SIMON64算法的积分分析

徐 洪[①②] 方玉颖*[①] 戚文峰[①②]

[①](信息工程大学 郑州 450001)

[②](数学工程与先进计算国家重点实验室 郑州 450001)

摘 要：SIMON系列算法自提出以来便受到了广泛关注。积分分析方面，Wang，Fu和Chu等人给出了SIMON32和SIMON48算法的积分分析，该文在已有的分析结果上，进一步考虑了更长分组的SIMON64算法的积分分析。基于Xiang等人找到的18轮积分区分器，该文先利用中间相遇技术和部分和技术给出了25轮SIMON64/128算法的积分分析，接着利用等价密钥技术进一步降低了攻击过程中需要猜测的密钥量，并给出了26轮SIMON64/128算法的积分分析。通过进一步的分析，该文发现高版本的SIMON算法具有更好抵抗积分分析的能力。

关键词：等价密钥；SIMON64；中间相遇；部分和；积分分析

中图分类号：TP309.7; TN918.1 **文献标识码**：A **文章编号**：1009-5896(2020)03-0720-09

DOI: 10.11999/JEIT190230

Integral Attacks on SIMON64

XU Hong[①②] FANG Yuying[①] QI Wenfeng[①②]

[①](*Information Engineering University*, *Zhengzhou* 450001, *China*)

[②](*State Key Laboratory of Mathematical Engineering and Advanced Computing*, *Zhengzhou* 450001, *China*)

Abstract: The SIMON block cipher receives extensive attention since its proposed. With respect to integral attacks, some integral attacks on SIMON32 and SIMON48 are presented by Wang, Fu and Chu *et al*. In this paper, on the basis of the existing analysis results, the integral attacks on SIMON64 are further studied. Based on known 18-round integral distinguisher presented by Xiang *et al.*, the integral attacks on 25-round SIMON64/128 are presented using meet-in-the-middle and partial-sum techniques. Then the amount of subkeys that need to be guessed during the attack is further reduced by equivalent-subkey technique, and the improved integral attacks on 26-round SIMON64/128 are also presented. Through further analysis, it is found that the higher version of SIMON algorithm has better resistance to integral analysis.

Key words: Equivalent-subkey; SIMON 64; Meet-in-the-middle; Partial-sum; Integral attacks

1 引言

积分分析是分组密码的重要分析方法，由Knudsen等人[1]在Square攻击[2]等方法基础上提出，对AES[3], MISTY[4]等算法都有很好的攻击效果。积分分析的核心思想是通过选择特定输入明文集，分析经多轮加密后某些位置比特是否具有特定的积分性质，比如平衡性，即对应的状态集求和为0的情形，再根据这些积分性质排除错误密钥。

SIMON系列算法[5]由美国国家安全局设计提出，是一类重要的轻量分组密码。算法轮函数结构简单，仅由循环移位、异或和按位与构成，其中按位与为主要的非线性部件。目前对SIMON算法安全性能的分析主要包括差分分析[6–11]、线性分析[8,12–14]、不可能差分分析[12,15–17]、零相关线性分析[15,18]和积分分析[19–21]等。在积分分析方面，Wang等人[15]给出了SIMON32算法的15轮积分区分器和21轮积分分析，Xiang等人[19]利用混合整数线性规划方法给出了SIMON系列算法的积分区分器，Fu等人[20]利用等价密钥技术将SIMON32算法积分分析的轮数提高了一轮，并且给出了SIMON48算法的积分分析，Chu等人[21]利用动态密钥猜测技术进一步提高了攻击轮数，本文在此基础上考虑了更长分组的SIMON64算法的积分分析。基于Xiang等人[19]找到的18轮积分区分器，本文首先利用中间相遇技术和部分和技术

收稿日期：2019-04-09；改回日期：2019-12-04；网络出版：2019-12-10

*通信作者： 方玉颖 fangyywy@163.com

基金项目：十三五国家密码发展基金(MMJJ20180204, MMJJ20170103)

Foundation Items: The National Cryptography Development Fund (MMJJ20180204, MMJJ20170103)

给出了对25轮SIMON64/128算法的积分分析，接着利用等价密钥技术进一步降低攻击过程中需要猜测的密钥量，将攻击轮数又提高了一轮，给出了26轮SIMON64/128算法的积分分析。

本文后续部分安排如下：第2节简要介绍SIMON系列算法及相关符号。第3节基于已有的SIMON64算法的18轮积分区分器，用中间相遇技术和部分和技术给出对25轮SIMON64/128算法的积分分析。第4节介绍等价密钥技术，并基于此给出改进的26轮SIMON64/128算法的积分分析。第5节是结束语。

2　SIMON算法简介

2.1　符号说明

X_r表示算法第r轮左边的输入；Y_r表示算法第r轮右边的输入；K_r表示第r轮的子密钥；K_r^*表示第r轮的等价子密钥；$X_{r,\{i\sim j\}}$表示X_r的第$i\sim j$ bit; $Y_{r,\{i\sim j\}}$表示Y_r的第$i\sim j$ bit; $X<<<s$表示X循环左移s bit；$X>>>s$表示X循环右移s bit；$\oplus$表示按位异或；$\wedge$表示按位与。

2.2　SIMON算法简介

SIMON系列算法[5]采用Feistel结构，其轮函数如图1所示。设初始明文为(X_0,Y_0)，第r轮的输入为(X_r,Y_r)，记算法的轮函数$F(X)=(X<<<1)\wedge(X<<<8)\oplus(X<<<2)$，则第$r+1$轮的输出为

$$(X_{r+1},Y_{r+1})=(F(X_r)\oplus Y_r\oplus K_r,X_r) \tag{1}$$

SIMON系列算法支持多种分组长度和密钥长度的组合，不妨简记分组长度为$2n$ bit，密钥长度为mn bit的SIMON算法为SIMON$2n/mn$算法，其中$n\in\{16,24,32,48,64\}$, $m\in\{2,3,4\}$。SIMON算法的密钥扩展方案会根据字数m值的不同而有所不同，前m轮子密钥由主密钥直接生成，剩余轮子密钥的生成满足式(2)的关系

$$K_{i+m}=\begin{cases}c\oplus z_{j,\{i\}}\oplus K_i\oplus(K_{i+1}>>>3)\oplus(K_{i+1}>>>4), & m=2\\ c\oplus z_{j,\{i\}}\oplus K_i\oplus(K_{i+2}>>>3)\oplus(K_{i+2}>>>4), & m=3\\ c\oplus z_{j,\{i\}}\oplus K_i\oplus K_{i+1}\oplus(K_{i+1}>>>1)\oplus(K_{i+3}>>>3)\oplus(K_{i+3}>>>4), & m=4\end{cases} \tag{2}$$

其中$i=0,1\cdots,T-m$, $c=2^n-4$, $z_j\,(j\in\{0,1,2,3,4\})$是和版本有关的常数序列。从SIMON的密钥扩展算法可以看出，每个子密钥均可用主密钥线性表示，并且已知任意连续m个子密钥就可以恢复出主密钥。

3　SIMON64算法的积分分析

本节利用Xiang等人[19]找到的积分区分器给出对SIMON 64/128和SIMON 64/96算法的积分分析。他们找到的SIMON 64算法的17轮积分区分器形如$(CA\cdots A,A\cdots A)\rightarrow_{17}(?\cdots?,BBBBBBBBBBB$ $?????B?????BBBBBBBBBB)$，其中A表示活跃比特，B表示平衡比特，?表示未知比特，C表示常量比特。类似于Wang等人[15]的处理方法，该区分器可以自然向前扩展一轮，得到如图2所示的SIMON64算法的18轮积分区分器。

图2所示的18轮积分区分器输出的右边部分有22个平衡比特，下面以最右边的平衡比特为例，在后面添加7轮，给出对SIMON64/128算法的25轮积分分析。图3给出了密钥恢复过程需要用到的子密钥和中间状态比特。密钥恢复过程需要猜测$K_{19,\{24,30,31\}}$，$K_{20,\{0,16,22,23,28\sim30\}}$，$K_{21,\{8,14,15,20\sim22,24,26\sim31\}}$，$K_{22,\{0,6,7,12\sim14,16,18\sim30\}}$，$K_{23,\{4\sim6,8,10\sim31\}}$, $K_{24,\{0,2\sim30\}}$共99 bit子密钥，攻击中需要的选择明文数为2^{63}，直接计算的复杂度超过2^{128}。下面采用中间相遇技术和部分和技术来降低密钥恢复过程的计算复杂度。

注意到$(Y_{18}\oplus K_{18})_{\{0\}}=(X_{18,\{31\}}\wedge X_{18,\{24\}})\oplus(X_{18}\oplus X_{19})_{\{30\}}$，要判断$\oplus(Y_{18}\oplus K_{18})_{\{0\}}=0$是否成立，只要判断$\oplus(X_{18,\{31\}}\wedge X_{18,\{24\}})=\oplus(X_{18}\oplus X_{19})_{\{30\}}$是否成立，因此可以分别计算和式$\oplus(X_{18,\{31\}}\wedge X_{18,\{24\}})$和$\oplus(X_{18}\oplus X_{19})_{\{30\}}$，并比较二者是否相等。图4和图5分别给出了计算

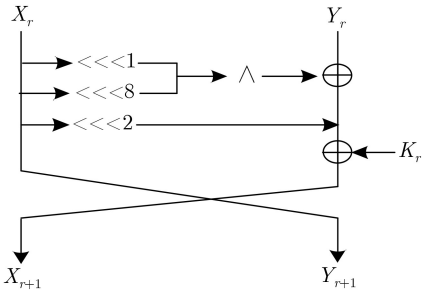

图 1 SIMON算法的轮函数

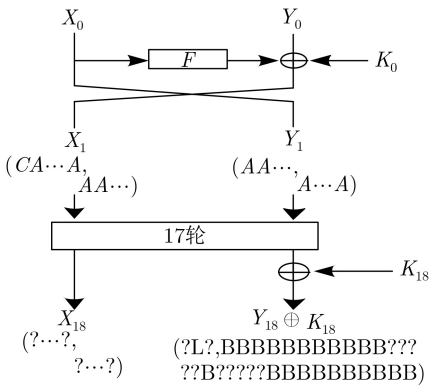

图 2 SIMON 64算法的18轮积分区分器

$\oplus\left(X_{18,\{31\}} \wedge X_{18,\{24\}}\right)$和$\oplus(X_{18} \oplus X_{19})_{\{30\}}$时需要猜测的子密钥和中间状态比特，其中计算前者需要猜测73 bit子密钥，计算后者需要猜测81 bit子密钥，重复猜测55 bit子密钥。

对25轮SIMON64/128算法进行积分分析的主要步骤如下：

(1) 利用密钥扩展算法，将子密钥比特$K_{19,\{24,30,31\}}$和$K_{20,\{0,16,22,23,28\sim30\}}$用$(K_{21}, K_{22}, K_{23}, K_{24})$线性表出；

(2) 选择2^{63}个明文，它们在一轮加密后仅左边第1个比特为常数，其余均为活动比特；

(3) 猜测73 bit相关的子密钥，计算$\oplus\left(X_{18,\{31\}} \wedge X_{18,\{24\}}\right)$，并将结果存储在表$T_1$中；

(4) 猜测81 bit相关的子密钥，计算$\oplus(X_{18} \oplus X_{19})_{\{30\}}$，并将结果存储在表$T_2$中；

(5) 对每组猜测的子密钥，若表T_1和T_2匹配，则保留相应的子密钥为正确的候选子密钥，匹配后剩下$2^{73+81-1-55} = 2^{98}$个候选子密钥集；

(6) 对剩下的子密钥比特，猜测$K_{21,\{2\sim7,9\sim13,18,19\}}$, $K_{22,\{4,5,8\sim11,15,17,31\}}$, $K_{23,\{1,2,3,7,9\}}$和$K_{24,\{1,31\}}$共29 bit

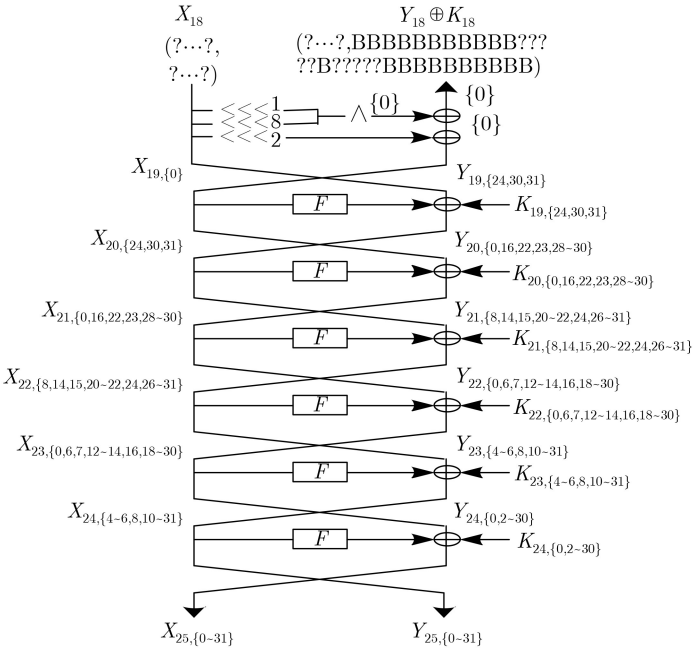

图 3 25轮积分分析的密钥恢复过程

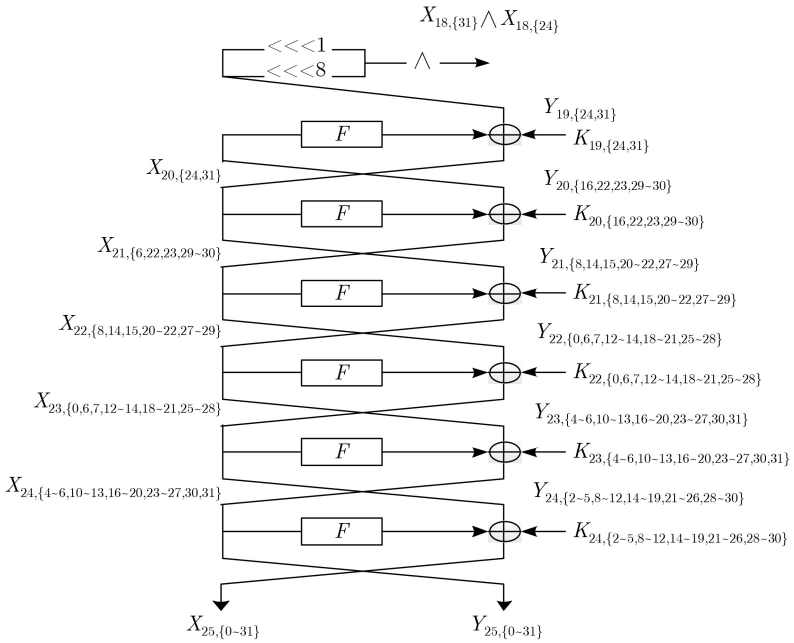

图 4 计算$X_{18,\{31\}} \wedge X_{18,\{24\}}$的过程

密钥，根据步骤(1)的线性关系用高斯消元法可求得$K_{21,\{0,1,16,17,23,25\}}$，$K_{22,\{1\sim3\}}$和$K_{23,\{0\}}$共10 bit的密钥信息再利用密钥扩展算法由最后4轮子密钥$(K_{21},K_{22},K_{23},K_{24})$恢复出主密钥，并用两组明密文对验证密钥的正确性。

表1和表2分别给出了第3步和第4步中计算$\oplus\left(X_{18,\{31\}}\wedge X_{18,\{24\}}\right)$和$\oplus(X_{18}\oplus X_{19})_{\{30\}}$的详细过程和复杂度分析，其中计算$\oplus\left(X_{18,\{31\}}\wedge X_{18,\{24\}}\right)$的具体步骤如下：

(1) 对2^{24}个猜测的密钥比特$K_{24,\{2\sim5,8\sim12,14\sim19,21\sim26,28\sim30\}}$和$2^{63}$个密文值，计算$Y_{24,\{2\sim5,8\sim12,14\sim19,21\sim26,28\sim30\}}$的值，统计43 bit状态$X_{24,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}$和$Y_{24,\{2\sim5,8\sim12,14\sim19,21\sim26,28\sim30\}}$，保留出现奇数次的情形，时间复杂度约为$2^{81.94}$次25轮SIMON64/128算法加密；

(2) 对2^{19}个猜测的密钥比特$K_{23,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}$和$2^{43}$个保留状态，计算$Y_{23,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}$的值，统计33 bit状态$X_{23,\{0,6,7,12\sim14,18\sim21,25\sim28\}}$和$Y_{23,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}$，保留出现奇数次的情形，时间复杂度约为$2^{80.61}$次25轮SIMON64/128算法加密；

(3) 对2^{14}个猜测的密钥比特$K_{22,\{0,6,7,12\sim14,18\sim21,25\sim28\}}$和$2^{33}$个保留状态，计算$Y_{22,\{0,6,7,12\sim14,18\sim21,25\sim28\}}$的值，统计23 bit状态

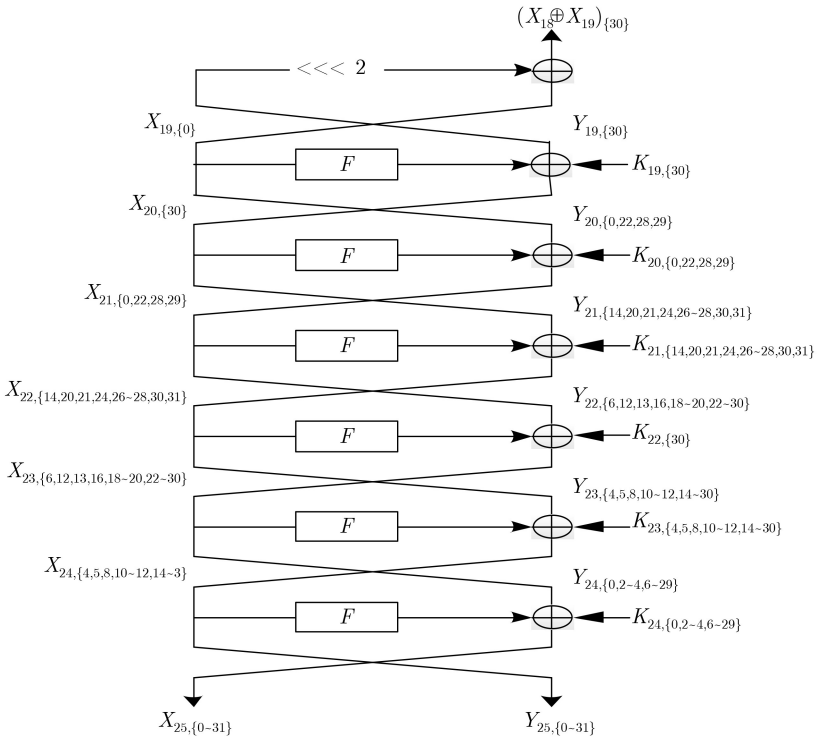

图 5 计算$(X_{18}\oplus X_{19})_{\{30\}}$的过程

表 1　计算$\oplus\left(X_{18,\{31\}}\wedge X_{18,\{24\}}\right)$的复杂度

步骤	猜测密钥(比特数)	统计状态(比特数)	时间复杂度
(1)	$K_{24,\{2\sim5,8\sim12,14\sim19,21\sim26,28\sim30\}}(24)$	$X_{24,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}(19)$, $Y_{24,\{2\sim5,8\sim12,14\sim19,21\sim26,28\sim30\}}(24)$	$2^{24}\cdot 2^{63}\cdot\frac{24}{32\cdot25}\approx 2^{81.94}$
(2)	$K_{23,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}(19)$	$X_{23,\{0,6,7,12\sim14,18\sim21,25\sim28\}}(14)$, $Y_{23,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}(19)$	$2^{43}\cdot 2^{43}\cdot\frac{19}{32\cdot25}\approx 2^{80.61}$
(3)	$K_{22,\{0,6,7,12\sim14,18\sim21,25\sim28\}}(14)$	$X_{22,\{8,14,15,20\sim22,27\sim29\}}(9)$, $Y_{22,\{0,6,7,12\sim14,18\sim21,25\sim28\}}(14)$	$2^{57}\cdot 2^{33}\cdot\frac{14}{32\cdot25}\approx 2^{84.16}$
(4)	$K_{21,\{8,14,15,20\sim22,27\sim29\}}(9)$	$X_{21,\{16,22,23,29,30\}}(5)$, $Y_{21,\{8,14,15,20\sim22,27\sim29\}}(9)$	$2^{66}\cdot 2^{23}\cdot\frac{9}{32\cdot25}\approx 2^{82.53}$
(5)	$K_{20,\{16,22,23,29,30\}}(5)$	$X_{20,\{24,31\}}(2)$, $Y_{20,\{16,22,23,29,30\}}(5)$	$2^{71}\cdot 2^{14}\cdot\frac{5}{32\cdot25}\approx 2^{77.68}$
(6)	$K_{19,\{24,31\}}(2)$	$X_{18,\{24,31\}}(2)$, $X_{18,\{24\}}\wedge X_{18,\{31\}}(1)$	$2^{73}\cdot 2^{7}\cdot\frac{3}{32\cdot25}\approx 2^{71.95}$

$X_{22,\{8,14,15,20\sim22,27\sim29\}}$和$Y_{22,\{0,6,7,12\sim14,18\sim21,25\sim28\}}$，保留出现奇数次的情形，时间复杂度约为$2^{84.16}$次25轮SIMON 64/128算法加密；

(4) 对2^9个猜测的密钥比特$K_{21,\{8,14,15,20\sim22,27\sim29\}}$和$2^{23}$个保留状态，计算$Y_{21,\{8,14,15,20\sim22,27\sim29\}}$的值，统计14 bit状态$X_{21,\{16,22,23,29,30\}}$和$Y_{21,\{8,14,15,20\sim22,27\sim29\}}$，保留出现奇数次的情形，时间复杂度约为$2^{82.53}$次25轮SIMON64/128算法加密；

(5) 对2^5个猜测的密钥比特$K_{20,\{16,22,23,29,30\}}$和$2^{14}$个保留状态，计算$Y_{20,\{16,22,23,28,29\}}$的值，统计7 bit状态$X_{20,\{24,31\}}$和$Y_{20,\{16,22,23,29,30\}}$，保留出现奇数次的情形，时间复杂度约为$2^{77.68}$次25轮SIMON64/128算法加密；

(6) 对2^2个猜测的密钥比特$K_{19,\{24,31\}}$和2^7个保留状态，计算并统计3 bit状态$X_{18,\{24,31\}}$和$X_{18,\{24\}}\wedge X_{18,\{31\}}$，保留出现奇数次的情形，时间复杂度约为$2^{71.95}$次25轮SIMON64/128算法加密。

上述计算$\oplus\left(X_{18,\{31\}}\wedge X_{18,\{24\}}\right)$的过程中总共需要猜测73 bit子密钥，计算复杂度约为$2^{84.87}$次25轮SIMON64/128算法加密，存储复杂度约为$2^{73}\cdot 74\approx 2^{79.21}$ bit。类似地，计算$\oplus(X_{18}\oplus X_{19})_{\{30\}}$的过程中总共需要猜测81 bit子密钥，计算复杂度约为$2^{100.51}$次25轮SIMON64/128算法加密，存储复杂度约为$2^{81}\cdot 82\approx 2^{87.36}$ bit。

复杂度分析：根据上面的分析，第(3)和第(4)步构造表T_1和T_2的时间复杂度约为$2^{100.51}+2^{84.87}\approx 2^{100.51}$次25轮SIMON64/128算法加密，剩下$2^{73+81-1-55}=2^{98}$个候选子密钥。第(6)步中高斯消元的时间开销远低于25轮算法加密的复杂度可以忽略不计，而猜测剩下的29 bit子密钥，并用2组明密文对分别验证的复杂度约为$2^{98}\cdot 2^{29}\left(1+2^{-64}\right)\approx 2^{127}$次25轮SIMON64/128算法加密，故总的计算复杂度约为2^{127}次25轮SIMON64/128算法加密，存储复杂度约为$2^{79.21}+2^{87.36}\approx 2^{87.37}$ bit，约为2^{85} Byte。

4 SIMON64算法改进的积分分析

本节利用Fu等人[20]提出的等价密钥技术进一步降低密钥恢复过程的密钥猜测量和计算复杂度，给出SIMON64算法的改进的积分分析，基于此方法攻击轮数可以再增加一轮。本文先给出SIMON64/128算法的26轮积分分析。

如图6所示，利用等价密钥技术，可以将SIMON64/128算法最后一轮(第25轮)的子密钥K_{25}移至上一轮中，其中图6(a)为最后2轮变换的原

表 2 计算$\oplus(X_{18}\oplus X_{19})_{\{30\}}$的复杂度

步骤	猜测密钥(bit数)	统计状态(bit数)	时间复杂度
(1)	$K_{24,\{0,2\sim4,6\sim29\}}(28)$	$X_{24,\{4,5,8,10\sim12,14\sim30\}}(23)$, $Y_{24,\{0,2\sim4,6\sim29\}}(28)$	$2^{28}\cdot 2^{63}\cdot\frac{28}{32\cdot 25}\approx 2^{86.17}$
(2)	$K_{23,\{4,5,8,10\sim12,14\sim30\}}(23)$	$X_{23,\{6,12,13,16,18\sim20,22\sim30\}}(16)$, $Y_{23,\{4,5,8,10\sim12,14\sim30\}}(23)$	$2^{51}\cdot 2^{51}\cdot\frac{23}{32\cdot 25}\approx 2^{96.88}$
(3)	$K_{22,\{6,12,13,16,18\sim20,22\sim30\}}(16)$	$X_{22,\{14,20,21,24,26\sim28,30,31\}}(9)$, $Y_{22,\{6,12,13,16,18\sim20,22\sim30\}}(16)$	$2^{67}\cdot 2^{39}\cdot\frac{16}{32\cdot 25}\approx 2^{100.36}$
(4)	$K_{21,\{14,20,21,24,26\sim28,30,31\}}(9)$	$X_{21,\{0,22,28,29\}}(4)$, $Y_{21,\{14,20,21,24,26\sim28,30,31\}}(9)$	$2^{76}\cdot 2^{25}\cdot\frac{9}{32\cdot 25}\approx 2^{94.53}$
(5)	$K_{20,\{0,22,28,29\}}(4)$	$X_{20,\{30\}}(1)$, $Y_{20,\{0,22,28,29\}}(4)$	$2^{80}\cdot 2^{13}\cdot\frac{4}{32\cdot 25}\approx 2^{85.36}$
(6)	$K_{19,\{30\}}(1)$	$X_{18,\{31\}}(1)$, $\oplus(X_{18}\oplus X_{19})_{\{30\}}$ (1)	$2^{81}\cdot 2^{5}\cdot\frac{2}{32\cdot 25}\approx 2^{77.36}$

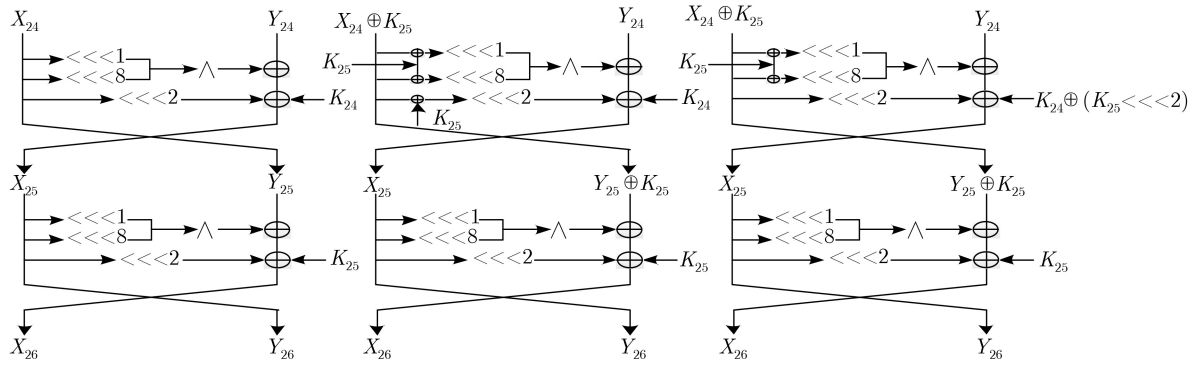

图 6 等价密钥技术示意图

形式，图6(c)为相应的等价形式，前面几轮也可以类似处理。一般地，记第j轮的等价子密钥为K_{j+1}^*，则有$K_{25}^*=K_{25}$，$K_j^*=K_j\oplus\left(K_{j+1}^*<<<2\right)$，其中$19\le j\le 24$，这些等价密钥$(K_{19}^*,K_{20}^*,\cdots,K_{25}^*)$可以由原子密钥$(K_{19},K_{20},\cdots,K_{25})$线性表出。

从图6可以看出，采用等价密钥技术后算法的中间状态也会相应变成原状态和某些等价子密钥的异或，然而由于积分分析研究的是中间状态值求和的性质，而添加密钥常数不影响状态和的值，为叙述方便，下面仍用符号X_i和Y_i表示算法的中间状态。图7给出了等价密钥情形下对SIMON64/128算法进行26轮积分分析密钥恢复过程中需要猜测的子密钥比特和中间状态比特。攻击过程中需要猜测$K^*_{19,\{24,31\}}$，$K^*_{20,\{16,22,23,29,30\}}$，$K^*_{21,\{8,14,15,20\sim22,27\sim29\}}$，$K^*_{22,\{0,6,7,12\sim14,18\sim21,25\sim28\}}$，$K^*_{23,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}$，$K^*_{24,\{2\sim5,8\sim12,14\sim19,21\sim26,28\sim30\}}$和$K^*_{25,\{0\sim4,6\sim29\}}$共102 bit子密钥，同样采用中间相遇方法降低计算复杂度。

注意到$Y_{18,\{0\}}=\left(X_{18,\{31\}}\wedge X_{18,\{24\}}\right)\oplus Y_{19,\{30\}}\oplus X_{19,\{0\}}$，判断$\oplus Y_{18,\{0\}}$是否等于0可以转化为判断$\oplus\left(\left(X_{18,\{31\}}\wedge X_{18,\{24\}}\right)\oplus X_{19,\{0\}}\right)=\oplus Y_{19,\{30\}}$是否成立。不妨记$M_1=\left(X_{18,\{31\}}\wedge X_{18,\{24\}}\right)\oplus X_{19,\{0\}}$，$M_2=Y_{19,\{30\}}$，图8和图9分别给出了计算$\oplus M_1$和$\oplus M_2$的过程中需要猜测的等价子密钥比特和中间状态比特。

表3和表4分别给出了计算$\oplus M_1$和$\oplus M_2$的详细过程和复杂度分析。计算$\oplus M_1$时需要猜测89 bit子密钥，计算复杂度约为$2^{92.31}$次26轮加密，存储复杂度约为$2^{89}\cdot 90\approx 2^{95.50}$ bit，而计算$\oplus M_2$时需要猜测69 bit子密钥，计算复杂度约为$2^{71.92}$次26轮加密，存储复杂度约为$2^{69}\cdot 70\approx 2^{75.13}$ bit。

复杂度分析： 同第3节的分析。26轮

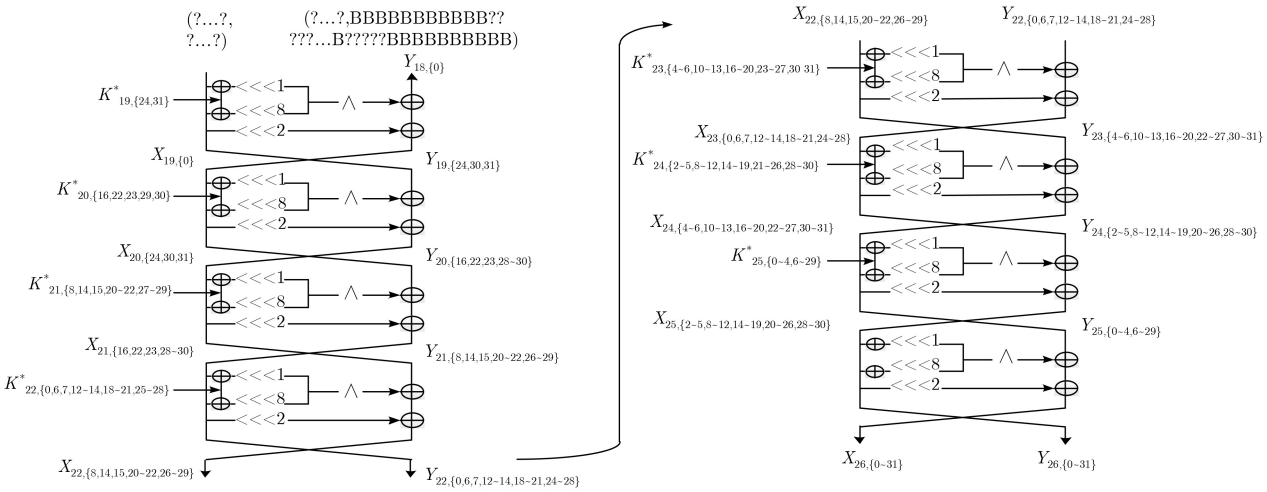

图 7 SIMON64/128算法的26轮积分分析

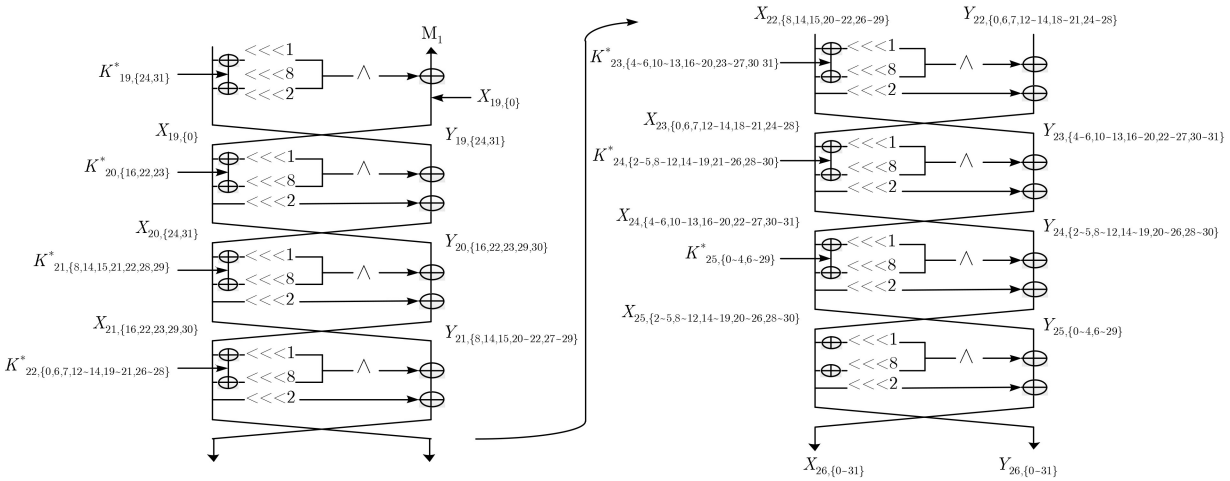

图 8 计算$\oplus M_1$的过程

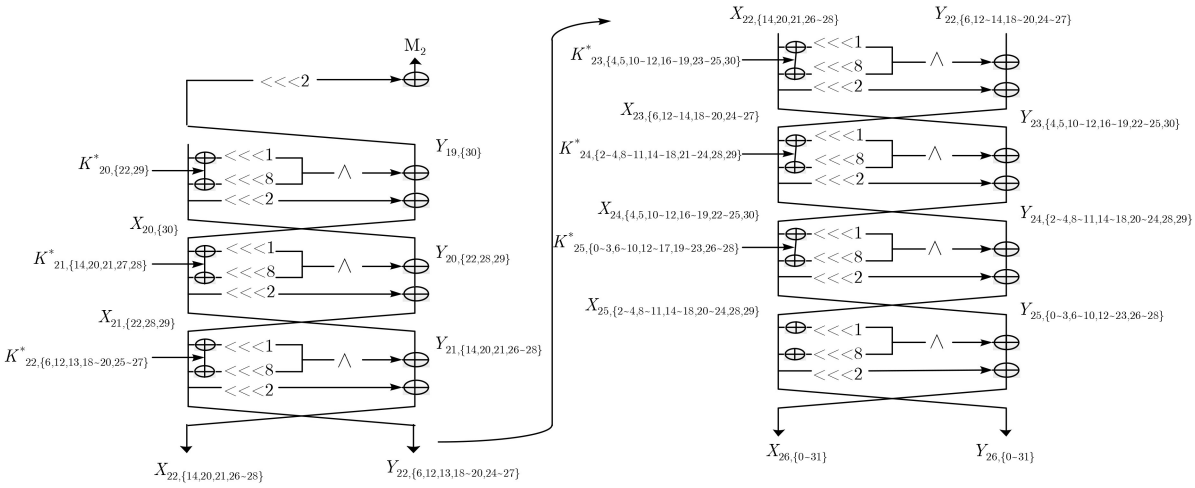

图 9 计算$\oplus M_2$的过程

表 3 计算$\oplus M_1$值的复杂度

步骤	猜测密钥(比特数)	统计状态(比特数)	时间复杂度
(1)	–	$X_{25,\{2\sim5,8\sim12,14\sim19,21\sim26,28\sim30\}}(24)$, $Y_{25,\{0\sim4,6\sim29\}}(29)$	$2^{63}\cdot\frac{29}{32\cdot26}\approx2^{58.16}$
(2)	$K^*_{25,\{0\sim4,6\sim11,13\sim18,20\sim29\}}(27)$	$X_{24,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}(19)$, $Y_{24,\{2\sim5,8\sim12,14\sim19,21\sim26,28\sim30\}}(24)$	$2^{27}\cdot2^{53}\cdot\frac{24}{32\cdot26}\approx2^{74.89}$
(3)	$K^*_{24,\{2,5,9,12,16,19,23,26,30\}}(9)$	$X_{23,\{2,3,9,10,16,17,23,24,28\}}(9)$, $Y_{23,\{6,10,13,17,20,24,27,31\}}(8)$, $X_{24,\{4,11,18,25\}}(4)$	$2^{36}\cdot2^{43}\cdot\frac{8}{32\cdot26}\approx2^{72.30}$
(4)	$K^*_{24,\{3,10,17,24,28\}}(5)$	$X_{23,\{0,3,4,6\sim8,10\sim15,17\sim22,25\sim29\}}(23)$, $Y_{23,\{4,11,18,25\}}(4)$, $X_{24,\{2,12,16,19,23,26,30\}}(7)$	$2^{41}\cdot2^{21}\cdot\frac{4}{32\cdot26}\approx2^{54.30}$
(5)	$K^*_{24,\{4,8,11,15,18,22,25,29\}}(8)$	$X_{23,\{0,6,7,12\sim14,18\sim21,25\sim28\}}(14)$, $Y_{23,\{4\sim6,10\sim13,16\sim20,23\sim27,30,31\}}(19)$	$2^{49}\cdot2^{34}\cdot\frac{19}{32\cdot26}\approx2^{77.55}$
(6)	$K^*_{23,\{4,11,18,25\}}(4)$	$X_{22,\{4,5,11,12,18,19,25,26,30\}}(9)$, $Y_{22,\{12,19,26\}}(3)$, $X_{23,\{6,13,20,27\}}(4)$	$2^{53}\cdot2^{33}\cdot\frac{3}{32\cdot26}\approx2^{77.88}$
(7)	$K^*_{23,\{5,12,19,26\}}(4)$	$X_{22,\{5,6,8,10,12\sim17,19\sim24,26\sim31\}}(22)$, $Y_{22,\{6,13,20,27\}}(4)$, $X_{23,\{0,7,14,18,21,25,28\}}(7)$	$2^{57}\cdot2^{16}\cdot\frac{4}{32\cdot26}\approx2^{65.30}$
(8)	$K^*_{23,\{6,10,13,17,20,24,27,31\}}(8)$	$X_{22,\{8,14,15,20\sim22,27\sim29\}}(9)$, $Y_{22,\{0,6,7,12\sim14,18\sim21,25\sim28\}}(14)$	$2^{65}\cdot2^{33}\cdot\frac{14}{32\cdot26}\approx2^{92.11}$
(9)	$K^*_{22,\{0,7,14,21,28\}}(5)$	$X_{21,\{6,12,13,19,20,27,28\}}(7)$, $Y_{21,\{8,15,22,29\}}(4)$, $X_{22,\{14,21,28\}}(3)$	$2^{70}\cdot2^{23}\cdot\frac{4}{32\cdot26}\approx2^{85.30}$
(10)	$K^*_{22,\{6,13,20,27\}}(4)$	$X_{21,\{12,16,18,19,22,23,25,26,29,30\}}(10)$, $Y_{21,\{14,21,28\}}(3)$, $X_{22,\{20,27\}}(2)$	$2^{74}\cdot2^{14}\cdot\frac{3}{32\cdot26}\approx2^{79.88}$
(11)	$K^*_{22,\{12,19,26\}}(4)$	$X_{21,\{16,22,23,29,30\}}(5)$, $Y_{21,\{8,14,15,20\sim22,27\sim29\}}(9)$	$2^{77}\cdot2^{15}\cdot\frac{9}{32\cdot26}\approx2^{85.47}$
(12)	$K^*_{21,\{8,15,22,29\}}(4)$	$X_{20,\{14,20,21,24,27,28,31\}}(7)$, $Y_{20,\{16,23,30\}}(3)$, $X_{21,\{22,29\}}(2)$	$2^{81}\cdot2^{14}\cdot\frac{3}{32\cdot26}\approx2^{86.88}$
(13)	$K^*_{21,\{14,21,28\}}(3)$	$X_{20,\{24,31\}}(2)$, $Y_{20,\{16,22,23,29,30\}}(5)$	$2^{84}\cdot2^{12}\cdot\frac{5}{32\cdot26}\approx2^{88.62}$
(14)	$K^*_{20,\{16,23,30\}}(3)$	$X_{19,\{0\}}(1)$, $Y_{19,\{24,31\}}(2)$	$2^{87}\cdot2^{7}\cdot\frac{2}{32\cdot26}\approx2^{85.30}$
(15)	$K^*_{19,\{24,31\}}(2)$	$\left(X_{18,\{31\}}\wedge X_{18,\{24\}}\right)\oplus X_{19,\{0\}}(1)$	$2^{89}\cdot2^{3}\cdot\frac{1}{32\cdot26}\approx2^{82.30}$

SIMON64/128算法积分分析总的计算复杂度约为$2^{92.31}+2^{71.92}+2^{101+26}\left(1+2^{-64}\right)\approx2^{127}$次26轮算法加密，存储复杂度约为$2^{95.5}+2^{75.13}\approx2^{95.5}$ bit，约为2^{93} Byte。

5　结束语

本文考虑了SIMON64算法的积分分析，先利用中间相遇和部分和技术给出了25轮SIMON64/128算法的积分分析，接着利用等价密钥技术实现了更高一轮的积分分析。类似的方法，同样可以考虑更高分组长度的SIMON算法的积分分析，本文及对更高分组长度SIMON算法的攻击结果见表5。结合已有的攻击结果可以看出，随着SIMON算法版本的提高，积分分析能攻击的轮数并没有明显提高，相对而言它们具有更强的抵抗积分分析的能力。

表 4　计算$\oplus M_2$值的复杂度

步骤	猜测密钥(比特数)	统计状态(比特数)	时间复杂度
(1)	–	$X_{25,\{2\sim4,8\sim11,14\sim18,20\sim24,28,29\}}(19)$, $Y_{25,\{0\sim3,6\sim10,12\sim23,26\sim28\}}(24)$	$2^{63}\cdot\frac{24}{32\cdot26}\approx2^{57.89}$
(2)	$K^*_{25\{0\sim3,6\sim10,12\sim17,19\sim23,26\sim28\}}(22)$	$X_{24,\{4,5,10\sim12,16\sim19,22\sim25,30\}}(14)$, $Y_{24,\{2\sim4,8\sim11,14\sim18,20\sim24,28,29\}}(19)$	$2^{22}\cdot2^{43}\cdot\frac{19}{32\cdot26}\approx2^{59.55}$
(3)	$K^*_{24\{2\sim4,8\sim11,14\sim18,21\sim24,28,29\}}(18)$	$X_{23,\{6,12,13,18\sim20,24\sim27\}}(10)$, $Y_{23,\{4,5,10\sim12,16\sim19,22\sim25,30\}}(14)$	$2^{40}\cdot2^{33}\cdot\frac{14}{32\cdot26}\approx2^{67.11}$
(4)	$K^*_{23,\{4,5,10\sim12,16\sim19,23\sim25,30\}}(13)$	$X_{22,\{14,20,21,26\sim28\}}(6)$, $Y_{22,\{6,12,13,18\sim20,24\sim27\}}(10)$	$2^{53}\cdot2^{24}\cdot\frac{10}{32\cdot26}\approx2^{70.63}$
(5)	$K^*_{22\{6,12,13,18\sim20,25\sim27\}}(9)$	$X_{21,\{22,28,29\}}(3)$, $Y_{21,\{14,20,21,26\sim28\}}(6)$	$2^{62}\cdot2^{16}\cdot\frac{6}{32\cdot26}\approx2^{70.89}$
(6)	$K^*_{21\{14,20,21,27,28\}}(5)$	$X_{20,\{30\}}(1)$, $Y_{20,\{22,28,29\}}(3)$	$2^{67}\cdot2^{9}\cdot\frac{3}{32\cdot26}\approx2^{67.89}$
(6)	$K^*_{20\{22,29\}}(2)$	$\oplus Y_{19,\{30\}}(1)$	$2^{69}\cdot2^{4}\cdot\frac{1}{32\cdot26}\approx2^{63.30}$

表 5　SIMON算法的积分分析(分组长度64/96/128-bit)

算法	区分器轮数	数据量(CP)	攻击轮数	猜测密钥量(bit)	攻击复杂度(E)
SIMON64/96	18	2^{63}	25	73	2^{95}
SIMON64/128	18	2^{63}	26	102	2^{127}
SIMON96/96	22	2^{95}	28	64	2^{95}
SIMON96/144	22	2^{95}	30	138	2^{95}
SIMON128/128	26	2^{127}	33	98	2^{127}
SIMON128/192	26	2^{127}	35	187	2^{127}
SIMON128/256	26	2^{127}	36	241	2^{127}

参 考 文 献

[1] KNUDSEN L and WAGNER D. Integral cryptanalysis[C]. The 9th International Workshop on Fast Software Encryption, Leuven, Belgium, 2002: 112–127.

[2] DAEMEN J, KNUDSEN L, and RIJMEN V. The block cipher Square[C]. The 4th International Workshop on Fast Software Encryption, Haifa, Israel, 1997: 149–165.

[3] FERGUSON N, KELSEY J, LUCKS S, *et al*. Improved cryptanalysis of rijndael[C]. The 7th International Workshop on Fast Software Encryption, New York, USA, 2001: 213–230.

[4] TODO Y. Integral cryptanalysis on full MISTY1[C]. The 35th Annual Cryptology Conference, Santa Barbara, USA, 2015: 413–432.

[5] BEAULIEU R, SHORS D, SMITH J, *et al*. The SIMON and SPECK families of lightweight block ciphers[EB/OL]. https: //eprint.iacr.org/2013/404, 2013.

[6] ABED F, LIST E, LUCKS S, *et al*. Differential cryptanalysis of round-reduced SIMON and SPECK[C]. The 21st International Workshop on Fast Software Encryption, London, UK, 2015: 525–545.

[7] BIRYUKOV A, ROY A, and VELICHKOV V. Differential analysis of block ciphers SIMON and SPECK[C]. The 21st International Workshop on Fast Software Encryption, London, UK, 2015: 546–570.

[8] KÖLBL S, LEANDER G, and TIESSEN T. Observations on the SIMON block cipher family[C]. The 35th Annual Cryptology Conference, Santa Barbara, USA, 2015: 161–185.

[9] QIAO Kexin, HU Lei, and SUN Siwei. Differential analysis

on simeck and simon with dynamic key-guessing techniques[C]. The 2nd International Conference on Information Systems Security and Privacy, Rome, Italy, 2017: 64–85.

[10] LIU Zhengbin, LI Yongqiang, and WANG Mingsheng. Optimal differential trails in SIMON-like ciphers[J]. *IACR Transactions on Symmetric Cryptology*, 2017(1): 358–379. doi: 10.13154/tosc.v2017.i1.358-379.

[11] WANG Ning, WANG Xiaoyun, JIA Keting, *et al*. Differential attacks on reduced SIMON versions with dynamic key-guessing techniques[J]. *Science China Information Sciences*, 2018, 61(9): 098103. doi: 10.1007/s11432-017-9231-5.

[12] ALIZADEH J, ALKHZAIMI H A, AREF M R, *et al*. Cryptanalysis of SIMON variants with connections[C]. The 10th International Workshop on Radio Frequency Identification: Security and Privacy Issues, Oxford, United Kingdom, 2014: 90–107.

[13] ABDELRAHEEM N A, ALIZADEH J, ALKHZAIMI H A, *et al*. Improved linear cryptanalysis of reduced-round SIMON[EB/OL]. https: //eprint.iacr.org/2014/681, 2014.

[14] CHEN Huaifeng and WANG Xiaoyun. Improved linear hull attack on round-reduced Simon with dynamic key-guessing techniques[C]. The 23rd International Conference on Fast Software Encryption, Bochum, Germany, 2016: 428–449.

[15] WANG Qingju, LIU Zhiqiang, VARICI K, *et al*. Cryptanalysis of reduced-round SIMON32 and SIMON48[C]. The 15th International Conference on Cryptology in India, New Delhi, India, 2014: 143–160.

[16] BOURA C, NAYA-PLASENCIA M, and SUDER V. Scrutinizing and improving impossible differential attacks: Applications to CLEFIA, Camellia, LBlock and Simon[C]. The 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, China, 2014: 179–199.

[17] 陈展, 王宁. SIMON算法的不可能差分分析[J]. 密码学报, 2015, 2(6): 505–514. doi: 10.13868/j.cnki.jcr.000097.
CHEN Zhan and WANG Ning. Impossible differential cryptanalysis of reduced-round SIMON[J]. *Journal of Cryptologic Research*, 2015, 2(6): 505–514. doi: 10.13868/j.cnki.jcr.000097.

[18] YU Xiaoli, WU Wenling, SHI Zhenqing, *et al*. Zero-correlation linear cryptanalysis of reduced-round SIMON[J]. *Journal of Computer Science and Technology*, 2015, 30(6): 1358–1369. doi: 10.1007/s11390-015-1603-5.

[19] XIANG Zejun, ZHANG Wentao, BAO Zhenzhen, *et al*. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers[C]. The 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, 2016: 648–678.

[20] FU Kai, SUN Ling, and WANG Meiqin. New integral attacks on SIMON[J]. *IET Information Security*, 2017, 11(5): 277–286. doi: 10.1049/iet-ifs.2016.0241.

[21] CHU Zhihui, CHEN Huaifeng, WANG Xiaoyun, *et al*. Improved integral attacks on SIMON32 and SIMON48 with dynamic key-guessing techniques[J]. *Security and Communication Networks*, 2018: 5160237. doi: 10.1155/2018/5160237.

徐 洪：女，1979年生，硕士生导师，主要研究方向为对称密码的设计与分析.

方玉颖：男，1994年生，硕士生，研究方向为分组密码分析.

戚文峰：男，1963 年生，教授，主要研究方向为对称密码的设计与分析.