密码算法旁路立方攻击改进与应用

王永娟 王 涛* 袁庆军 高 杨 王相宾
(战略支援部队信息工程大学 郑州 450001)
(河南省网络密码技术重点实验室 郑州 450001)

摘 要：立方攻击的预处理阶段复杂度随输出比特代数次数的增长呈指数级增长，寻找有效立方集合的难度也随之增加。该文对立方攻击中预处理阶段的算法做了改进，在立方集合搜索时，由随机搜索变为带目标的搜索，设计了一个新的目标搜索优化算法，优化了预处理阶段的计算复杂度，进而使离线阶段时间复杂度显著降低。将改进的立方攻击结合旁路方法应用在MIBS分组密码算法上，从旁路攻击的角度分析MIBS的算法特点，在第3轮选择了泄露位置，建立关于初始密钥和输出比特的超定的线性方程组，可以直接恢复33 bit密钥，利用二次检测恢复6 bit密钥。所需选择明文量$2^{21.64}$，时间复杂度2^{25}。该结果较现有结果有较大改进，恢复的密钥数增多，在线阶段的时间复杂度降低。

关键词：立方攻击；旁路攻击；预处理；二次检测；MIBS算法
中图分类号：TP309.7 **文献标识码**：A **文章编号**：1009-5896(2020)05-1087-07
DOI: 10.11999/JEIT181075

Side Channel Cube Attack Improvement and Application to Cryptographic Algorithm

WANG Yongjuan WANG Tao YUAN Qingjun
GAO Yang WANG Xiangbin
(*PLA Strategic Support Force Information Engineering University*, *Zhengzhou* 450001, *China*)
(*Henan Key Laboratory of Network Cryptography Technology*, *Zhengzhou* 450001, *China*)

Abstract: The complexity of the pre-processing phase of the cubic attack grows exponentially with the number of output bit algebras, and the difficulty of finding an effective cube set increases. In this paper, the algorithm of preprocessing stage in cubic attack is improved. In the cube set search, from random search to target search, a new target search optimization algorithm is designed to optimize the computational complexity of the preprocessing stage. In turn, the offline phase time complexity is significantly reduced. The improved cubic attack combined with the side-channel method is applied to the MIBS block cipher algorithm. The algorithm characteristics of MIBS are analyzed from the perspective of side-channel attack. The leak location is selected in the third round, and the overdetermined linear equations from initial key and output bit are established, which can directly recover 33bit key. Then the 6bit key can be recovered by quadric-detecting. The amount of plaintext required is $2^{21.64}$, time complexity is 2^{25}. This result is greatly improved compared with the existing results, the number of keys recovered is increased, and the time complexity of the online phase is reduced.

Key words: Cube attack; Side channel attack; Preprocessing; Quadric-detecting; MIBS algorithm

1 引言

2009年，Dinur和Shamir[1]在欧密会上提出了立方攻击。立方攻击是一种代数攻击方法，其主要思想是将密码算法视为黑盒多项式，通过寻找指标集与其对应的低次超多项式的方法恢复密钥。其原理与高阶差分攻击[2]和AIDA攻击[3]相似。立方攻击适用于分组密码、序列密码、哈希函数等多种密码算法，出现了很多理论成果[4–7]。2009年，Dinur等人[4]对767轮的Trivium算法进行了立方攻击，以2^{45}的时间复杂度恢复了35位密钥。2010年，Mroczkowski等人[5]将二次检测应用于对709轮Trivium算法，恢复了80 bit密钥，其中二次检测中恢复了39 bit。2017

收稿日期：2018-11-23；改回日期：2019-11-27；网络出版：2020-03-25
*通信作者： 王涛 1072637697@qq.com
基金项目：国家自然科学基金(61872381, 61602512)
Foundation Items: The National Natural Science Foundation of China(61872381, 61602512)

年，Todo等人[6]利用可分性将立方攻击视为非黑盒攻击，恢复了832轮Trivium, 183轮Grain128a, 704轮ACORN以及872轮Kreyvium的部分密钥。2017年，Szmidt[7]攻击了约简至4轮的CTC算法，恢复了全部120 bit密钥，预处理阶段的复杂度为2^{11}次4轮CTC加密，并利用中间相遇方法将攻击扩展到5轮。

立方攻击的成功率与密码算法的代数次数有很大关系，代数次数的大小关系着指标集的大小，而攻击的时间复杂度随指标集的增加而增加。随着算法轮数的增加，代数次数越来越高，为了攻击更高的轮数，传统的立方攻击已不适用。后来，在之前立方攻击相关工作基础上，出现了一些立方攻击的变种，比如旁路立方攻击[8]，动态立方攻击[9]，故障立方攻击[10]，条件立方攻击[11]，相关立方攻击[12]。

旁路立方攻击是将旁路攻击与立方攻击结合的一种新的攻击方法，由Dinur等人[8]于2009年提出，主要思想是通过旁路信息恢复密码算法运行的中间状态的某一比特，然后通过寻找关于这一比特的低次超多项式进行攻击。旁路立方攻击同样出现了很多理论成果[13–16]。2009年，Yang等人[13]将旁路立方攻击应用到PRESENT算法中，选取第3轮第1, 2, 3位为泄露位，以2^{15}的选择明文量可恢复48 bit密钥。2010年，Abdul-latip等人[14]对NOEKEON密码进行攻击，以$2^{10.89}$的选择明文量，2^{68}的时间复杂度恢复全部密钥。此外，旁路立方攻击还应用于Simeck[15], MIBS[16]等算法上。

本文主要对旁路立方攻击的预处理阶段进行改进，代数次数的增加会使立方攻击的预处理阶段复杂度带来指数级的增长，对预处理阶段的优化可以极大提高旁路立方攻击的效率。在立方集合搜索时，设计了一个新的目标搜索优化算法，优化了预处理阶段的计算复杂度，进而使离线阶段时间复杂度显著降低。将改进的立方攻击结合旁路方法应用在MIBS分组密码算法上，在第3轮选择了泄露位置，建立关于初始密钥和输出比特的超定的线性方程组，可以直接恢复27 bit密钥，利用二次检测恢复6 bit密钥。所需选择明文量为$2^{21.61}$，时间复杂度为2^{31}。

2 基础知识

2.1 立方攻击原理

密码算法的最后输出可以描述为F_2上关于密钥K和公开变量(明文比特或者IV比特)的低次多项式，通过合理选择公开变量值可能获得一些关于密钥的线性关系，从而得到关于密钥的线性方程组，进而通过解方程组恢复一定数目的密钥。

考虑由下面的布尔函数描述的密码体制：$z=f(p_1,p_2,\cdots,p_m,k_1,k_2,\cdots,k_n)$。其中$p_1,p_2,\cdots,p_m$是$m$个公开变量，$k_1,k_2,\cdots,k_n$是$n$个秘密变量，函数值代表密文输出。在序列密码中一般IV为公开变量，K为秘密变量，分组密码中明文为公开变量，密钥为秘密变量。通常f的次数很高，甚至f的具体形式可能是未知的(黑盒)。为得到线性/低次方程，有如定理1：

定理1[5]任意n元布尔函数$f(x_1,x_2,\cdots,x_n)$，对任意指标集$I=\{i_1,i_2,\cdots,i_k\}\subseteq\{1,2,\cdots,n\}$，则函数$f$总可以表示为如下形式：$f(x_1,x_2,\cdots,x_n)=t_I\cdot p_I\oplus q(x_1,x_2,\cdots,x_n)$，其中$\oplus$表示模2加，$t_I=x_{i_1}x_{i_2}\cdots x_{i_k}$，$q(x_1,x_2,\cdots,x_n)$中单项式不是$t_I$的倍式。本文称$t_I$为cube，即立方体，称$p_I$为立方体对应的超级多项式。

性质1[5]$\sum_{\{i_1,i_2,\cdots,i_k\}}f(x_1,x_2,\cdots,x_n)=p_I$。证明见文献[5]。

为说明这一定理与性质，给出例1。

例1：设五元布尔函数$f(x_1,x_2,\cdots,x_5)=x_1x_2x_3\oplus x_1x_2x_4\oplus x_2x_4x_5\oplus x_1x_2\oplus x_3x_5\oplus x_5$，取立方体$I=\{1,2\}$，$t_I=x_1x_2$，则$f=x_1x_2(x_3\oplus x_4\oplus 1)\oplus x_2x_4x_5\oplus x_3x_5\oplus x_5$，超多项式$p_I=x_3\oplus x_4\oplus 1$。

立方攻击一般分为预处理阶段与在线阶段。立方攻击的主要难点在第1个阶段，如何寻找满足条件的指标集I与超级多项式。第1个阶段的具体算法介绍见3.1节。

预处理阶段：通过改变公共变量与秘密变量的值，寻找满足条件的指标集I，即I对应的超级多项式可以通过BLR线性测试[17]或二次测试。

在线阶段：固定集合I之外的变量，穷举I并计算加和，得到超级多项式的值。多个指标集可以得到多个超级多项式组成的方程组，通过解方程组获得密钥。

2.2 旁路立方攻击原理

立方攻击中，输出比特对应的多项式f随轮数的增长而增加，当次数达到一定界限时，立方体维数超过计算机能够计算的最大值，立方攻击失败。当通过旁路分析得到算法运行的中间状态比特时，本文可以通过选择泄露的比特控制其对应多项式的次数，在计算机可以计算的范围内恢复尽可能多的密钥。旁路立方攻击等效于约简轮的密码分析。而且旁路立方攻击可以构成对密码算法的实现构成威胁，具有研究价值。

2.3 MIBS算法

MIBS算法在CANS2009上由Izadi等人[18]提出，是一种Feistel结构的分组密码算法。MIBS分组长64 bit，迭代32轮，初始密钥长64或80 bit，用

MIBS-64和MIBS-80表示密钥长度。本文的攻击对象是64 bit密钥的MIBS，攻击方法也可应用在80 bit版本。MIBS中的所有运算都作用在4 bit，即半字节上。轮函数结构如图1。

轮函数的主体F具有由轮密钥加，4×4 S盒的S层和混淆层M和字节置换P构成的SPN结构，包含下面3个步骤(由于M混淆和P字节置换均为线性变换，统一组合为线性变换P)：

轮密钥加：将上一轮输出的左半部分与轮密钥异或。

混淆层S：将轮密钥加的结果每4位一组过S盒，S盒见表1。

扩散层P：将S代换的输入用$\{y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8\}$表示，输出用$\{y_1',y_2',y_3',y_4',y_5',y_6',y_7',y_8'\}$表示，线性变换为

$$\left.\begin{aligned}
y_1' &= y_1\oplus y_2\oplus y_4\oplus y_5\oplus y_7\oplus y_8\\
y_2' &= y_2\oplus y_3\oplus y_4\oplus y_5\oplus y_6\oplus y_7\\
y_3' &= y_1\oplus y_2\oplus y_3\oplus y_5\oplus y_6\oplus y_8\\
y_4' &= y_2\oplus y_3\oplus y_4\oplus y_7\oplus y_8\\
y_5' &= y_1\oplus y_3\oplus y_4\oplus y_5\oplus y_8\\
y_6' &= y_1\oplus y_2\oplus y_4\oplus y_5\oplus y_6\\
y_7' &= y_1\oplus y_2\oplus y_3\oplus y_6\oplus y_7\\
y_8' &= y_1\oplus y_3\oplus y_4\oplus y_6\oplus y_7\oplus y_8
\end{aligned}\right\}\tag{1}$$

MIBS-64初始密钥为64 bit，记为$K(k_{63},k_{62},\cdots,k_0)$。每处理一轮后将前32 bit留存作为轮密钥，共处理32轮。密钥扩展方案见文献[17]。

3　立方攻击改进

立方攻击的预处理阶段复杂度随输出比特代数次数的增长呈指数级增长，寻找有效立方集合的难度也随之增加。本节对立方攻击中预处理阶段的算法做了改进，在立方集合搜索时，设计了一个新的目标搜索优化算法，优化了预处理阶段的计算复杂度，进而使离线阶段时间复杂度显著降低。

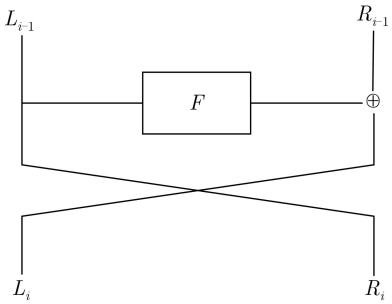

图 1 MIBS算法结构

表 1　MIBS算法S盒

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	4	f	3	8	d	a	c	0	b	5	7	e	2	6	1	9

3.1　立方攻击预处理阶段

对于一个黑盒多项式，本文不知道其具体的代数标准型，可以通过图2中的方法寻找攻击需要的指标集I和超多项式p_I：

步骤 1　随机选择公共变量的一个子集I，其它公共变量设为固定值；固定值一般全设置为0，文献[9]中指出也可以一部分设为0，一部分设为1。

步骤 2　设I对应的超多项式为p_I，随机选择两个密钥$x,y\in\{0,1\}^n$，进行BLR线性测试：

若$p_I[0]\oplus p_I[x]\oplus p_I[y]=p_I[x\oplus y]$，则$p_I$可能是线性的，否则$p_I$必不是线性的。

步骤 3　若测试不通过，转步骤1；若测试通过，转步骤2继续测试，若连续通过N次测试，认为p_I以接近1的概率是线性的，保存指标集I，计算p_I的代数标准型。

步骤 4　转到步骤1，直到获得足够数量的指标集I。

若p_I为线性多项式，接下来的方法可以确定p_I的代数标准型：

定理 2[1]　设布尔函数的代数标准型为$f(x_1,x_2,\cdots,x_n)=\oplus_{i=0}^{2^n-1}a_ix^i$，其中$i=\{i_1,i_2,\cdots,i_n\}$，$x^i=x_1^{i_1}\cdots x_n^{i_n}$。则布尔函数的每一项的系数$a_i=\sum_{x\le i}f(x_1,x_2,\cdots,x_n)$，$x\le i$表示$x$的汉明重量小于等于$i$。证明见文献[1]。

将定理2应用到确定p_I的代数标准型上需要两步：

步骤 1　首先确定常数项。此时$i=0$，系数$a_0=f(0)$，即密钥取全0，计算$p_I(0)$;

步骤 2　确定p_I中一次项的系数。取密钥X_j第j位为1，其余位为0，则该一次项的系数为$p_I(0)\oplus p_I(X_j)$。

3.2　目标搜索优化算法

常规的立方攻击预处理阶段指标集I是随机选取的，这样选择的效率比较低，会做很多重复性的工作。考虑线性测试，在进行N次测试后，其结果可以分为3种情况：

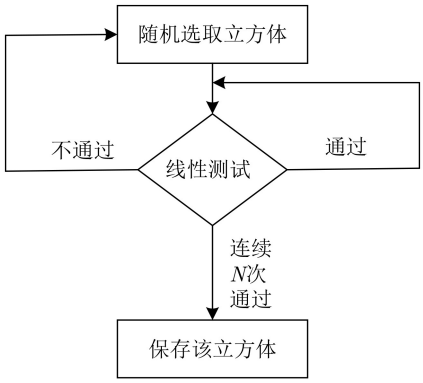

图 2 标准预处理阶段流程图

情况1 p_I通过线性检测，且对于每次测试随机选择的密钥$x,y\in\{0,1\}^n$，始终有$p_I[0]=p_I[x]=p_I[y]=p_I[x\oplus y]$。此时认为超多项式$p_I$的次数为0，即$p_I$只有常数项。由于这样的$p_I$不包含秘密变量，对密钥恢复没有意义。

情况2 p_I通过线性检测，且在某次测试$p_I[0],p_I[x],p_I[y],p_I[x\oplus y]$中存在两项不相等。此时认为超多项式$p_I$的次数为1，这种情况是密钥恢复需要的情况。

情况3 p_I不通过线性检测。此时超多项式p_I的次数大于等于2，需要重新随机选择I进行测试。

本文对于立方攻击的改进在于情况2出现后的处理方法。当选择的指标集I对应的超多项式p_I出现情况2时，不是重新选择指标集，而是在现有的指标集中添加一个元素，然后对新的指标集进行线性测试。此时p_I仍然有可能出现3种情况，在出现情况2时继续添加元素进行测试，直到出现情况0或情况1。这样立方体搜索算法就从一个随机算法变成了一个有目标的算法，显著减少了预处理阶段所需要的时间复杂度。重新设计后的算法分为如下4步，见图3。

步骤 1 随机选择公共变量的一个子集I，其它公共变量设为固定值；

步骤 2 设I对应的超多项式为p_I，随机选择两个密钥$x,y\in\{0,1\}^n$，进行N次BLR线性测试；

步骤 3 若测试结果为情况0，转到步骤1；若测试结果为情况1，保存I，计算p_I的代数标准型，转到步骤4；若测试结果为情况2，在I中添加一个元素，转到步骤2；

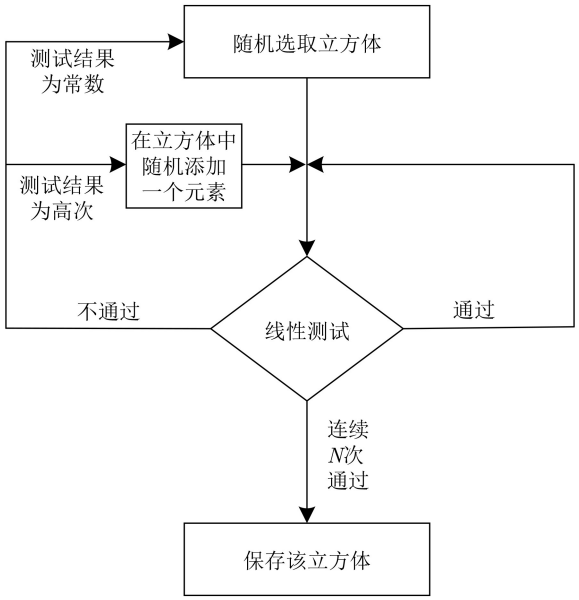

图 3 改进预处理阶段流程

步骤 4 转到步骤1，直到获得足够数量的指标集I。

4 对MIBS算法的旁路立方攻击

本节利用目标搜索优化算法，针对MIBS算法进行旁路立方攻击，在第3轮采集泄露信息，进行了线性检测与二次检测，具体攻击过程如下。

4.1 泄露位置选取

旁路立方攻击中进行最佳泄露位置选取时，泄露比特需尽可能的覆盖更多的密钥变量，且对应的多项式次数应尽可能地小，以降低攻击复杂度。

令$[x_0x_1x_2x_3]$和$[y_0y_1y_2y_3]$分别表示MIBS的S盒输入和输出比特，根据真值表计算出其对应的代数标准型为

$$\left.\begin{aligned}
y_0 &= x_0x_1x_2\oplus x_1x_2x_3\oplus x_0x_2\oplus x_1x_3\\
&\quad\oplus x_0\oplus x_1\oplus x_3\\
y_1 &= x_0x_2x_3\oplus x_0x_1x_3\oplus x_1x_2\oplus x_0x_3\oplus x_1x_3\\
&\quad\oplus x_0\oplus x_2\oplus 1\\
y_2 &= x_0x_2x_3\oplus x_1x_2x_3\oplus x_0x_1x_3\oplus x_1x_2\\
&\quad\oplus x_0x_2\oplus x_0\oplus x_2\oplus x_3\\
y_3 &= x_0x_1x_2\oplus x_0x_2x_3\oplus x_1x_2x_3\oplus x_0x_3\\
&\quad\oplus x_0x_2\oplus x_0\oplus x_1\oplus x_2\oplus x_3
\end{aligned}\right\}\tag{2}$$

根据代数标准型可以看到，每个S盒输出比特的代数次数均为3。根据这一性质可以估计MIBS算法每轮S盒输出比特的代数次数上界，第r轮S盒输出比特的代数次数上界为3^r。前4轮的次数上界为3, 9, 27和81。由于MIBS算法明文为64 bit，所以第4轮次数上界应为64，超过了计算机能够计算的范围。由于第3轮输出比特覆盖的密钥变量最多，能够提取出的立方体最多，能恢复的密钥量最多，所以选择第3轮S盒输出为泄露位置。

运用符号计算方法可以得出第3轮S盒输出比特的准确代数次数，见表2。

用如下算法可以测试第3轮S盒输出覆盖的密钥变量个数，见表3。其中$D[64]$存储输出比特涉及密钥的个数；f函数为MIBS 3轮加密，输入是明文和密钥，输出为MIBS算法第3轮S盒输出；$K'=\neg K(j)$表示第j位密钥取反，其它值不变。算法基于这样

表 2 第3轮S盒输出比特代数次数

位置	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
次数	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27
位置	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
次数	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27
位置	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
次数	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
位置	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
次数	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

一个原理：当一个密钥参与运算时，改变它的值有可能改变最终输出；当一个密钥不参与运算时，改变它的值不会改变最终输出。

第3轮S盒输出比特覆盖密钥数见表4。通过该表可以看出，涉及密钥比特最多的位置是8-11, 24-27，均有59位密钥参与运算。因此MIBS算法的泄露位置可以取第3轮S盒输出的第8-11, 24-27位。

4.2 线性测试

攻击中，选取第8位输出作为泄露位置，按照3.2节中的方法进行立方体搜索，经高斯消元后得到表5。选择明文量为$2\times 2^7+2\times 2^8+4\times 2^{10}+3\times 2^{11}+7\times 2^{12}+5\times 2^{13}+4\times 2^{14}+2\times 2^{15}+2^{17}+2^{18}+2\times 2^{20}\approx 2^{21.37}$，可恢复33 bit密钥信息。

表 3　算法2的过程

算法2 第3轮S盒输出覆盖密钥变量个数检测

输出：每个比特覆盖的密钥变量个数

(1) set $D[64]$ to 0
(2) for $i=0$ to 63 do//表示64个输出比特
(3)　for $j=0$ to 63 do//检测64个密钥是否参与输出比特运算
(4)　　for $k=1$ to N do//测试N次
(5)　　GetRandom(K)
(6)　　$K'=\neg K(j)$
(7)　　$t=f(K)\oplus f(K')$
(8)　　if $\exists t=1$
(9)　　$D[i]++$
(10) return　D

4.3 二次测试

立方攻击得到的超多项式不一定必须是线性的，二次多项式也可以恢复密钥信息。特别在一次超多项式已经恢复一定密钥信息的情况下，二次多项式有可能转化为一次多项式。将3.2节中的线性测试换为二次，可以恢复更多的密钥。选择明文量为$2^7+2^{10}+3\times 2^{14}+2^{19}\approx 2^{19.13}$，可恢复6 bit密钥信息，结果见表6。

二次测试：设I对应的超多项式为p_I，随机选择3个密钥$x,y,z\in\{0,1\}^n$，若$p_I[0]\oplus p_I[x]\oplus p_I[y]\oplus p_I[z]\oplus p_I[x\oplus y]\oplus p_I[y\oplus z]\oplus p_I[z\oplus x]=p_I[x\oplus y\oplus z]$，则$p_I$的次数可能小于等于2，否则$p_I$次数必大于2。

表 4　第3轮S盒输出比特覆盖密钥数

位置	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
密钥数	58	58	58	58	58	58	58	58	59	59	59	59	58	58	58	58
位置	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
密钥数	58	58	58	58	58	58	58	58	59	59	59	59	58	58	58	58
位置	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
密钥数	43	43	43	43	43	43	43	43	44	44	44	44	43	43	43	43
位置	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
密钥数	43	43	43	43	43	43	43	43	44	44	44	44	43	43	43	43

表 5　第3轮输出第8位泄露立方体

维数	立方体	超多项式
7	29 30 38 40 43 44 45	$k_2+k_3+k_4+k_6+k_7+k_8+k_{13}+k_{16}+k_{38}$
7	23 27 28 35 47 60 61	$k_1+k_4+k_{34}+k_{37}+k_{50}+k_{57}+k_{60}$
8	0 1 2 3 4 5 40 41	k_{55}
8	0 1 2 3 8 9 52 53	k_{59}
10	0 1 2 3 4 5 6 8 32 56	$k_{59}+k_{60}$
10	0 1 2 3 4 5 6 16 43 48	k_3+k_4
10	0 1 2 3 8 9 10 20 35 55	k_7+k_8
10	4 7 16 19 29 30 31 32 33 34	k_2+k_{54}
11	0 1 2 3 4 5 6 12 13 48 63	k_0
11	0 1 2 3 4 5 6 16 17 40 49	k_3
11	0 1 2 3 8 9 10 20 21 32 53	k_7
12	0 1 2 3 4 5 6 8 10 24 36 39	$k_{11}+k_{12}$
12	0 1 2 3 4 5 6 16 17 19 31 43	k_{13}
12	1 2 3 4 5 6 7 8 9 10 18 32	k_2
12	1 2 3 4 5 6 7 8 9 10 26 32	k_{10}
12	0 1 2 3 4 5 6 8 9 11 15 43	k_{61}
12	0 1 2 4 5 6 7 8 10 12 43 52	k_0+k_{63}
12	0 1 2 3 4 5 6 8 9 11 12 14 41	k_{62}

续表 5

维数	立方体	超多项式
13	0 1 2 3 4 5 6 16 17 19 28 29 40	k_{15}
13	0 1 2 3 4 5 6 16 17 19 28 29 41	$k_{15}+k_{16}$
13	0 1 2 3 4 5 6 16 17 19 28 30 41	k_{14}
13	5 6 17 19 25 26 27 28 29 30 31 32 33	$1+k_{53}$
13	9 11 21 23 29 30 31 32 33 35 36 38 39	k_5+k_{57}
14	1 2 3 4 5 6 8 9 10 11 20 21 22 32	$1+k_0+k_{39}+k_{40}+k_{43}+k_{44}+k_{63}$
14	1 2 3 4 5 6 8 9 10 11 20 21 22 33	$1+k_0+k_{40}+k_{44}$
14	1 2 3 4 5 6 8 9 10 11 20 21 23 32	$k_0+k_1+k_{38}+k_{39}+k_{40}+k_{41}+k_{42}+k_{43}+k_{44}+k_{45}+k_{62}+k_{63}$
14	1 2 3 4 5 6 8 9 10 11 20 21 23 33	$1+k_{38}+k_{42}+k_{62}$
15	0 1 2 3 4 5 6 8 9 10 12 14 25 27 63	$1+k_{11}$
15	1 2 3 4 5 6 8 9 10 11 12 13 15 27 6227 62	k_9
17	4 5 13 14 15 16 17 18 19 20 21 22 24 25 26 28 30	k_{56}
18	1 2 3 4 5 6 7 8 9 11 12 13 14 16 17 18 20 23	k_6+k_7
20	1 2 3 4 5 6 8 9 10 12 13 14 16 17 19 20 22 23 58 61	$k_{38}+k_{41}+k_{58}+k_{61}$
20	1 2 3 4 5 7 8 9 10 12 13 14 16 17 18 28 30 31 38 61	$1+k_1+k_{42}+k_{45}+k_{54}+k_{57}+k_{58}+k_{61}+k_{62}$

表 6 二次测试立方体

维数	立方体	超多项式
7	0 1 2 3 4 40 41	$1+k_{55}+k_{56}+k_{54}k_{55}+k_{55}k_{56}$
10	0 1 2 3 4 5 6 8 32 40	$1+k_{59}+k_{58}k_{60}+k_{59}k_{60}$
14	0 1 2 3 4 5 6 8 9 10 12 13 14 17	$k_3+k_1k_4+k_3k_4$
14	0 1 2 3 4 5 6 8 9 10 12 13 14 21	$k_7+k_5k_8+k_7k_8$
14	0 1 2 3 4 5 6 8 9 10 12 13 14 29	$k_{15}+k_{13}k_{16}+k_{15}k_{16}$
19	1 2 3 4 5 6 8 9 10 12 13 14 16 17 19 20 22 58 61	$k_8k_{38}+k_8k_{41}+k_8k_{58}+k_8k_{61}$

表 7 结果对比

方法	攻击轮数	选择明文	密钥搜索空间
文献[16]	1	$2^{6.39}$	2^{40}
文献[19]	4	未提及	2^{72}
本文	3	$2^{21.64}$	2^{25}

4.4 结果比较

文献[16]利用传统的立方攻击方法，针对MIBS加密第1轮输出的第5比特泄露，利用选择明文分析将MIBS-64密钥搜索空间降低至2^{40}。文献[19]分析了轻量级分组密码抵抗立方攻击的能力，指出SIMECK和MIBS相比于KATAN, SIMON, SPECK等算法对立方攻击有较好的抗性。该论文没有给出具体的立方体，但指出恢复MIBS-80密钥需要2^{72}时间复杂度，意味着只找到8个低次超多项式，恢复了8 bit密钥信息。本文的方法与与之前文献相比相比，选择明文量至$2^{21.37}+2^{19.13}\approx 2^{21.64}$，但穷举量降低至$2^{25}$，总体复杂度降低，如表7所示。

5 结束语

本文对立方攻击中预处理阶段的算法做了改进，由随机搜索变为带目标的搜索，离线阶段时间复杂度显著降低。从旁路攻击的角度分析了MIBS的算法特点，选择了泄露位置。将改进的立方攻击结合旁路方法应用在MIBS算法上，恢复的密钥数增多，在线阶段的时间复杂度降低。此方法的改进具有普适性，未来可以用在其它分组密码、序列密码以及哈希函数的立方攻击中。

参 考 文 献

[1] DINUR I and SHAMIR A. Cube attacks on tweakable black box polynomials[C]. The 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, 2009: 278–299. doi: 10.1007/978-3-642-01001-9_16.

[2] LAI Xuejia. Higher Order Derivatives and Differential Cryptanalysis[M]. BLAHUT R E, COSTELLO JR D J, MAURER U, *et al*. Communications and Cryptography. Boston: Springer, 1994: 227–233. doi: 10.1007/978-1-4615-2694-0_23.

[3] VIELHABER M. Breaking ONE. FIVIUM by AIDA an algebraic IV differential attack[R]. 2007.

[4] AUMASSON J P, DINUR I, MEIER W, *et al*. Cube testers and key recovery attacks on reduced-round MD6 and trivium[C]. The 16th International Workshop on Fast Software Encryption, Leuven, Belgium, 2009: 1–22. doi: 10.1007/978-3-642-03317-9_1.

[5] MROCZKOWSKI P and SZMIDT J. The cube attack on stream cipher trivium and quadraticity tests[J]. *Fundamenta Informaticae*, 2012, 114(3/4): 309–318. doi: 10.3233/FI-2012-631.

[6] TODO Y, ISOBE T, HAO Yonglin, *et al*. Cube attacks on non-blackbox polynomials based on division property[J]. *IEEE Transactions on Computers*, 2018, 67(12): 1720–1736. doi: 10.1109/TC.2018.2835480.

[7] SZMIDT J. The cube attack on Courtois toy cipher[C]. The 1st International Conference on Number-Theoretic Methods in Cryptology, Warsaw, Poland, 2017: 241–253. doi: 10.1007/978-3-319-76620-1_14.

[8] DINUR I and SHAMIR A. Side channel cube attacks on block ciphers[J]. *IACR Cryptology Eprint Archive*, 2009: 127.

[9] DINUR I and SHAMIR A. Breaking Grain-128 with dynamic cube attacks[C]. The 18th International Workshop on Fast Software Encryption, Lyngby, Denmark, 2011: 167–187. doi: 10.1007/978-3-642-21702-9_10.

[10] 马云飞, 王韬, 陈浩, 等. SIMON系列轻量级分组密码故障立方攻击[J]. 浙江大学学报: 工学版, 2017, 51(9): 1770–1779. doi: 10.3785/j.issn.1008-973X.2017.09.011.
MA Yunfei, WANG Tao, CHEN Hao, *et al*. Fault-cube attack on SIMON family of lightweight block ciphers[J]. *Journal of Zhejiang University*: *Engineering Science*, 2017, 51(9): 1770–1779. doi: 10.3785/j.issn.1008-973X.2017.09.011.

[11] HUANG Senyang, WANG Xiaoyun, XU Guangwu, *et al*. Conditional cube attack on reduced-round Keccak sponge function[C]. The 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, 2017: 259–288. doi: 10.1007/978-3-319-56614-6_9.

[12] LIU Meicheng, YANG Jingchun, WANG Wenhao, *et al*. Correlation cube attacks: From weak-key distinguisher to key recovery[C]. The 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, 2018: 715–744. doi: 10.1007/978-3-319-78375-8_23.

[13] YANG Lin, WANG Meiqin, and QIAO Siyuan. Side channel cube attack on PRESENT[C]. The 8th International Conference on Cryptology and Network Security, Kanazawa, Japan, 2009: 379–391. doi: 10.1007/978-3-642-10433-6_25.

[14] ABDUL-LATIP S F, REYHANITABAR M R, SUSILO W, *et al*. On the security of NOEKEON against side channel cube attacks[C]. The 6th International Conference on Information Security Practice and Experience, Seoul, South Korea, 2010: 45–55. doi: 10.1007/978-3-642-12827-1_4.

[15] BUJA A G, ABDUL-LATIP S F, and AHMAD R. A security analysis of IoT encryption: Side-channel cube attack on Simeck32/64[J]. *International Journal of Computer Networks & Communications*, 2018, 10(4): 79–90. doi: 10.5121/ijcnc.2018.10406.

[16] 刘会英, 王韬, 郭世泽, 等. MIBS密码旁路立方体攻击[J]. 计算机仿真, 2013, 30(5): 302–305. doi: 10.3969/j.issn.1006-9348.2013.05.069.
LIU Huiying, WANG Tao, GUO Shize, *et al*. Side channel cube attacks on MIBS[J]. *Computer Simulation*, 2013, 30(5): 302–305. doi: 10.3969/j.issn.1006-9348.2013.05.069.

[17] FISCHER S, KHAZAEI S, and MEIER W. Chosen IV statistical analysis for key recovery attacks on stream ciphers[C]. The 1st International Conference on Cryptology in Africa, Casablanca, Morocco, 2008: 236–245. doi: 10.1007/978-3-540-68164-9_16.

[18] IZADI M, SADEGHIYAN B, SADEGHIAN S S, *et al*. MIBS: A new lightweight block cipher[C]. The 8th International Conference on Cryptology and Network Security, Kanazawa, Japan, 2009: 334–348. doi: 10.1007/978-3-642-10433-6_22.

[19] ZAHERI M and SADEGHIAN B. Comparing resistance against cube like attacks[C]. The 24th Iranian Conference on Electrical Engineering, At Shiraz, Iran, 2016.

王永娟：女，1972年生，副教授，主要研究方向为网络空间安全、密码算法分析.

王　涛：男，1995年生，硕士生，研究方向为侧信道攻击.

袁庆军：男，1993年生，助教，研究方向为侧信道攻击.

高　杨：男，1994年生，硕士生，研究方向为故障攻击.

王相宾：男，1996年生，硕士生，研究方向为侧信道攻击.