性偏差为 2^{-25} 的线性逼近式，改进了 Turan 等人提出的线性偏差为 2^{-31} 的线性分析结果.

2　Trivium 算法描述

2.1　符号说明

$\boldsymbol{K}$：80 bit 密钥，$\boldsymbol{K}=(k_1,k_2,\cdots,k_{80})$；

$\boldsymbol{IV}$：80 bit 初始化向量，$\boldsymbol{IV}=(iv_1,iv_2,\cdots,iv_{80})$；

$s_t(i)$：第 t 时刻内部状态的第 i 比特，$1\leq i\leq 288$；

s_t：第 t 时刻 288 bit 内部状态，$s_t=(s_t(1),s_t(2),\cdots,s_t(288))$；

t_i：更新比特，$1\leq i\leq 3$；

z_i：第 i 时刻的输出密钥流比特；

$+$：$GF(2)$上的加法运算；

$\cdot$：$GF(2)$上的乘法运算.

2.2　算法描述

Trivium 算法分为初始化算法和密钥流生成算法两部分：在初始化阶段，系统在 t 时刻使用特定位置的 15 bit 内部状态更新 $t+1$ 时刻的 3 bit 内部状态，然后运行初始化算法 $4\cdot 288$ 个时钟周期，不输出任何密钥流；在密钥流生成阶段，由 3 个移位寄存器的各 2 比特异或生成密钥流. 具体算法用伪代码描述如下：

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

$(s_{94},\cdots,s_{177})\leftarrow(iv_1,\cdots,iv_{80},0,\cdots,0)$,

$(s_{178},\cdots,s_{288})\leftarrow(0,\cdots,0,1,1,1)$,

for $i=1$ to N do

$z_i\leftarrow s_{66}+s_{93}+s_{162}+s_{177}+s_{243}+s_{288}$

$t_1\leftarrow s_{66}+s_{91}\cdot s_{92}+s_{93}+s_{171}$

$t_2\leftarrow s_{162}+s_{175}\cdot s_{176}+s_{177}+s_{264}$

$t_3\leftarrow s_{243}+s_{286}\cdot s_{287}+s_{288}+s_{69}$

$(s_1,s_2,\cdots,s_{80})\leftarrow(t_3,s_1,\cdots,s_{92})$

$(s_{94},s_{95},\cdots,s_{177})\leftarrow(t_1,s_{94},\cdots,s_{176})$

$(s_{178},s_{179},\cdots,s_{288})\leftarrow(t_2,s_{178},\cdots,s_{287})$

end for

3　288 轮 Trivium 的线性分析

3.1　基础知识

线性密码分析[10]是 Matsui 在 1993 年欧洲密码年会上提出的一种对迭代型分组密码算法的已知明文攻击方法，其基本思想是通过寻找分组密码算法的一个有效的线性逼近来破译密码系统.

线性密码分析的方法是寻找一个给定密码算法的具有下列形式的“有效的”线性表达式：

$$P_{[i_1,i_2,\cdots,i_a]}\oplus C_{[j_1,j_2,\cdots,j_b]}=K_{[k_1,k_2,\cdots,k_c]} \tag{1}$$

其中，$i_1,i_2,\cdots,i_a,j_1,j_2,\cdots,j_b$ 和 $k_1,k_2,\cdots,k_c$ 表示固定的比特位置，并且对随机给定的明文 P 和相应的密文 C，等式(1)成立的概率 $p\neq 1/2$，用 $|p-1/2|$ 来刻画等式(1)的有效性，称 $|p-1/2|$ 为线性偏差 ε. 如果 $|p-1/2|$ 是最大的，将对应的线性表达式称为最佳线性逼近式.

针对多轮的分组密码[11]，首先对不同轮的非线性函数进行逼近，然后将各个逼近有效地组合，最终得到有效的线性逼近. 分组密码的线性逼近的概率与每一轮线性逼近的概率都有关，可由下面的堆积引理来计算形如式(1)成立的概率.

引理 1[11]. 设 $X_i(1\leq i\leq n)$ 是独立的随机变量，$Pr(X_i=0)=p_i$，$Pr(X_i=1)=1-p_i$，则

$$Pr(X_1\oplus X_2\oplus\cdots\oplus X_n=0)=\frac{1}{2}+2^{n-1}\cdot\prod_1^n(p_i-1/2) \tag{2}$$

任意一个流密码都可以看作布尔函数 $F_i:F_2^k\times$

$F_2^v \to F_2$，$i=1,2,\cdots$的集合，其中F_i为由k比特密钥和v比特iv生成第i个输出密钥流比特z_i的函数.目前已经有人把针对分组密码提出的线性分析方法用于流密码算法的分析上[8,12]，基本思路如下：

流密码算法的初始化过程一般都是由相同的状态函数迭代一定的圈数实现内部状态变量的更新，为了便于寻找F_i的线性逼近，将初始化阶段分成n轮，其中第i轮有t_i个时钟，对每一轮进行线性逼近，然后将每一轮的线性逼近进行有效组合，最终得到整个密码算法的线性逼近.

由 Matsui 的线性密码分析理论[10]，线性分析的成功率为

$$\gamma = \frac{1}{\sqrt{2\pi}} \int_{-2\sqrt{N}\left|p-\frac{1}{2}\right|}^{\infty} \mathrm{e}^{-x^2/2} \mathrm{d}x \tag{3}$$

3.2 主要结论

3.2.1 修正方程

对于初始化为 288 轮的简化版 Trivium 算法，有下式成立[8]：

$$z_1 = s_{288}(66)+s_{288}(93)+s_{288}(162)+ s_{288}(177)+s_{288}(243)+s_{288}(288) \tag{4}$$

通过迭代，z_1关于第 144 轮内部状态的方程如下所示：

$$\begin{aligned} z_1 = {} & s_{144}(6)+s_{144}(16)\cdot s_{144}(17)+s_{144}(31)\cdot \\ & s_{144}(32)+s_{144}(33)+s_{144}(57)+s_{144}(82)\cdot \\ & s_{144}(83)+s_{144}(84)+s_{144}(96)+s_{144}(97)\cdot \\ & s_{144}(98)+s_{144}(99)+s_{144}(111)+s_{144}(129)+ \\ & s_{144}(142)\cdot s_{144}(143)+s_{144}(144)+s_{144}(150)+ \\ & s_{144}(162)+s_{144}(163)\cdot s_{144}(164)+s_{144}(165)+ \\ & s_{144}(186)+s_{144}(192)+s_{144}(208)\cdot s_{144}(209)+ \\ & s_{144}(210)+s_{144}(231)+s_{144}(235)\cdot s_{144}(236)+ \\ & s_{144}(237)+s_{144}(252) \end{aligned} \tag{5}$$

由堆积引理，式(5)以$2^7 \cdot 0.25^8 = 2^{-9}$的线性偏差逼近下式：

$$\begin{aligned} z_1 = {} & s_{144}(6)+s_{144}(33)+s_{144}(57)+s_{144}(84)+ \\ & s_{144}(96)+s_{144}(99)+s_{144}(111)+s_{144}(129)+ \\ & s_{144}(144)+s_{144}(150)+s_{144}(162)+s_{144}(165)+ \\ & s_{144}(186)+s_{144}(192)+s_{144}(210)+s_{144}(231)+ \\ & s_{144}(237)+s_{144}(252) \end{aligned} \tag{6}$$

对于式(6)，将等式左边的密钥流比特z_1通过迭代用密钥和初始化向量比特表示出来，经过分析我们发现文献[8]附录中 Turan 等人给出的密钥流比特z_1关于密钥和 ***IV*** 比特的表达式有误，下面给出说明：

对于式(6)中第 144 轮的内部状态变量，通过迭代用密钥和 ***IV*** 比特表示出来，在这些表达式中：

(1) 没有出现$s_0(79)$这个变量，因此，z_1关于密钥和 ***IV*** 比特的表达式中不存在二次项$s_0(77)\cdot s_0(79)$；

(2) 用二次项$s_0(76)\cdot s_0(77)$表示的，只有$s_{144}(144)$和$s_{144}(237)$这两项，因此，两项异或后z_1关于密钥和 ***IV*** 比特的表达式中不存在二次项$s_0(76)\cdot s_0(77)$；

(3) 用三次项$s_0(146)\cdot s_0(147)\cdot s_0(148)$和$s_0(146)\cdot s_0(147)\cdot s_0(149)$表示的只有$s_{144}(6)$一项，因此，$z_1$关于密钥和 ***IV*** 比特的表达式中存在三次项$s_0(146)\cdot s_0(147)\cdot s_0(148)$和$s_0(146)\cdot s_0(147)\cdot s_0(149)$；

综合(1)、(2)和(3)的分析，z_1关于密钥和 ***IV*** 比特的表达式中不存在二次项$s_0(76)\cdot s_0(77)$和$s_0(77)\cdot s_0(79)$，而三次项$s_0(146)\cdot s_0(147)\cdot s_0(148)$和$s_0(146)\cdot s_0(147)\cdot s_0(149)$是存在的.

故对于式(6)，文献[8]附录中 Turan 等人的表达式有误，进一步，我们使用 MATLAB 软件编程验证了(1)、(2)和(3)，z_1关于密钥和 ***IV*** 比特的正确表达式如下所示：

$$\begin{aligned} z_1 = {} & 1+s_0(3)+s_0(6)+s_0(15)+s_0(21)+s_0(27)+ \\ & s_0(30)+s_0(39)+s_0(54)+s_0(57)+s_0(67)+ \\ & s_0(68)+s_0(69)+s_0(72)+s_0(96)+s_0(99)+ \\ & s_0(114)+s_0(117)+s_0(123)+s_0(126)+ \\ & s_0(132)+s_0(138)+s_0(144)+s_0(165)+ \\ & s_0(171)+s_0(4)\cdot s_0(5)+s_0(13)\cdot s_0(14)+ \\ & s_0(13)\cdot s_0(41)+s_0(13)\cdot s_0(119)+ \\ & s_0(14)\cdot s_0(40)+s_0(14)\cdot s_0(118)+ \\ & s_0(16)\cdot s_0(17)+s_0(19)\cdot s_0(20)+ \\ & s_0(19)\cdot s_0(47)+s_0(19)\cdot s_0(125)+ \\ & s_0(20)\cdot s_0(46)+s_0(20)\cdot s_0(124)+ \\ & s_0(22)\cdot s_0(23)+s_0(25)\cdot s_0(26)+ \\ & s_0(28)\cdot s_0(29)+s_0(34)\cdot s_0(35)+ \\ & s_0(37)\cdot s_0(38)+s_0(37)\cdot s_0(65)+ \\ & s_0(37)\cdot s_0(143)+s_0(38)\cdot s_0(64)+ \\ & s_0(38)\cdot s_0(142)+s_0(39)\cdot s_0(40)+ \\ & s_0(40)\cdot s_0(119)+s_0(41)\cdot s_0(118)+ \\ & s_0(43)\cdot s_0(44)+s_0(45)\cdot s_0(46)+ \\ & s_0(46)\cdot s_0(125)+s_0(47)\cdot s_0(124)+ \\ & s_0(49)\cdot s_0(50)+s_0(52)\cdot s_0(53)+ \\ & s_0(58)\cdot s_0(59)+s_0(58)\cdot s_0(164)+ \\ & s_0(59)\cdot s_0(163)+s_0(61)\cdot s_0(62)+ \end{aligned}$$

$$
\begin{aligned}
&s_0(63)\cdot s_0(64)+s_0(64)\cdot s_0(65)+\\
&s_0(64)\cdot s_0(143)+s_0(64)\cdot s_0(170)+\\
&s_0(65)\cdot s_0(142)+s_0(65)\cdot s_0(169)+\\
&s_0(67)\cdot s_0(68)+s_0(70)\cdot s_0(71)+\\
&s_0(103)\cdot s_0(104)+s_0(106)\cdot s_0(107)+\\
&s_0(118)\cdot s_0(119)+s_0(124)\cdot s_0(125)+\\
&s_0(127)\cdot s_0(128)+s_0(130)\cdot s_0(131)+\\
&s_0(133)\cdot s_0(149)+s_0(134)\cdot s_0(148)+\\
&s_0(142)\cdot s_0(143)+s_0(147)\cdot s_0(148)+\\
&s_0(151)\cdot s_0(152)+s_0(154)\cdot s_0(155)+\\
&s_0(160)\cdot s_0(161)+s_0(163)\cdot s_0(164)+\\
&s_0(166)\cdot s_0(167)+s_0(13)\cdot s_0(39)\cdot s_0(40)+\\
&s_0(14)\cdot s_0(38)\cdot s_0(39)+\\
&s_0(19)\cdot s_0(45)\cdot s_0(46)+s_0(20)\cdot s_0(44)\cdot\\
&s_0(45)+s_0(37)\cdot s_0(63)\cdot s_0(64)+\\
&s_0(38)\cdot s_0(39)\cdot s_0(40)+s_0(38)\cdot s_0(39)\cdot\\
&s_0(41)+s_0(38)\cdot s_0(39)\cdot s_0(119)+\\
&s_0(38)\cdot s_0(62)\cdot s_0(63)+s_0(39)\cdot s_0(40)\cdot\\
&s_0(118)+s_0(44)\cdot s_0(45)\cdot s_0(46)+\\
&s_0(44)\cdot s_0(45)\cdot s_0(47)+s_0(44)\cdot s_0(45)\cdot\\
&s_0(125)+s_0(45)\cdot s_0(46)\cdot s_0(124)+\\
&s_0(62)\cdot s_0(63)\cdot s_0(64)+s_0(62)\cdot s_0(63)\cdot\\
&s_0(65)+s_0(62)\cdot s_0(63)\cdot s_0(143)+\\
&s_0(63)\cdot s_0(64)\cdot s_0(142)+s_0(133)\cdot s_0(147)\cdot\\
&s_0(148)+s_0(134)\cdot s_0(146)\cdot s_0(147)+\\
&s_0(146)\cdot s_0(147)\cdot s(148)+\\
&s_0(146)\cdot s_0(147)\cdot s_0(149)
\end{aligned}
\tag{7}
$$

3.2.2　最佳线性逼近式的搜索算法

观察可知，式(7)中共有 79 个非线性项：57 个二次项和 22 个三次项，如果令式(7)中所有的非线性项均为 0，则我们找到了该式的一个线性逼近：

$$
\begin{aligned}
z_1=&1+k_3+k_6+k_{15}+k_{21}+k_{27}+k_{30}+k_{39}+k_{54}+\\
&k_{57}+k_{67}+k_{68}+k_{69}+k_{72}+iv_3+iv_6+iv_{21}+\\
&iv_{24}+iv_{30}+iv_{33}+iv_{39}+iv_{45}+iv_{51}+iv_{72}+iv_{78}
\end{aligned}
\tag{8}
$$

由堆积引理，式(8)由式(7)逼近的线性偏差为 $2^{78}\cdot(0.25)^{57}\cdot(0.375)^{22}=2^{-67.13}$.

综合式(5)～(8)，由堆积引理，式(8)成立的线性偏差为 $\varepsilon=2\cdot2^{-9}\cdot2^{-67.13}=2^{-75.13}<2^{-|K|/2}=2^{-40}$ ($|\boldsymbol{K}|$ 表示密钥规模，对 Trivium 而言，密钥规模 $|\boldsymbol{K}|=80$). 但是这个偏差太小，对于线性密码分析没有意义.

此时，可以通过选择特殊的密钥和 ***IV*** 的方法来增大式(8)成立的线性偏差. 设 $\Omega_{\boldsymbol{K}}=\{k_i\mid k_i=0,1\leq i\leq 80\}$ 表示选择为 0 的密钥比特组成的集合，$|\Omega_{\boldsymbol{K}}|$ 表示 $\Omega_{\boldsymbol{K}}$ 的规模；$\Omega_{\boldsymbol{IV}}=\{iv_i\mid iv_i=0,1\leq i\leq 80\}$ 表示选择为 0 的 ***IV*** 比特组成的集合，$|\Omega_{\boldsymbol{IV}}|$ 表示 $\Omega_{\boldsymbol{IV}}$ 的规模；n_1 和 n_2 分别表示选择 $\Omega_{\boldsymbol{K}}$ 和 $\Omega_{\boldsymbol{IV}}$ 之后式(7)中剩余二次项和三次项的数量，则式(8)成立的线性偏差 ε 为

$$
\begin{aligned}
\varepsilon&=2\cdot2^{-9}\cdot2^{(n_1+n_2)-1}\cdot(0.25)^{n_1}\cdot(0.375)^{n_2}\\
&=(0.5)^{n_1+9}\cdot(0.75)^{n_2}
\end{aligned}
\tag{9}
$$

经分析，对于式(7)，选择不同的 $\Omega_{\boldsymbol{K}}$ 和 $\Omega_{\boldsymbol{IV}}$ 将会影响式(7)中二次项数量 n_1 和三次项数量 n_2，进而影响到式(8)成立的线性偏差 ε 的大小，但是文献[8]和文献[9]中并没有给出 $\Omega_{\boldsymbol{K}}$ 和 $\Omega_{\boldsymbol{IV}}$ 的选择准则. 下面，我们就这一问题给出搜索最佳线性逼近式的算法 1.

算法 1.　最佳线性逼近式的搜索算法.

$\Omega_{\boldsymbol{K}}$ 和 $\Omega_{\boldsymbol{IV}}$ 初始化为空集；

输入：$\Omega_{\boldsymbol{K}}$ 选择的规模 $n_{\boldsymbol{K}}$，$\Omega_{\boldsymbol{IV}}$ 选择的规模 $n_{\boldsymbol{IV}}$；

1. 集合 Ω 和 Π 初始化为空集，统计式(7)的所有非线性项中各个比特的频次，将频次最大的比特存入到集合 Ω 中；

2.1. 若 $|\Omega|=1$，判断 Ω 中比特类型，若为密钥则存入 $\Omega_{\boldsymbol{K}}$ 中，否则存入 $\Omega_{\boldsymbol{IV}}$ 中；

2.2. 若 $|\Omega|\geq2$，统计 Ω 中各个比特涉及到的二次项的频次，将频次最大的比特存入到集合 Π 中；

2.2.1. 若 $|\Pi|=1$，判断 Π 中比特类型，若为密钥则存入 $\Omega_{\boldsymbol{K}}$ 中，否则存入 $\Omega_{\boldsymbol{IV}}$ 中；

2.2.2. 若 $|\Pi|\geq2$，则任选 Π 中一个比特，判断该比特类型，若为密钥则存入 $\Omega_{\boldsymbol{K}}$ 中，否则存入 $\Omega_{\boldsymbol{IV}}$ 中；

3. 根据选择的 $\Omega_{\boldsymbol{K}}$ 和 $\Omega_{\boldsymbol{IV}}$，重新计算式(7)；

4.

if $((|\Omega_{\boldsymbol{K}}|<n_{\boldsymbol{K}})\&\&(|\Omega_{\boldsymbol{IV}}|<n_{\boldsymbol{IV}}))$

　返回步 1；

else if $((|\Omega_{\boldsymbol{K}}|=n_{\boldsymbol{K}})\&\&(|\Omega_{\boldsymbol{IV}}|\neq n_{\boldsymbol{IV}}))$

　则停止搜索密钥比特，返回步 1，继续搜索满足条件的 ***IV*** 比特；

else if $((|\Omega_{\boldsymbol{K}}|\neq n_{\boldsymbol{K}})\&\&(|\Omega_{\boldsymbol{IV}}|=n_{\boldsymbol{IV}}))$

　则停止搜索 ***IV*** 比特，返回步 1，继续搜索满足条件的密钥比特；

else if $((|\Omega_{\boldsymbol{K}}|=n_{\boldsymbol{K}})\&\&(|\Omega_{\boldsymbol{IV}}|=n_{\boldsymbol{IV}}))$

输出：$((\Omega_{\boldsymbol{K}},\Omega_{\boldsymbol{IV}}),\varepsilon)$.

定理 1.　当给定选择为 0 的密钥和 ***IV*** 比特的规模时，算法 1 搜索到的是最佳线性逼近式. 即设定算法 1 搜索到的密钥和 ***IV*** 比特为 0 后，式(8)成立的线性偏差 ε 最大.

证明.　当 $|\Omega|=1$，将该比特设定为 0，则消掉

的非线性项数量最多，显然式(8)成立的偏差最大.当$|\Omega|\geq 2$，不妨设$\Omega=\{a,b\}$，设比特a和b的频次为n，分别统计比特a和b涉及到的二次项频次，记为m_1和m_2(不妨设$m_1>m_2$)，设未选择a或者b为0之前，式(7)中有r个二次项、t个三次项.若把a设定为0，式(8)成立的线性偏差为$\varepsilon_a=(0.5)^{r-m_1+9}\cdot(0.75)^{t-(n-m_1)}$；若把$b$设定为0，式(8)成立的线性偏差为$\varepsilon_b=(0.5)^{r-m_2+9}\cdot(0.75)^{t-(n-m_2)}$，显然$\varepsilon_a>\varepsilon_b$，因此欲使偏差最大，应选择$a=0$.

算法1就是按照这样的准则搜索满足要求的密钥和***IV***比特的.因此，当设定算法1搜索到的密钥和***IV***比特为0后，式(8)成立的线性偏差最大，即算法1搜索的是最佳线性逼近式. 证毕.

此外，我们给出的最佳线性逼近式的搜索算法具有普适性，可以把它进一步推广，用于其它密码算法的线性分析中.

根据算法1，表1给出了不同的$|\Omega_K|$和$|\Omega_{IV}|$对应的最大偏差.

表1 ($|\Omega_K|$,$|\Omega_{IV}|$)对应的最大偏差ε

$\|\Omega_{IV}\|$	$\|\Omega_K\|$										
	0	1	2	3	4	5	6	7	8	9	10
2	—	—	—	—	—	—	—	—	—	$2^{-38.41}$	$2^{-37.41}$
3	—	—	—	—	—	—	—	$2^{-39.41}$	$2^{-37.41}$	$2^{-35.41}$	$2^{-34.41}$
4	—	—	—	—	—	—	—	2^{-37}	2^{-35}	2^{-33}	2^{-32}
5	—	—	—	—	—	—	2^{-38}	2^{-35}	2^{-33}	2^{-31}	2^{-30}
6	—	—	—	—	—	2^{-39}	2^{-36}	2^{-34}	2^{-32}	2^{-30}	2^{-29}
7	—	—	—	—	—	2^{-38}	2^{-35}	2^{-33}	2^{-31}	2^{-29}	2^{-28}
8	—	—	—	—	—	2^{-37}	2^{-34}	2^{-32}	2^{-30}	2^{-28}	2^{-27}
9	—	—	—	—	$2^{-39.41}$	2^{-36}	2^{-33}	2^{-31}	2^{-29}	2^{-27}	2^{-26}
10	—	—	—	—	$2^{-38.41}$	2^{-35}	2^{-32}	2^{-30}	2^{-28}	2^{-26}	2^{-25}
11	—	—	—	$2^{-39.41}$	$2^{-37.41}$	2^{-34}	2^{-31}	2^{-29}	2^{-27}	2^{-25}	2^{-24}
12	—	—	—	$2^{-38.41}$	$2^{-36.41}$	2^{-33}	2^{-30}	2^{-28}	2^{-26}	2^{-24}	2^{-23}
13	—	—	—	$2^{-37.41}$	$2^{-35.41}$	2^{-32}	2^{-29}	2^{-27}	2^{-25}	2^{-23}	2^{-22}
14	—	—	$2^{-39.49}$	$2^{-36.41}$	$2^{-34.41}$	2^{-31}	2^{-28}	2^{-26}	2^{-24}	2^{-22}	2^{-21}
15	—	—	$2^{-38.49}$	$2^{-35.41}$	$2^{-33.41}$	2^{-30}	2^{-27}	2^{-25}	2^{-23}	2^{-21}	2^{-20}
16	—	—	$2^{-37.49}$	$2^{-34.41}$	$2^{-32.41}$	2^{-29}	2^{-26}	2^{-24}	2^{-22}	2^{-21}	2^{-20}
17	—	$2^{-39.15}$	$2^{-36.49}$	$2^{-33.41}$	$2^{-31.41}$	2^{-29}	2^{-25}	2^{-24}	2^{-22}	2^{-21}	2^{-20}
18	—	$2^{-38.15}$	$2^{-35.49}$	$2^{-32.41}$	$2^{-30.41}$	2^{-29}	2^{-25}	2^{-24}	2^{-22}	2^{-21}	2^{-20}
19	—	$2^{-37.15}$	$2^{-34.49}$	$2^{-31.41}$	$2^{-29.41}$	2^{-29}	2^{-25}	2^{-24}	2^{-22}	2^{-21}	2^{-20}
20	$2^{-39.98}$	$2^{-36.15}$	$2^{-33.49}$	$2^{-30.41}$	$2^{-28.41}$	2^{-29}	2^{-25}	2^{-24}	2^{-22}	2^{-21}	2^{-20}

注：“—”表示在$|\Omega_K|$和$|\Omega_{IV}|$取相应值时的偏差$\varepsilon<2^{-40}$.

根据表1的结论，攻击者可以依据攻击能力的大小利用算法1选择最佳的Ω_K和Ω_{IV}.例如，若攻击者仅能够选择***IV***，则可以得到线性逼近式的最大线性偏差为$2^{-39.98}$；若攻击者能够选择10个密钥比特和15个***IV***比特，则可以得到线性逼近式的最大偏差为2^{-20}.附录1中给出了利用算法1搜索到的最佳密钥和***IV***比特的部分实例.

特别地，根据算法1，如果选择如下10个特定的密钥比特和10个特定的***IV***比特时，有以下结论.

表2 ($|\Omega_K|=10$,$|\Omega_{IV}|=10$)对应的最佳(Ω_K,Ω_{IV})及ε

Ω_{IV}	Ω_K		线性偏差ε
$iv_{25}iv_{31}iv_{40}iv_{50}iv_{54}$	$k_{13}k_{19}k_{38}k_{40}$	k_5	2^{-25}
$iv_{58}iv_{62}iv_{67}iv_{70}iv_{73}$	$k_{45}k_{46}k_{58}k_{64}k_{65}$	k_{67}	2^{-25}
		k_{68}	2^{-25}

根据表2中Ω_K的不同选择，我们找到了如下3个线性逼近式，其线性偏差均为2^{-25}：

$$z_1=1+k_3+k_6+k_{15}+k_{21}+k_{27}+k_{30}+k_{39}+k_{54}+k_{57}+k_{68}+k_{69}+k_{72}+iv_3+iv_6+iv_{21}+iv_{24}+iv_{30}+iv_{33}+iv_{39}+iv_{45}+iv_{51}+iv_{72}+iv_{78};$$

$$z_1=1+k_3+k_6+k_{15}+k_{21}+k_{27}+k_{30}+k_{39}+k_{54}+k_{57}+k_{67}+k_{69}+k_{72}+iv_3+iv_6+iv_{21}+iv_{24}+iv_{30}+iv_{33}+iv_{39}+iv_{45}+iv_{51}+iv_{72}+iv_{78};$$

$$z_1=1+k_3+k_6+k_{15}+k_{21}+k_{27}+k_{30}+k_{39}+k_{54}+k_{57}+k_{67}+k_{68}+k_{69}+k_{72}+iv_3+iv_6+iv_{21}+iv_{24}+iv_{30}+iv_{33}+iv_{39}+iv_{45}+iv_{51}+iv_{72}+iv_{78}.$$

特别地，如果选择10个特定的密钥比特和13个特定的***IV***比特时，我们可以找到如上的3个线性逼近式，其线性偏差均为2^{-22}.

3.2.3 结果对比

文献[8]中Turan等人选择规模为$|\Omega_K|=10$和$|\Omega_{IV}|=10$的Ω_K和Ω_{IV}，找到了1个线性偏差为2^{-31}的线性逼近式；文献[9]中贾艳艳等人在文献[8]的

基础上，选择规模为 $|\Omega_K|=10$ 和 $|\Omega_{IV}|=13$ 的 Ω_K 和 Ω_{IV}，找到了另一个线性偏差为 2^{-31} 的线性逼近式；本文选择规模为 $|\Omega_K|=10$ 和 $|\Omega_{IV}|=10$ 的 Ω_K 和 Ω_{IV}，找到了 3 个具有相同线性偏差 2^{-25} 的线性逼近式；另外，若选取规模为 $|\Omega_K|=10$ 和 $|\Omega_{IV}|=13$ 的 Ω_K 和 Ω_{IV}，可得到 3 个具有相同线性偏差 2^{-22} 的线性逼近式.

因此，本文的分析结果优于 Turan 和贾艳艳等人的结果，对比如表 3 所示.

表 3　结果对比

	$(\|\Omega_K\|,\|\Omega_{IV}\|)$	线性偏差 ε	线性逼近个数	数据量 N	成功率/%
Turan 等人[8]的结果	(10,10)	2^{-31}	1	2^{62}	97.77
本文的结果	(10,10)	2^{-25}	3	2^{50}	97.77
贾艳艳等人[9]的结果	(10,13)	2^{-31}	2	2^{61}	97.77
本文的结果	(10,13)	2^{-22}	3	2^{44}	97.77

4　结束语

本文针对初始化为 288 轮的简化版 Trivium 算法进行了线性分析，更正了 Turan 等人给出的关于密钥、初始化向量和密钥流比特的表达式，并给出了当允许选取特殊的密钥和 ***IV*** 时，搜索最佳线性逼近式的算法. 据此算法，找到了 3 个线性偏差为 2^{-25} 的线性逼近式，改进了 Turan 等人提出的线性偏差为 2^{-31} 的线性分析结果. 此外，该搜索算法可以进一步推广到其他密码算法的线性分析中. 如何对更多轮的 Trivium 算法进行线性分析是值得进一步研究的问题.

参　考　文　献

[1] eSTREAM. The ECRYPT stream cipher project. http://www.ecrypt.eu.org/stream/

[2] De Cannière C, Preneel B. Trivium specifications. http://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf

[3] Maximov A, Biryukov A. Two trivial attacks on Trivium//Proceedings of the Workshop on The State of the Art of Stream Ciphers (SASC2007). Bochum, 2007: 1-16

[4] Raddum H. Cryptanalytic results on Trivium. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/039, 2006

[5] Priemuth-Schmid D, Biryukov A. Slid pairs in salsa20 and Trivium//Proceedings of the INDOCRYPT 2008 Proceedings. Lecture Notes in Computer Science 5365. Berlin, 2008: 1-14

[6] Dinur I, Shamir A. Cube attacks on tweakable black box polynomials. Cryptology ePrint Archive, Report 2008/385, 2008

[7] Sun Guo-Ping, Hu Yu-Pu, Bai Sheng-Jiang. Guess attack on Trivium based on chosen differential. Computer Engineering, 2010, 36(9): 129-130(in Chinese)
(孙国平，胡予濮，白生江. 基于选择差分的 Trivium 猜测攻击. 计算机工程，2010，36(9)：129-130)

[8] Turan M S, Kara O. Linear approximations for 2-round Trivium//Proceedings of the Workshop on the State of the Art of Stream Cipher (SASC2007). Bochum, 2007: 22-31

[9] Jia Yan-Yan, Hu Yu-Pu, Yang Wen-Feng, Gao Jun-Tao. Linear cryptanalysis of 2-round Trivium with multiple approximations. Journal of Electronics & Information Technology, 2011, 33(1): 223-227(in Chinese)
(贾艳艳，胡予濮，杨文峰，高俊涛. 2 轮 Trivium 的多线性密码分析. 电子与信息学报，2011，33(1)：223-227)

[10] Matsui M. Linear cryptanalysis method for DES cipher//Proceedings of the Advances in Cryptology-EUROCRYPT'93. Lecture Notes in Computer Science 765. Lofthus, 1993: 386-397

[11] Wu Wen-Ling, Feng Deng-Guo, Zhang Wen-Tao. Design and analysis of block cipher. 2nd Edition. Beijing: Tsinghua University Press, 2009(in Chinese)
(吴文玲，冯登国，张文涛. 分组密码的设计与分析. 第 2 版. 北京：清华大学出版社，2009)

[12] Golić J D, Bagini V, Morgari G. Linear cryptanalysis of Bluetooth stream cipher//Proceedings of the Advances in Cryptology-EUROCRYPT 2002. Amsterdam, 2002: 238-255

SUN Wen-Long, born in 1988, master. His research interests focus on the design and analysis of stream ciphers.

GUAN Jie, born in 1974, Ph. D., associate professor, master supervisor. Her research interests include design and analysis of symmetric ciphers.

LIU Jian-Dong, born in 1974, senior engineer. His research interest is communication security.

附录 1. 部分实例.

Ω_K		Ω_{IV}		线性偏差 ε
Ω_K	Ω_{IV}	Ω_K	Ω_{IV}	
0	$\varnothing$	20	$iv_{11}iv_{13}iv_{25}iv_{26}iv_{31}iv_{32}iv_{35}iv_{38}iv_{40}iv_{41}iv_{49}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}iv_{76}iv_{77}$	$2^{-39.98}$
1	k_{64}	20	$iv_{11}iv_{13}iv_{25}iv_{26}iv_{31}iv_{32}iv_{35}iv_{38}iv_{40}iv_{41}iv_{49}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}iv_{76}$	$2^{-36.15}$
2	$k_{38}\ k_{64}$	20	$iv_{11}iv_{13}iv_{25}iv_{26}iv_{31}iv_{32}iv_{35}iv_{38}iv_{40}iv_{41}\ iv_{49}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}iv_{76}$	$2^{-33.49}$
3	$k_{38}k_{45}k_{64}$	20	$iv_{11}iv_{13}iv_{25}iv_{26}iv_{31}iv_{32}iv_{35}iv_{38}iv_{40}iv_{41}iv_{49}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}iv_{76}$	$2^{-30.41}$
4	$k_{38}\ k_{40}k_{45}k_{64}$	20	$iv_{11}iv_{13}iv_{25}iv_{26}iv_{31}iv_{32}iv_{35}iv_{38}iv_{40}iv_{41}iv_{49}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}iv_{76}$	$2^{-28.41}$
5	$k_{38}\ k_{40}k_{45}k_{64}k_{65}$	17	$iv_{11}iv_{13}iv_{25}iv_{31}iv_{32}iv_{35}iv_{38}iv_{40}iv_{41}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}$	2^{-28}
6	$k_{13}k_{38}k_{40}\ k_{45}k_{64}k_{65}$	17	$iv_{11}iv_{13}iv_{25}iv_{31}iv_{32}iv_{35}iv_{38}iv_{40}\ iv_{41}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}$	2^{-25}
7	$k_{13}k_{19}k_{38}k_{40}\ k_{45}k_{64}k_{65}$	16	$iv_{11}iv_{13}iv_{25}iv_{31}iv_{35}iv_{38}iv_{40}\ iv_{41}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}$	2^{-24}
8	$k_{13}k_{19}k_{38}k_{40}k_{45}\ k_{46}k_{64}k_{65}$	16	$iv_{11}iv_{13}iv_{25}iv_{31}iv_{35}iv_{38}iv_{40}\ iv_{41}\ iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{71}iv_{73}$	2^{-22}
9	$k_{13}k_{19}k_{38}\ k_{40}k_{45}k_{46}\ k_{58}k_{64}k_{65}$	15	$iv_{11}iv_{13}iv_{25}iv_{31}iv_{35}iv_{38}iv_{40}\ iv_{41}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{73}$	2^{-21}
10	$k_{13}\ k_{19}k_{26}k_{38}k_{40}k_{45}k_{46}\ k_{58}k_{64}k_{65}$	15	$iv_{11}iv_{13}iv_{25}iv_{31}iv_{35}iv_{38}iv_{40}\ iv_{41}iv_{50}iv_{54}iv_{58}iv_{62}iv_{67}iv_{70}iv_{73}$	2^{-20}

注:"$\varnothing$"表示空集.

Background

The initialization processes of stream ciphers are our main research areas, due to the secure faults often existing in this stage. With trying all kinds of attacks on the ciphers, the secure faults in the initialization process are hoped to found. Based on our analysis, some secure suggestions are hoped to presented, to improve the design and to enhance the security of ciphers.

For some ciphers, such as SNOW2.0, Trivium, Py-family, Salsa20 and so on, our analysis results have been published. Stream ciphers SNOW2.0, Trivium, Py and Salsa20 are all submitted to the eSTREAM project. We point out that SNOW2.0 can't be against the Guess and Determine Attack, that some differential chains with high possibility in Trivium can be found by a differential cryptanalysis based on automatic deduction, that improved related-key attacks on the Py-family of stream ciphers are presented by extending the least significant bitwise distinguisher to a word-based distinguisher, and that the differential algebraic analysis techniques can obtain the best result on 5-round Salsa20.

In general, the results above are based on our careful analysis on the initialization process of these ciphers with trying all kinds of attacks. So it is very important to analyze the initialization for ensuring the security of stream ciphers.

In this paper, we apply linear cryptanalysis to the simplified Trivium with the initialization of 288 rounds. Trivium is a hardware-oriented stream cipher designed in 2005 by Cannière and Preneel, and has been chosen as one of the final ciphers by eSTREAM project. It is well known that Trivium is on the edge of low cost and compactness, with a simple and elegant structure. Although Trivium has attracted a lot of interest, there are no any standard attacks faster than the exhaustive attack, except for side-channel attacks. For the simplified Trivium, Turan et al found one linear approximation with bias 2^{-31}. In this paper, the equation, which involves the key bits, initial vector bits and the first key stream bit in linear approximations of Turan, is corrected. In addition, when special Key bits and IV bits are allowed to be chosen, an algorithm to search the best linear approximations is presented. Based on this algorithm, 3 linear approximations with the same linear bias 2^{-25} are found, which is better than Turan's, and our results are the best so far.