简化 SIMON 类算法的立方测试与分析*

万刘蝉[1]，韦永壮[1,2]

（1. 桂林电子科技大学 认知无线电与信息处理省部共建教育部重点实验室，广西 桂林 541004；2. 中国科学院信息工程研究所 信息安全国家重点实验室，北京 100190）

摘　要： 针对轻量级分组密码算法 SIMON 的安全性分析，对 SIMON32/64 算法抵抗立方攻击的能力和算法内部结构对密钥比特的混淆和扩散性能力进行了评估。基于 SIMON 类算法的密钥编排特点和轮函数结构，结合立方分析的基本思想，利用 FPGA 测试平台设计了一个 SIMON32/64 的立方攻击和密钥中比特检测算法。攻击结果表明：在立方变元取 6 维至 24 维时，对于 7 轮 SIMON32/64 算法，通过立方攻击能够直接恢复 47 bit 密钥，攻击时间复杂度约为 $2^{18.08}$；对于 8 轮 SIMON32/64 算法，能够直接恢复 39 bit 密钥，攻击时间复杂度约为 $2^{25.00}$。对于 10 轮，11 轮 SIMON32/64 算法，通过立方测试均能够捕获到密钥中比特。

关键词： SIMON32/64 密码算法；密钥中比特；立方测试；立方攻击；FPGA

中图分类号： TP309.7　　**文献标志码：** A　　**文章编号：** 1001-3695(2017)01-0243-05

doi:10.3969/j.issn.1001-3695.2017.01.055

Cube test and analysis for reduced SIMON family of block ciphers

Wan Liuchan[1], Wei Yongzhuang[1,2]

(1. *Key Laboratory of Cognitive Radio & Information Processing for Ministry of Education*, *Guilin University of Electronic Technology*, *Guilin Guangxi* 541004, *China*; 2. *State Key Laboratory of Information Security*, *Institute of Information Engineering*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*)

Abstract: Focusing on the safety analysis of the SIMON lightweight block cipher, this paper proposed a method on this algorithm against the cube attack and cube neutral-bit detection. Using the FPGA test platform, this paper proposed a SIMON32/64 cube attack and cube neutral-bit detection algorithm by combining the basic idea of cube analysis. Simulation experiments show that when the cube variables are in the range of 6 to 20 dimensions, refering to 7-round SIMON32/64 algorithm, the algorithm can recover 47 bit key directly with time complexity of $2^{18.08}$ by using cube attack. Moreover, it can recover 39 bit key on 8-round SIMON32/64 algorithm with time complexity of $2^{25.00}$ by using cube attack. With regard to 10-round, 11-round SIMON32/64 algorithm, it can capture neutral secret key bits by using cube test.

Key words: SIMON32/64 cryptographic algorithms; key neutral-bit; cube test; cube attack; FPGA

0　引言

SIMON[1]类算法是在 2013 年由美国国家安全局提出的一类轻量级分组密码算法。该算法内存占用少、安全性高且效率方面要优于大多数现有的轻量级分组密码。此外，SIMON 类算法设计非常简单和紧凑，结构上采用 Feistal 结构，且只使用了适合硬件实现的基本算术运算，如模 2 加、异或、移位和旋转，因此该算法在硬件方面有较为出色的表现。鉴于 SIMON 的重要性和广泛使用性，SIMON 的安全性一直备密码学界关注，目前出现了大量针对 SIMON 算法族的密码分析，包括差分分析[2~6]、线性分析[2,7]、差分故障分析[8]。目前最好的结果为文献[6]的差分分析，攻击轮数为 18 轮，时间复杂度为 $2^{46.00}$，数据复杂度为 $2^{31.20}$。

立方攻击是 Dinur 等人[9]于 2008 年提出的一种新型密码分析方法。其核心思想是攻击者在无法知道密码系统的加密算法和内部结构（即黑盒）的情况下，仅通过观察算法的输入和输出，建立一定数量线性关系的方程。随后 Aumasson 等人[10]将立方攻击的思想推广为立方测试。立方测试的目的不是直接恢复密钥，而是通过对密码性能的测试，用于构建区分器和检测密码系统的非随机性。目前，立方攻击在分组密码、序列密码和 hash 函数上都有广泛的应用，如文献[11]中利用立方攻击恢复了 55 轮 Katan32 算法的 47 bit 密钥；文献[9]中利用立方攻击找到了初始化过程空跑 735 拍的 Trivium 的 52 个关于密钥的线性表达式；文献[12]中，Mroczkowski 等人首次将二次检测应用于对 709 轮 Trivium 算法的立方攻击中，可以恢复全部 80 bit 密钥，其中有 39 bit 密钥是在二次检测中恢复的；文献[13]中，Dinur 等人又提出了一种改进的立方攻击方法，称为动态立方攻击，运用这种方法在理论上完全破解了 Grain-128 算法，等等。

但是一个重要的问题是：SIMON 算法能否抵抗立方攻击，它的密钥比特与轮函数的迭代组合是否最佳？特别是，如何检测 SIMON 算法内部结构对密钥比特的混淆和扩散性程度？本

收稿日期： 2015-10-16；**修回日期：** 2015-12-28　　**基金项目：** 国家自然科学基金项目（61572148）；广西自然科学基金项目（2015GXNSFGA139007）；广西高等学校优秀中青年骨干教师培养工程（第二期）；桂林电子科技大学研究生创新项目（YJCXS201525）

作者简介： 万刘蝉（1990-），男，江西南昌人，~~硕士研究生~~，主要研究方向为信息安全（wan09042122@163.com）；韦永壮（1976-），男（壮族），广西田阳人，教授，博士，主要研究方向为信息安全.

文基于 SIMON 算法内部结构及密钥编排特点，分别结合立方攻击和立方测试的基本思想，在 FPGA 平台上对简化轮 SIMON32/64 算法进行了测试与分析。立方攻击方面：对于 7 轮 SIMON32/64 算法，通过立方攻击能够直接恢复 47 bit 密钥，攻击时间复杂度约为 $2^{18.08}$；对于 8 轮 SIMON32/64 算法，能够直接恢复 39 bit 密钥，攻击时间复杂度约为 $2^{25.00}$。立方测试方面：发现在取 6 ~ 24 维立方变元的情况下，对简化到 10 轮、11 轮的 SIMON32/64 算法均能捕获到密钥中比特。

本文介绍了 SIMON 类算法和立方分析，描述了 SIMON32/64 算法的立方攻击和立方测试以及在 FPGA 上高效率实现的细节。

1　SIMON 类算法简介[1]

SIMON 算法族整体采用了 Feistel 结构，分组长度为 $2n$（$n = 16, 24, 32, 48, 64$）、密钥长度为 mn，根据不同的密钥规模，m 取值为 2、3 或 4，称为 SIMON2n/mn。轮函数仅采用循环移位、与和异或这样简单的操作。SIMON 算法族的单轮变换结构如图 1 所示。其中“ $\lll i$ ”表示循环左移 i 位，“&”表示与操作。

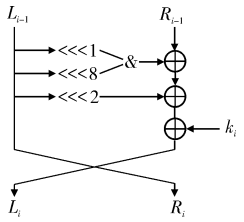

图 1　SIMON 算法族的单轮变换结构

由于篇幅有限，密钥编排可参见文献[1]。

2　立方分析简介

立方攻击是一种新型的代数攻击方法，旨在敌手无法得知密码算法的内部结构，仅仅能够观察密码算法的输入和输出的情况下，寻找密码算法的线性方程以恢复密钥或进行区分攻击，它吸收了饱和攻击及高阶差分分析的思想，该攻击方法主要基于下述定义：

定理[9] 1　设 $f(x_1, x_2, \cdots, x_n)$ 为含有 n 个变量的布尔函数，I 为集合 $S = \{x_1, x_2, \cdots, x_n\}$ 的真子集，则布尔函数可表示为

$$f(x_1, x_2, \cdots, x_n) = p(I) \cdot (p(x_j \mid j \in S - I)) \oplus R(x_j \mid j \in S)$$

其中：$p(I) = (\prod_{i \in I} x_i)$；$p(\cdot)$ 和 $R(\cdot)$ 均为代数标准型（ANF）表示的布尔函数，$p(\cdot)$ 函数中的变量均取自集合 I 的补集 $S - I$，$R(\cdot)$ 函数中的每一项均不含 I 中的全部变量，那么

$$\sum_I f(x_1, x_2, \cdots, x_n) = \sum_I p(I) \cdot (p(x_j \mid j \in S - I)) \oplus R(x_j \mid j \in S) = p(x_j \mid j \in S - I)$$

在分组密码中，明文为公开变量，密钥为未知变量，如果集合 I 中的变量均为公开变量，即明文，$p(\cdot)$ 为非常量的线性表达式，那么根据

$$\sum_I p(I) \cdot (p(x_j \mid j \in S - I)) \oplus R(x_j \mid j \in S)$$

就可以得到一个关于未知变量 $p(x_j \mid j \in S - I)$ 的线性方程，并称 $p(I)$ 为极大项，$p(\cdot)$ 为超级多项式。

立方攻击分为两个阶段：a）预处理阶段和密钥恢复阶段。立方攻击的有效性很大程度依赖了超级多项式 $p(\cdot)$ 能否通过线性检测，通过有目的地选择 I，去捕获尽量含线性关系的 $p(\cdot)$；b）建立多个线性方程，利用高斯消元法求解方程，进而恢复主密钥。

立方测试是 Aumasson 等人在立方攻击的思想上进一步提出的一种新的密码分析方法。以下是立方测试检测密码算法密钥中比特的基本思想。

定理[10] 2　对于任意一个密码算法 $E(K, IV)$，对于给定输出位，如果存在一些密钥比特对输出结果不产生任何影响，则称这些比特为密钥中比特。

举例如下：设加密函数

$$E(k_1, k_2, k_3, k_4, v_1, v_2, v_3, v_4) = v_1 v_2 k_1 \oplus v_1 v_2 k_2 \oplus v_3 v_4 k_3 \oplus v_3 v_4 k_4 \oplus 1$$

令 $I = \{v_3, v_4\}$，则有等式

$$E(k_1, k_2, k_3, k_4, v_1, v_2, v_3, v_4) = v_1 v_2 (k_1 \oplus k_2) \oplus v_3 v_4 (k_3 \oplus k_4) \oplus 1$$

遍历 I 中所有可能的值，而 v_1、v_2 取值不变，对加密函数进行累加可得

$$\sum_I E(k_1, k_2, k_3, k_4, v_1, v_2, v_3, v_4) = \sum_I v_1 v_2 (k_1 \oplus k_2) \oplus \sum_I v_3 v_4 (k_3 \oplus k_4) \oplus 1 = k_3 \oplus k_4$$

则 k_1、k_2 为密钥中比特。

混淆和扩散是香农提出的一般密码设计基本原则，其主要目的是使密文和明文统计特性之间的关系复杂化。因此，密码输出位的密钥中比特数量是度量算法安全强度的一个重要参数。

3　对 SIMON32/64 算法的立方分析算法设计

3.1　对 SIMON32/64 算法的立方攻击算法设计

立方攻击的关键在于攻击的预处理阶段，也就是找到足够多的极大项 $p(I)$ 和超级多项式 $p(\cdot)$。本文根据 SIMON32/64 算法的结构特点并结合立方攻击的基本思想，设计了如下步骤对 SIMON32/64 进行立方攻击，如图 1 所示。

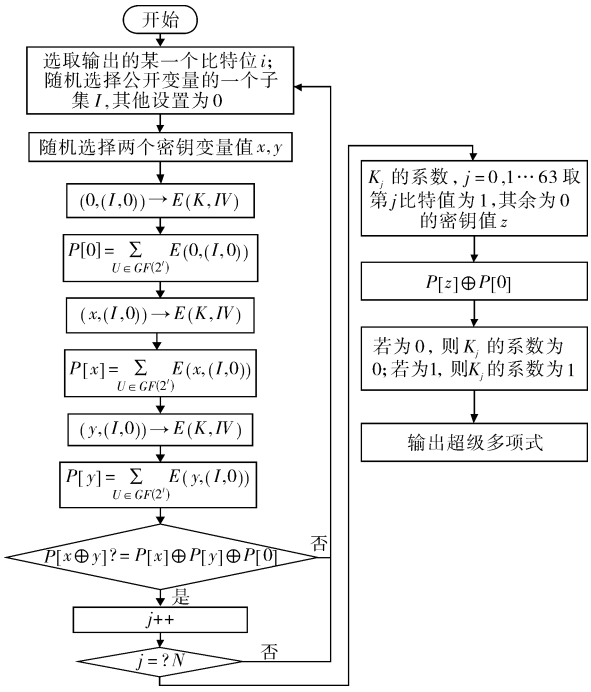

图 2　SIMON32/64 算法的立方攻击流程图

分析说明如下：假定加密函数 SIMON32/64 为 $E(K, IV)$。其中，公开变量集合为 $IV = (v_1, v_2, \cdots, v_{32})$，主密钥集合 SIMON32/64 为 $K = (k_1, k_2, \cdots, k_{64})$。

a)选定输出的某一比特位 i,p 表示该输出比特位的多项式。

b)任意选择公开变量集合的子集 $U=(v_1,v_2,\cdots,v_n)$ 作为 n 维立方变元集合(n 取 6 ~ 24),使剩余的公开变量都固定不变(如全取 0)。

c)随机生成两个主密钥串 $\{K_1,K_2\}$,检验 $p(0)\oplus p(K_1)\oplus p(K_2)=p(0\oplus K_1\oplus K_2)$ 是否成立。其中 $p(x)$ 的计算方法为:对密钥 x,遍历 $U=(v_1,v_2,\cdots,v_n)$ 的所有可能值,并将对应输出值进行累加,即:$p(x)=\sum_{U\in GF(2^n)}E(x,(U,0))$。

d)若 c)中的等式成立,则继续执行 c)检验等式,若连续 N 次都成立,则输出 U;否则执行 b)。

说明:此处 N 取 200,对通过的等式再进行连续 10 000 次检测,若通过,则认为此多项式以极大概率为线性的,否则丢弃。

e)若以上步骤均通过,则可确定超级多项式 $p(x)$ 的具体表达式。方法如下:

(a)首先确定自由项 $p(0)$,若为 0,则自由项为 0,反之为 1。

(b)确定超级多项式 $p(x)$ 中的有关密钥的系数,对 x 中的 $k_i(i=0,1,\cdots 63)$,取第 i 比特为 1,其余取 0,计算 $p(x)\oplus p(0)$,如果为 1,则 k_i 的系数为 1,反之为 0。

3.2　对 SIMON32/64 算法的立方中比特检测算法设计

根据 SIMON32/64 算法的结构特点并结合密钥中比特检测的基本思想,设计了如下步骤检测 SIMON32/64 的密钥中比特,如图 3 所示。

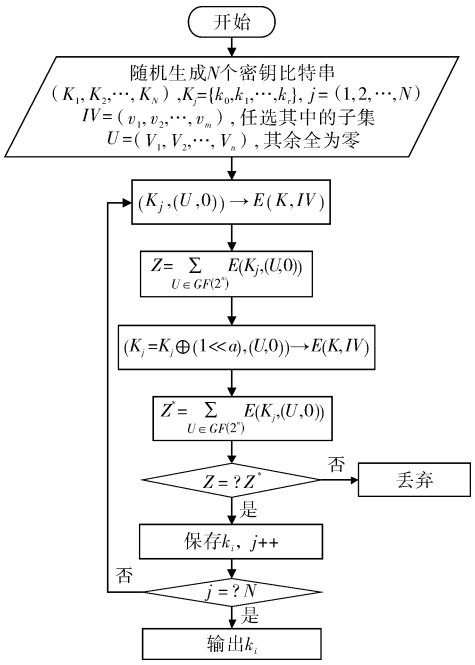

图 3　SIMON32/64 算法的密钥中比特检测流程图

分析说明如下:假定加密函数 SIMON32/64 为 $E(K,IV)$。其中,公开变量集合为 $IV=(v_1,v_2,\cdots,v_{32})$,主密钥集合 SIMON32/64 为 $K=(k_1,k_2,\cdots,k_{64})$。

a)任意选择公开变量集合的子集 $U=(v_1,v_2,\cdots,v_n)$ 作为 n 维立方变元集合(n 取 6 ~ 24),使剩余的公开变量都固定不变(如全取 0),随机生成 40 个主密钥串 $\{K_1,K_2,\cdots,K_{40}\}$。

b)对密钥 K_1,跑遍 $U=(v_1,v_2,\cdots,v_n)$ 的所有可能值,并将对应输出值进行累加,即 $Z_1=\sum_{U\in GF(2^n)}E(K_1,(U,0))$。

c)对主密钥 $K_1=(k_1,k_2,\cdots,k_m)$(SIMON32/64 算法中 m 取 64)进行如下处理,每次只改变一个密钥比特的值,即 $k_i^*=k_i\oplus 1$,$i=(1,2,\cdots,m)$,将改变后的主密钥 k_i^* 重新执行 b)得到 Z_1^*,检查 $Z=?\ Z_1^*$,如果 $Z=Z_1^*$,那么 k_i 可能为中比特,并记录,否则 k_i 为卷入的密钥比特。

d)对 c)检测到可能的中比特 k_i 再次进行检测,选取密钥 K_2,重新执行 b)和 c),如果 k_i 仍是可能的密钥中比特,那么选取下一个密钥继续进行检测,直到取完 40 个密钥,k_i 仍是可能的密钥中比特,则确定 k_i 就是密钥中比特。

4　硬件实现

在对 SIMON32/64 算法的分析中,本文采用了 Altera EP2C8Q208CN FPGA 芯片来实现立方攻击和立方中比特的检测。该芯片提供了较多的可编程逻辑门电路、内存和时钟管理。

4.1　立方攻击的实现

根据对 SIMON32/64 的立方攻击的算法设计,除了 SIMON32/64 加密模块,本文还设计了另外三个模块:第一个模块为立方攻击模块,主要实现上述立方攻击算法的步骤 a) ~ e);第二个模块收集检测到的极大项和超级多项式;最后一个模块是一个传输单元。根据所采用 FPGA 芯片的内部资源的数量,以及 SIMON32/64 算法消耗资源的大小,决定并行运行 20 个 SIMON32/64 的立方攻击算法实例,采用 6 ~ 24 个立方变元。图 4 给出了实现架构。

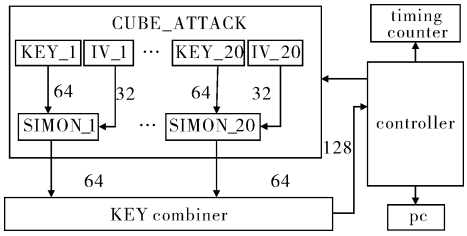

图 4　立方攻击的硬件结构框图

4.2　密钥中比特检测的实现

在对 SIMON32/64 进行密钥中比特的实现中,除了 SIMON32/64 加密模块,本文设计了另外三个模块:第一个模块为每个实例提供伪随机密钥 KEY 和所选的含有立方变元 IV* 的初始明文 IV(除立方变元,其余全取 0);第二个模块收集检测到的中比特 k 的输出;最后一个模块是一个传输单元。本文把 64 位密钥分为 ~~16 个部分,各有 4 bit,~~并行运行 ~~16~~ 个 SIMON32/64 算法实例,采用 6 ~ 24 个立方变元。图 5 给出了实现架构。

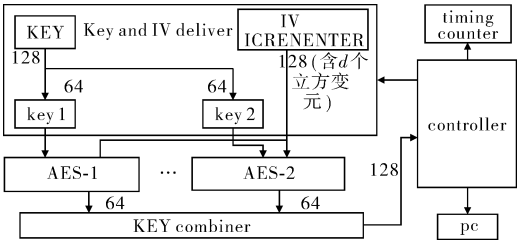

图 5　密钥中比特检测的硬件结构框图

此外,本文还用 C 在 PC(Intel Core I7-2600 3.4 GHz,4 GB RAM)中实现了上述攻击。测试结果表明,在一片 FPGA 上实现上述攻击的速度至少可以超过在传统的个人电脑实现的 40 倍,从而可以进行更高维度的深度检测。

5 攻击结果及分析

本文按照上述方法对 SIMON32/64 算法进行立方攻击和密钥中比特检测。检测结果如下：

1)立方攻击方面　随机立方变元取 6 ~ 24 个，分别对简化到 7 轮及以上的 SIMON32/64 进行立方攻击，检测输出的第 0 ~ 31 位。经过测试，发现在对 9 轮及以上的 SIMON32/64 的立方攻击中，无法通过立方攻击的预处理阶段，即无法找到极大项和有关密钥的线性多项式。而在对 7 轮 SIMON32/64 的立方攻击中，在随机立方变元取 6 ~ 24 时，能直接恢复 47 bit 密钥，其他 17 bit 密钥可以通过 2^{17} 次穷举得到，恢复全部密钥的时间复杂度为 $O(m\times 2^k+m^2+2^{n-m})=O(10\times 2^6+12\times 2^7+6\times 2^8+2\times 2^9+2\times 2^{10}+4\times 2^{11}+4\times 2^{12}+7\times 2^{14}+47\times 47+2^{64-47})\approx 2^{18.08}$，空间复杂度可忽略不计。对 8 轮 SIMON32/64 的立方攻击中，能直接恢复 39 bit 密钥，其他 25 bit 密钥可以通过 2^{25} 次穷举得到，恢复全部密钥的时间复杂度为 $O(m\times 2^k+m^2+2^{n-m})=O(14\times 2^7+12\times 2^8+2^9+4\times 2^{10}+4\times 2^{11}+3\times 2^{12}+2^{14}+39\times 39+2^{64-39})\approx 2^{25.00}$，空间复杂度可忽略不计。结果分别如表 1，2 所示。利用上述立方攻击分析的极大项集合和超级多项式结果，随机选取密钥，利用 FPGA 测试平台对 SIMON32/64 算法进行立方攻击，对于 7 轮 SIMON32/64 算法，花费时间不足 1 s 便可恢复所有密钥；对于 8 轮 SIMON32/64 算法，恢复所有密钥的时间仅需要不到 10 s。

表 1　7 轮 SIMON32/64 立方攻击结果

极大项集合	超级多项式	输出位
9,11,17,23,26,29	k50	1
8,9,15,23,24,27	k52	1
10,11,16,21,24,25	k60	2
0,3,10,18,19,24	k58	2
0,11,14,17,20,30	k59	1
10,13,16,21,22,27	k61	5
4,10,15,16,18,21	k63	5
0,3,6,24,25,30	k48	8
1,10,13,21,25,31	k49	9
13,24,26,27,28,31	k32 + k49 + k50	5
2,4,9,14,18,22,30	k57	5
4,6,12,18,27,28,31	k56	5
0,3,11,18,22,24,30	k53	2
10,13,14,21,24,25,29	1 + k54	2
6,10,14,18,24,26,27	k51	2
1,2,4,16,23,24,31	k46 + k54	3
0,16,17,18,23,25,30	k45 + k53 + k63	2
2,5,10,11,18,21,24	1 + k43 + k60 + k61	2
3,4,10,12,18,30,31	k22 + k46 + k48 + k58	5
3,5,13,14,22,24,31	1 + k21 + k39 + k53 + k57	20
6,10,11,17,18,27,30	1 + k24 + k42 + k56 + k60	30
1,12,13,17,18,19,24	k16 + k48	31
15,16,17,20,24,27,29,30	k54 + k55	4
0,1,8,9,10,12,20,31	1 + k30 + k38	20
1,8,10,12,16,19,23,29	1 + k29 + k46	19
4,6,7,15,19,22,26,27	1 + k18 + k35 + k52	17
1,4,12,16,18,22,23,26	k39 + k57	5
5,11,13,17,18,23,26,27	k62	5
6,9,13,15,19,20,21,23,28	1 + k27 + k45 + k59 + k63	10
2,3,13,14,16,22,25,27,29	k20 + k44 + k62	5
7,9,18,19,22,23,26,27,29,31	k25 + k33 + k60	4
0,2,6,7,8,9,16,20,22,30	k37 + k55 + k61	8
2,9,14,16,17,18,19,22,23,26,27	1 + k33 + k57	8
3,6,8,10,13,17,18,21,22,29,30	k12 + k29 + k30 + k46 + k48 + k63	12
1,5,9,13,15,17,19,20,24,28,31	1 + k34 + k58	9
0,2,6,8,16,19,22,23,27,28,29	k36 + k37 + k44 + k53	9
0,1,5,8,15,18,22,23,25,26,27,30	k33 + k40	8
0,4,8,9,11,12,14,18,19,28,29,31	k25 + k42 + k59 + 1	8
6,10,15,17,20,21,22,23,27,29,30,31	k17 + k26 + k27 + k41	9
5,6,12,13,17,18,20,21,24,25,28,30	1 + k17 + k34 + k35 + k41	9
2,5,6,9,17,18,19,20,22,24,27,28,30,31	k19 + k26 + k34 + k36 + k46 + k48 + k53	9
1,4,5,7,9,10,11,16,22,24,26,29,30,31	1 + k38 + k39 + k40 + k47 + k49 + k53	13
0,2,3,4,5,7,8,14,16,18,19,22,26,29	k31 + k39 + k57	14
1,2,10,11,12,17,19,20,21,22,23,28,29,30	k32 + k39 + k40 + k41 + k50 + k59	14
1,2,6,11,13,14,15,16,19,21,23,24,29,30	k44	9
1,2,3,5,10,14,15,16,21,24,25,28,29,30	1 + k38 + k55	17
1,5,10,11,13,16,17,18,21,22,25,26,28,29	k26 + k44 + k62	9

表 2　8 轮 SIMON32/64 立方攻击结果

极大项集合	超级多项式	输出位
1,6,10,12,17,25,31	1 + k59	24
5,7,9,13,23,27,30	k50	16
9,10,14,20,24,28,31	k51	17
2,12,15,18,27,29,30	k49	22
1,5,7,15,16,19,23	k54	22
3,4,7,17,18,30,31	k56	26
6,9,10,12,23,24,31	k48	26
0,5,13,14,19,27,30	k58	29
2,6,10,14,20,30,31	k55	29
7,10,13,17,23,27,31	k62	30
0,2,8,15,17,21,30	k44 + k52	16
3,9,16,17,19,20,26	k38 + k62	19
6,11,16,22,25,28,29	k46 + k63	27
0,6,11,16,18,20,27	1 + k24 + k42 + k56 + k60	22
4,6,9,10,12,16,23,27	k63	26
4,8,10,11,14,22,25,31	1 + k36 + k53	24
5,13,16,19,21,24,25,30	1 + k47 + k55	21
0,1,8,11,12,17,24,30	1 + k33 + k50	23
5,8,14,21,22,24,25,28	k45 + k62	24
5,8,10,20,22,26,28,29	k35 + k53 + k59	25
2,5,8,15,19,22,29,31	k39 + k56 + k57	22
6,7,8,18,21,24,25,29	1 + k42 + k59 + k60	23
5,13,16,19,21,24,25,30	1 + k47 + k55	21
0,1,8,11,12,17,24,30	1 + k33 + k50	23
5,8,14,21,22,24,25,28	k45 + k62	24
4,10,14,18,20,23,29,31	k22 + k40 + k48 + k54 + k58	20
6,8,11,17,20,21,23,25,31	k61	18
1,4,5,7,14,18,20,25,27,28	1 + k39 + k56	20
1,4,10,16,18,22,24,29,30,31	k40 + k58	20
1,3,6,7,9,22,25,28,29,31	1 + k43 + k48 + k51	17
2,3,7,10,15,16,18,22,26,30	k20 + k44 + k55	18
1,4,5,8,9,11,16,19,20,29,31	k60	17
2,3,8,16,20,22,23,24,25,29,31	1 + k37	18
0,2,3,11,14,15,17,18,21,24,29	k26 + k34 + k51 + k52	17
5,9,10,11,16,21,22,24,26,28,29	1 + k26 + k44 + k52 + k62	17
2,6,7,9,12,16,20,24,27,28,29,31	k18 + k42	16
0,6,7,8,11,17,19,21,22,23,24,31	1 + k36 + k60	17
1,2,6,10,13,14,16,19,21,25,28,31	1 + k1 + k25 + k55	16
2,4,5,8,13,16,18,19,20,21,25,26,27,29	k18 + k19	16

2)密钥中比特检测方面　随机立方变元取 6 ~ 24 个，分别对简化到 10 轮及以上的 SIMON32/64 进行测试，检测输出的第0 ~ 31 位。经过测试，发现在对 12 轮及以上的 SIMON32/64 检测中无法找到密钥中比特。而在对 10 轮 SIMON32/64 的检测中，在随机立方变元取 6 ~ 24 时，能找到 4 个密钥中比特，在对 11 轮 SIMON32/64 的检测中，能找到 2 个密钥中比特。结

果分别如表 3、4 所示。

从表 1、2 可以看出:对于简化到 7 轮、8 轮的 SIMON32/64 算法,可以直接恢复 47 bit 密钥和 39 bit 密钥。随着算法部件迭代次数的增加,涉及的变元个数也将大幅度增加。对于 9 轮及以上的 SIMON32/64 算法,无法找到极大项和有关密钥的线性多项式。因此,本文认为 9 轮以上的 SIMON32/64 算法能有效抵抗立方攻击。

表 3　10 轮 SIMON32/64 密钥中比特检测结果

维度	立方变元	立方中比特
10	2,4,5,6,7,18,21,23,26,30	k47
10	1,2,6,11,14,15,17,21,25,27	k15
11	0,1,4,8,10,15,20,21,23,24,29	k50
13	7,8,10,11,14,19,20,22,24,25,26,29,30	k44

表 4　11 轮 SIMON32/64 密钥中比特检测结果

维度	立方变元	立方中比特
15	3,5,6,10,13,14,15,16,17,18,19,24,25,26,28	k47
16	0,3,6,8,9,11,13,14,18,19,20,23,26,29,30,31	k15

从表 3、4 可以看出:在立方变元维度为 6 ~ 24 的情况下,简化 10 轮和 11 轮的 SIMON32/64 算法均能检测到密钥中比特。对于 12 轮及以上的 SIMON32/64 算法,无法检测到密钥中比特。可见全轮的 SIMON32/64 算法具有稳固的密钥信息扩散性和混淆性。

6　相关结果比较

表 5 给出的是 SIMON32/64 算法主要攻击结果的对比。从表 5 可以看出,本文攻击方法虽然在攻击轮数上低于其他的攻击方法,但在时间复杂度和数据复杂度上要远远低于其他的攻击方法,且本文的攻击是可以实现的攻击,并非理论意义上的攻击。从上述的攻击结果可以看出,该攻击方法对 7 轮、8 轮的 SIMON32/64 算法几乎可以达到实时破译。因此,立方攻击作为一种低轮高效的密钥恢复攻击方法,在对 7 轮、8 轮的 SIMON32/64 算法可以产生实质性的威胁。

表 5　SIMON32/64 算法主要攻击结果的对比

攻击方法	轮数	时间复杂度	空间复杂度	文献
立方攻击	7	$2^{18.08}$	–	本文
立方攻击	8	$2^{25.00}$	–	本文
线性分析	11	$2^{63.00}$	$2^{23.00}$	文献[3]
不可能差分分析	13	$2^{50.10}$	$2^{30.00}$	文献[3]
差分分析	18	$2^{46.00}$	$2^{31.2}$	文献[8]

在 2011 年的 CANCS 国际会议上,国内密码学者吴文玲教授提出了一种新的轻量级分组密码算法——LBlock(鲁班锁)[14]。该算法不仅拥有出色的软硬件实现性能,而且对各种密码分析表现出良好的抗攻击性能。LBlock 和 SIMON32/64 算法同为采用 Feistel 结构的轻量级分组密码,且都是各自代表在轻量级分组密码设计的一个最新的水平。鉴于以上特点,表 6 对比了针对 LBlock 和 SIMON32/64 算法的立方攻击结果。由表 6 可以看出,低轮的 SIMON32/64 和 LBlock 算法均无法有效地抵抗立方攻击。

表 6　SIMON32/64 算法与 LBlock 攻击结果的对比

算法	分组大小 /bit	密钥长度 /bit	算法轮数	攻击轮数	时间复杂度	文献
LBlock	64	80	32	9	$2^{47.00}$	文献[15]
SIMON32/64	32	64	32	8	$2^{25.00}$	本文

7　结束语

本文主要根据 SIMON32/64 的密钥编排特点和轮函数结构,结合立方攻击和立方测试的基本思想,利用 FPGA 测试平台设计了一个 SIMON32/64 的立方攻击和密钥中比特检测算法。并利用此算法检测了 SIMON32/64 算法在抵抗立方攻击方面的抗攻击能力和算法内部结构对密钥的混淆扩散能力,并与其他攻击方法及国内同级别的算法 LBlock 进行了对比。下一步将利用检测到的密钥中比特来构建区分器以减少具体攻击中的时间复杂度。

参考文献:

[1] Beaulieu R, Shors D, Smith J, *et al*. The SIMON and SPECK families of lightweight block ciphers, report 2013/404[R]. [S. l.]:IACR , 2013.

[2] A1Khzaimi H, Lauridsen M M. Cryptanalysis of the SIMON family of block ciphers[R]. [S. l.]: IACR, 2013.

[3] Abed F, List E, Lucks S, *et al*. Differential and linear cryptanalysis of reduced-round SIMON report 2013/526[R]. Cryptology ePrint Archive, 2013.

[4] Biryukov A, Roy A, Velichkov V. Differential analysis of block ciphers SIMON and SPECK[C]//Proc of the 21st International Workshop on Fast Software Encryption. Berlin: Springer, 2014: 546-570.

[5] Wang Ning, Wang Xiaoyun, Jia Keting, *et al*. Improved differential attacks on reduced SIMON versions, report 2014/48[R]. [S. l.]: Shandong University, 2014.

[7] Alizadeh J, Alkhzaimi H A, Aref M R, *et al*. Improved linear cryptanalysis of round reduced SIMON, reprot 2014/681 2014[R/OL]. http://eprint. iacr. org/2014/681. pdf.

[8] Tupsarnudre H, Bisht S, Mukhopadhyay D. Differential fault analysis on the families of SIMON and SPECK ciphers[C]//Proc of Workshop on Fault Diagnosis and Tolerance in Cryptography-FDTC. [S. l.]: IEEE Press, 2014: 40-48.

[6] Wang Qingju, Liu Zhiqiang, Vancl K, *et al*. Cryptanalysis of reduced-round SIMON32 and SIMON48[C]//Proc of the 15th International Conference on Progress in Cryptology. [S. l.]:Spring International Publishing, 2014: 143-160.

[9] Dinur I, Shamir A. Cube attacks on tweakable black box polynomials [C]//Proc of EUROCRYPT 2009. Berlin: Springer, 2009: 278-299.

[10] Aumasson J, Dinur I, Meier W, *et al*. Cube tester and key recovery attacks on reduced-round MD6 and Trivium[C]//Proc of FSE 2009. Berlin: Springer, 2009: 1-22.

[11] Mroczkowski P, Szmidt J. The algebraic cryptanalysis of the block cipher Katan32 using modified cube attack[R]. Warsaw, Poland: Military University of Technology, 2011: 345-354.

[12] Mroczkowski P, Szmidt J. The cube attack on stream cipher trivium and quadraticity tests[C]//Proc of the 10th Central European Conference on Cryptology. 2010:309-318.

[13] Dinur I, Shamir A. Breaking grain-128 with dynamic cube attacks [C]//Proc of the 18th International Workshop on Fast Software Encryption. Berlin: Springer, 2011: 167-187.

[14] Wu Wenliang, Zhang Lei. LBlock: a lightweight block cipher[C]// Proc of the 9th International Conference on Applied Cryptography and Network Security. Berlin: Springer, 2011: 327-344.

[15] Islam S, Afzal M, Rashdi A. On the security of LBlock against the cube attack and side channel cube attack. Security Engineering and Intelligence Informatics [C]//LHCS, vol 8128. Berlin: Springer, 2013: 105-121.