轻量级密码算法 MIBS 的零相关和积分分析

伊文坛* 鲁林真 陈少真
(数学工程与先进计算国家重点实验室 郑州 450001)

摘 要：MIBS 是适用于 RFID 和传感资源受限环境的轻量级分组算法。该文构造了一些关于 MIBS 的 8 轮零相关线性逼近，结合密钥扩展算法的特点和部分和技术，对 13 轮 MIBS-80 进行了多维零相关分析。该分析大体需要 $2^{62.1}$ 个已知明文和 $2^{74.9}$ 次加密。此外，利用零相关线性逼近和积分区分器之间的内在联系，推导出 8 轮的积分区分器，并且对 11 轮的 MIBS-80 进行了积分攻击，大体需要 2^{60} 个选择明文和 $2^{59.8}$ 次加密。

关键词：分组密码；MIBS；零相关分析；积分攻击
中图分类号： TP309 **文献标识码**：A **文章编号**：1009-5896(2016)04-0819-08
DOI: 10.11999/JEIT150498

Integral and Zero-correlation Linear Cryptanalysis of Lightweight Block Cipher MIBS

YI Wentan LU Linzhen CHEN Shaozhen
(*State Key Laboratory of Mathematical Engineering and Advanced Computing*, *Zhengzhou* 450001, *China*)

Abstract: MIBS is a light weight block cipher for constrained resources environments such as RFID tags and sensor networks. This paper investigates the construction of zero-correlation linear approximations of 8-round MIBS and presents an attack on 13-round MIBS-80 by means of zero-correlation linear cryptanalysis with the properties of key schedule and partial-sum technique, which needs $2^{62.1}$ known plaintexts and $2^{74.9}$ encryptions. Furthermore, an 8-round integral distinguisher is deduced from the zero-correlation linear approximations using the relations between them, and as an application, integral attack on 11-round MIBS-80 is conducted with 2^{60} chosen plaintexts and $2^{59.8}$ encryptions.

Key words: Block cipher; MIBS; Zero-correlation linear cryptanalysis; Integral attack

1 引言

MIBS[1]是在 2009 年提出的一个轻量级分组密码算法，具有资源占用量较少的优点，主要适用于 RFID(Radio Frequency IDentification)、无线传感技术等设备资源和计算能力有限的设备和环境中。该算法整体采用 Feistel 结构，分组长度为 64 bit，密钥长度可以为 64 bit 和 80 bit，分别记作 MIBS-64 和 MIBS-80，都迭代 32 轮。目前针对 MIBS 的分析有差分分析、线性分析、不可能差分分析、积分分析、中间相遇分析以及相关密钥条件下的不可能差分分析等。

文献[2]给出了 13 轮 MIBS-64 的差分分析；之后，文献[3]改进了关于 14 轮 MIBS-64 的差分分析结果，需要的时间复杂度为 $2^{37.2}$ 次加密，数据复杂度为 2^{40} 个选择明文；文献[3]对 MIBS 算法的抗线性分析的能力进行了估计，结果显示对 18 轮 MIBS-80 的线性分析大体需要 $2^{60.9}$ 个已知明文和 $2^{76.1}$ 次加密；文献[3]给出了 12 轮 MIBS-80 的不可能差分分析。随后文献[4]指出文献[3]工作中存在错误，并重新给出了 12 轮 MIBS-80 的不可能差分分析结果；文献[5]首次利用积分分析方法分析了 8 轮 MIBS-64 和 9 轮 MIBS-80；随后，文献[6]和文献[7]分别给出了 10 轮 MIBS-80 关于积分分析的结果；2013 年，文献[8]发现了 MIBS 的 6 轮中间相遇区分器，结合密钥扩展算法的特点，给出了 11 轮 MIBS-80 的中间相遇分析结果，大体需要 $2^{24.9}$ 个选择明文，$2^{66.2}$ 次加密和 2^{50} 次预计算；2014 年，文献[9]构造了相关密钥条件下的 10 轮的不可能差分特征，并对 14 轮 MIBS-80 进行了攻击，大体需要 2^{54} 个选择明文和 2^{56} 次加密。

可以看出，评估 MIBS 算法的安全性一直是研究的热点问题。然而，MIBS 算法关于最近提出的零相关分析方法的分析结果还是空白。零相关线性分析方法由文献[10]在 2012 年提出。该方法是利用相关系数为零的线性逼近来区分密码算法和随机函

收稿日期：2015-04-30；改回日期：2016-01-06；网络出版：2016-02-18
*通信作者：伊文坛 nlwt89@sina.com

数，从而进行密钥恢复攻击的工作。最初，零相关线性分析方法利用一条零相关线性逼近，至少需要选择一半明密文空间。多重零相关[11]利用多条零相关线性逼近，在一定程度上降低了数据量，但是要求线性逼近的输入掩码和输出掩码独立。多维零相关[12]的提出克服了线性逼近的独立性条件，所需数据量和多重零相关相当。最近，零相关线性分析方法在 AES[10], TEA[11], CAST-256[12], LBlock[13], E2[14], Camellia[15] 以及 CLEFIA[15]等密码算法的分析中取得了很好的结果。另外，文献[12]还揭示了零相关线性逼近和积分区分器之间的关系。

本文利用零相关线性分析和积分分析方法评估 MIBS-80 密码算法的安全性。根据线性层的特点，构造了一些 8 轮 MIBS 密码算法的零相关线性逼近；利用多维零相关线性分析方法，结合密钥扩展算法的特点和部分和技术，对 13 轮的 MIBS-80 密码算法做安全性分析；利用零相关区分器和积分区分器之间的联系，推导出一些 8 轮的积分区分器；进一步，作为应用，对 11 轮的 MIBS-80 密码算法进行积分分析。

本文的结构大致安排如下：在第 2 节中，约定一些记号，简单介绍 MIBS 密码算法、零相关分析方法和积分攻击方法；在第 3 节中，构造了一些 8 轮 MIBS 密码算法零相关线性逼近并对 13 轮 MIBS-80 进行了多维零相关分析；在随后的小节中，推导出一个 8 轮的积分区分器，并且利用积分分析方法分析了 MIBS-80 密码算法的安全性；最后，对比了单密钥下 MIBS-80 密码算法的主要分析结果并总结本文的工作。

2 预备知识

2.1 一些记号

$a \| b$：向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的连接；$\oplus$：异或运算；$>>> i$：循环右移 i bit；S：基于半个字节，也就是 4 bit 的 S 盒运算；F：轮函数；$a[i]$：表示向量 $\boldsymbol{a}$ 的第 $i+1$ bit，0 是最低位；$a[i-j]$：表示向量 $\boldsymbol{a}$ 的第 $i+1$ 到第 $j+1$ bit 共 $j-i+1$ bit 值，$j \geq i$；$K_i^{j,k\cdots}$：表示第 $i+1$ 轮轮子密钥的第 $j+1$，$k+1$ 个半字节；L_i：第 $i+1$ 轮输入的左半 32 bit；R_i：第 $i+1$ 轮输入的右半 32 bit；$L_i^{j,k\cdots}$：L_i 的第 $i+1, j+1, \cdots$ 个半字节；$R_i^{j,k\cdots}$：R_i 的第 $i+1, j+1\cdots$ 个半字节；$\boldsymbol{M}^{\mathrm{T}}$：矩阵 $\boldsymbol{M}$ 的转置。

2.2 MIBS 算法介绍

分组密码算法 MIBS 总体采用了 Feistel 密码结构，分组长度为 64 bit，并且迭代 32 轮。密钥长度接受 64 bit 和 80 bit，仅仅是在密钥扩展算法上有区别，分别记为 MIBS-64 和 MIBS-80。轮函数采用了典型的 SPN(Substitution Permutation Network)结构，包括轮子密钥加运算、S 盒变换和线性层运算。所有的操作都是在 4 bit 的半个字节上完成的。

MIBS 的迭代过程可以描述为，对于 i 从 1 到 32，

$$L_i = F(L_{i-1}; K_{i-1}) \oplus R_{i-1};\ R_i = L_{i-1}$$

其中，$P = L_0 \| R_0$ 和 $C = L_{32} \| R_{32}$ 分别表示明文和密文。

F 函数由轮子密钥加、S 盒变换、线性层组成。轮子密钥加操作是每轮输入的左 32 bit 都与 32 bit 长的轮子密钥相异或；S 盒变换是由 8 个相同的规模为 4×4 的 S 盒并置而成；线性层包括扩散层和置换，整个变换可由矩阵 $\boldsymbol{P}$ 来表示。矩阵 $\boldsymbol{P}$ 可以表示为

$$\boldsymbol{P}:\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$$

其中 $x_i, y_i, i = 0,1,\cdots,7$ 是 4 bit 长的值。

本文工作主要针对 MIBS-80，这里只介绍 MIBS-80 的密钥扩展算法。MIBS-80 的密钥扩展算法受到 PRESENT 算法[16]的密钥扩展算法的启发。设 $K = (K[0], K[1], \cdots, K[79])$ 为长度为 80 bit 长主密钥，则由主密钥生成 32 个 32 bit 的轮密钥 $K_i (0 \leq i \leq 31)$ 的过程如下：

(1)初始化 $\mathrm{state}^{-1} \leftarrow K$；

(2)对于 $i = 0,1,\cdots,31$，做如下操作：

(a) $\mathrm{state}^i = \mathrm{state}^{i-1} >>> 19$; (b) $\mathrm{state}^i = S(\mathrm{state}^i_{[0-3]}) \| S(\mathrm{state}^i_{[4-7]}) \| \mathrm{state}^i_{[7-79]}$;

(c) $\mathrm{state}^i = (\mathrm{state}^i_{[0-59]}) \,||\, (\mathrm{state}^i_{[60-63]}) \oplus \mathrm{Round\_counter} \,||\, \mathrm{state}^i_{[64-79]}$；(d)输出 $K_i = \mathrm{state}^i_{[0-31]}$。

其中，轮常数 Round_counter 是所在轮序号的 4 bit 二进制表示。关于 MIBS-64 的密钥扩展算法见文献[1]。

2.3 多维零相关分析方法和积分分析介绍

2.3.1 多维零相关分析方法 对于给定的 n bit 长的输入掩码 α 和输出掩码 β 以及 F_2^n 上的函数 f，我们定义相应的线性逼近为

$$(\alpha \xrightarrow{f} \beta)$$

简单记为 $(\alpha \to \beta)$。进一步，该线性逼近的相关系

数则定义为

$$C_f(a,\beta)=\mathrm{Cor}_x(\beta\cdot f(x)\oplus a\cdot x)$$
$$=2\mathrm{Pr}_x(\beta\cdot f(x)\oplus a\cdot x=0)-1$$

若$C_f(a,\beta)=0$，则称该线性线性逼近是零相关线性逼近。

多维零相关线性分析方法[12]选取$\ell=2^m$个零相关线性逼近，并且这些线性逼近由m的基线向量通过线性关系扩展。不妨记作为$(a_i\to b_i)_{i=0,1,\cdots,m-1}$。攻击者选择$N$对明密文对，并建立计数器$N[\boldsymbol{z}]$，其中$\boldsymbol{z}$是$m$ bit长的向量。通过在线性逼近前后加轮，穷举部分子密钥并部分加解密所选取的明密文对，得到线性掩码首尾对应的中间状态，为了方便记作(p',c')。然后, 对于选择的每一个明密文对经过部分加解密得到(p',c')，计算$z=(z[0],z[1],\cdots,z[m-1])=(a_0\cdot p'\oplus b_0\cdot c',a_1\cdot p'\oplus b_1\cdot c',\cdots,a_{m-1}\cdot p\oplus b_{m-1}\cdot c')$，进而，得到向量$\boldsymbol{z}$的值。更新相应的对应的计数器$N[\boldsymbol{z}]$。然后，计算统计量$T$：

$$T=\sum_{z=0}^{2^m-1}\frac{(N[\boldsymbol{z}]-N2^{-m})^2}{N2^{-m}(1-2^{-m})}\approx N2^m\sum_{z=0}^{2^m-1}\left(\frac{N[\boldsymbol{z}]}{N}-\frac{1}{2^m}\right)^2 \quad (1)$$

若所猜测的密钥为正确密钥，则统计量T服从期望为$\mu_0=(n-1)\dfrac{2^n-N}{2^n-1}$，方差为$\sigma_0^n=2(n-1)\cdot\left(\dfrac{2^n-N}{2^n-1}\right)^2$的正态分布。若猜测的密钥是错误密钥，则统计量$T$服从期望$\mu_1=\ell-1$，方差为$\sigma_1^2=2(\ell-1)$的正态分布。若把正确密钥当作错误密钥的概率为$\alpha$，把错误密钥当作正确密钥的概率为$\beta$。判定条件$T<\tau$，其中$\tau=u_0+\sigma_0z_{1-\alpha}=u_1-\sigma_0z_{1-\beta}$，则攻击所需要的明密文对大体为

$$N=\frac{(2^n-1)(z_{1-\alpha}+z_{1-\beta})}{\sqrt{(\ell-1)/2}+z_{1-\alpha}}+1 \quad (2)$$

其中，n是分组长度，$z_{1-\alpha}$，$z_{1-\beta}$为标准正态分布的分位数。具体请参考文献[12]。

2.3.2 积分分析方法　积分分析是一种选择明文攻击方法，主要利用积分区分器来恢复密钥信息。积分区分器是根据轮函数的性质构造而成的。比如选择某些特定结构的明文，经过若干轮迭代之后，所有对应的输出的某些或者全部bit异或之后为零等。不妨设$r=r_1+r_2$轮密码算法$E(\cdot,K)$的前r_1轮$E_{r_1}(\cdot,K_{r_1})$是满足上述性质的区分器，设Ω是相应的选择明文空间，则有$\oplus_{p\in\Omega}E_{r_1}(p,K_{r_1})=0$。

在具体攻击过程中，攻击者在区分器的一端或者两端添加若干轮，猜测涉及到的子密钥，在根据区分器的性质进行密钥筛选。不妨在后面添加了剩余的r_2轮，Θ表示Ω对应的密文空间，攻击者穷举涉及到的子密钥K_{r_2}，计算并判断$\oplus_{c\in\Theta}E_{r_2}^{-1}(c,K_{r_2})=0$，是否成立。若成立，则认为穷举的密钥是正确密钥；这一小节中，K_r表示r轮迭代中涉及到的轮子密钥。

3　MIBS 密码 8 轮零相关线性逼近和密钥扩展算法的一些性质

本节主要介绍构造关于MIBS的8轮零相关线性逼近和密钥扩展算法的一些性质。对密码算法做分析是在零相关线性逼近的基础上往前和往后添加轮数，并做部分加解密，在此过程中应该最大限度地利用密钥扩展算法的性质减少密钥的穷举量，进而减少复杂度。这就涉及到寻找合适的零相关线性逼近的问题。通过分析，我们构造下面的零相关线性逼近。

3.1 MIBS 算法 8 轮零相关线性逼近

本文主要通过线性掩码在相关系数非零的条件下从前和从后两个方向从中间传播，最后在中间某个位置相遇，并且产生相关系数为零的矛盾状态的方式来构造零相关线性逼近。在非线性部件，比如S盒等部件，线性掩码的传播有下面的规律。

命题 1[10]　设h是可逆函数，输入掩码为α，输出掩码为β。若$C_h(\alpha,\beta)\neq 0$，则可得α，β同时为零或者同时不为零。

命题 2[10]　设$\boldsymbol{M}$是一个矩阵并且线性函数定义为$h(x)=\boldsymbol{M}x$。若输入掩码为α，输出掩码为β，则$C_h(\alpha,\beta)\neq 0$当且仅当$\alpha=\boldsymbol{M}^{\mathrm{T}}\beta$。

上面两个命题给出了非零相关系数条件下，线性掩码在非线性和线性部件的传播规律。利用这些规律，我们可以得到关于MIBS的8轮的零相关线性逼近。

定理 1　设a和h分别是4 bit长的非零向量，则

$$(a,a,0,a;a,a,0,a;0,0,0,0;0,0,0,0)$$
$$\longrightarrow(0,0,0,0;0,0,0,0;0,h,h,0;h,h,h,0)$$

是r轮到$r+7$轮的8轮零相关线性逼近。见图1。

证明　如图1所示，从加密方向，若第r轮的输入掩码为$(a,a,0,a;a,a,0,a;0,0,0,0;0,0,0,0)$，经过4轮MIBS密码迭代，非零相关系数条件下输出掩码右32 bit为$(c_0\oplus a,c_1\oplus a,0,c_3\oplus a,c_4\oplus a,c_5\oplus a,0,a)$。其中$c_i$，$i=0,1,3,4,5$，是非零4 bit的值。进一步，线性掩码在第$r+4$轮的轮函数传播，第3和第7个S盒的输入掩码为0，从命题1可知，若相关系数非零，则相应的输出掩码也分别为0。

从解密方向，若第$r+7$轮的输出掩码为$(0,0,0,0;0,0,0,0;0,h,h,0;h,h,h,0)$，经过3轮MIBS密码迭代，非零相关系数条件下输入掩码的左32 bit为

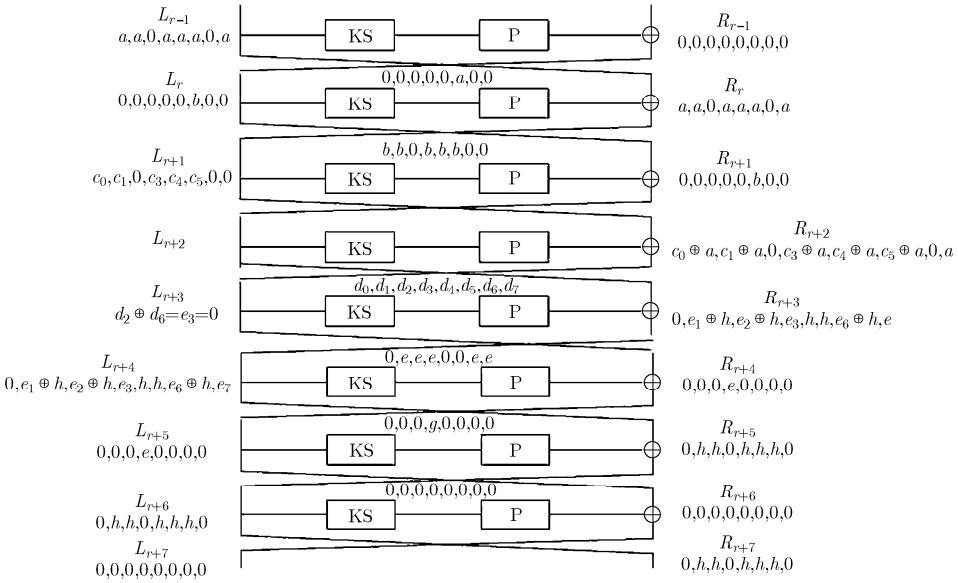

图 1 MIBS 算法 8 轮零相关线性逼近

$(0,e_1\oplus h,e_2\oplus h,e_3,h,h,e_6\oplus h,e_7)$，其中 e_i，$i=1,2,3,6,7$ 是非零 4 bit 的值。同样，线性掩码在第 $r+4$ 轮的轮函数传播的过程中，记该轮函数中线性层 P 的输入掩码为 $(d_0,d_1,d_2,d_3;d_4,d_5,d_6,d_7)$。由命题 2 可得，

$$d_2=e_1\oplus e_2\oplus e_3\oplus e_6\oplus e_7;\ d_6=e_1\oplus e_3\oplus e_6\oplus e_7$$

则 $d_2\oplus d_6=e_2\neq 0$，这与加密方向传播过程中 $d_2=0$，$d_6=0$ 这一状态相矛盾。 证毕

3.2 MIBS-80 密钥扩展算法的一些性质

本节主要介绍 MIBS-80 算法的密钥扩展算法的一些特点。该密钥扩展算法受到 PRESENT 算法的密钥扩展算法的影响。在密钥生成的过程中，若不计移位操作，每一轮仅改变 12 bit 主密钥，所以轮子密钥之间存在很多关系。充分利用轮子密钥之间的关系，减少攻击过程中的需要猜测的密钥量，从而减少复杂度。

命题 3[7] 对于 MIBS-80 而言，若需要猜测轮子密钥 K_i^j，则只要猜测 4 bit 主密钥 $K[a-(a+3)]$，其中 $a=(4j-19(i+1))\bmod 80$。若 $a=77,78,79$ 时，则 $K[a-(a+3)]$ 分别取值为 $K[a]$，$K[(a+1)\bmod 80]$，$K[(a+2)\bmod 80]$ 和 $K[(a+3)\bmod 80]$ 这 4 bit 值。

命题 3 结论是正确的，但是文献[7]的证明不太清楚。异或轮常数运算输出的每个 bit 仅仅和输入的相应 bit 有关系，所以没有扩散效果，然而 S 盒是每一个输出 bit 都和输入的每一个 bit 有关，也就具有一定的扩散效果。但是由于主密钥长度 80 bit 和循环左移位参数 19 bit 的设置，每经过 4 轮移位打乱的 4 bit 半个字又重新组成半个字。然而做过 S 盒变换的半字节必须经过 4 轮的倍数(包括 4)之后才有可能再做 S 盒操作。所以扩散也仅仅在最初分组的 4 bit 半字节中。

从密钥扩展算法和上面的命题，我们可以得到下面关于轮子密钥之间的关系。

推论 1 记 S，S^{-1} 为 S 盒运算和 S 盒逆运算，则下面的关系成立：

(1) $K_{12}^0=S(S(K_0^3\oplus\{0110\}))$

(2) $K_{12}^1=S(K_0^4)$

(3) $K_{12}^2[0-2]=K_0^5[0-2]$

(4) $K_0^5[3]=S^{-1}(K_{12}^2[3]\,\|\,K_{12}^3[0-2])[0]$

(5) $K_1^5=K_0^0[1-3]\,\|\,K_0^1[0]$

(6) $K_{10}^3[0]=K_{11}^7[3]$

(7) $K_{11}^1=K_{12}^5[3]\,\|\,K_{12}^6[0-2]$

(8) $K_{11}^2=K_{12}^6[3]\,\|\,K_{12}^7[0-2]$

(9) $K_{11}^3[0]=K_{12}^7[3]$

4 13 轮 MIBS-80 的多维零相关分析

本节主要利用上面构造的 8 轮(3–10)零相关线性逼近，往前扩展 2 轮并且往后扩展 3 轮，结合轮子密钥之间的关系和部分和技术，对 13 轮 MIBS-80 做多维零相关线性分析，如图 2 所示。具体攻击过程如下：

(1)建立 8 bit 的计数器 $\mathrm{N}_0[x_0]$，且全部置为零。取 $x_0=L_0^{0,1,3,4,5}\,\|\,R_0^5\,\|\,L_{13}^{1,2,3,6,7}\,\|\,R_{13}^{0,1,2,3,4,5,6,7}\,\|\,I_1\,\|\,I_2$；收集 N 个明文及对应的密文，并计算 $I_1=\oplus_{i=0,1,3,4,5,7}\cdot L_0^i$；$I_2=\oplus_{i=1,2,4,5,6}L_{13}^i$，相应的计数器 $\mathrm{N}_0[x_0]$ 加 1。不超过 2^{64} 个明密文对分成 2^{84} 个状态，所以计数器 8 bit 够用。

(2)建立计数器 $\mathrm{N}_1[x_1]$，且全部置为零。取 $x_1=$

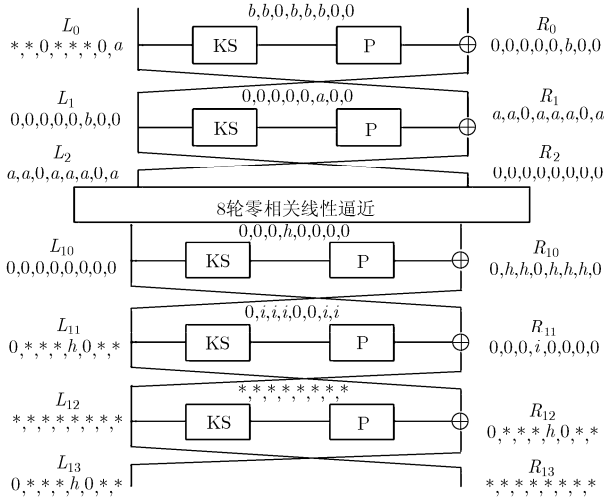

图 2 13 轮 MIBS-80 的多维零相关分析

$L_0^{0,1} \| R_0^5 \| L_{13}^{1,2,3,6,7} \| R_{13}^{3,4,5,6,7} \| I_1 \| I_2$；穷举 13 bit 轮子密钥 $K_0^{3,4,5}$ 和 $K_{12}^2[3]$，并且由推论 1 得到 12 bit 密钥 $K_{12}^{0,1,2}$，更新 $R_0^5 = R_0^5 \oplus S(L_0^3 \oplus K_0^3) \oplus S(L_0^4 \oplus K_0^4) \oplus S(L_0^5 \oplus K_0^5)$; $L_{13}^1 = L_{13}^1 \oplus S(R_{13}^1 \oplus K_{12}^1) \oplus S(R_{13}^2 \oplus K_{12}^2)$; $L_{13}^3 = L_{13}^3 \oplus S(L_{13}^1 \oplus K_{12}^1) \oplus S(L_{13}^2 \oplus K_{12}^2)$; $L_{13}^6 = L_{13}^6 \oplus S(R_{13}^0 \oplus K_{12}^0) \oplus S(R_{13}^1 \oplus K_{12}^1) \oplus S(R_{13}^2 \oplus K_{12}^2)$; $L_{13}^7 = L_{13}^7 \oplus S(R_{13}^0 \oplus K_{12}^0) \oplus S(R_{13}^2 \oplus K_{12}^2)$，然后更新计数器 $\mathrm{N}_1[x_1] += \mathrm{N}_0[x_0]$。此步大体需要 $N \times 2^{13}$ 次内存访问。

(3)建立计数器 $\mathrm{N}_2[x_2]$，且全部置为零。取 $x_2 = L_0^1 \| R_0^5 \| L_{13}^{1,2,3,6,7} \| R_{13}^{3,4,5,6,7} \| I_1 \| I_2$;穷举 4 bit 轮子密钥 K_0^0，计算并更新 $R_0^5 = R_0^5 \oplus S(L_0^0 \oplus K_0^0)$；然后更新计数器 $\mathrm{N}_2[x_2] += \mathrm{N}_1[x_1]$。此步需要 $2^{60} \times 2^{13} \times 2^4$ 次内存访问。

(4)建立计数器 $\mathrm{N}_3[x_3]$，且全部置为零。取 $x_3 = L_{13}^{1,2,3,6,7} \| R_{13}^{3,4,5,6,7} \| I_1 \| I_2$；穷举 4 bit 轮子密钥 K_0^1，并且推导出 K_1^5，计算并更新 $I_1 = I_1 \oplus S(R_0^5 \oplus S(L_0^1 \oplus K_0^1) \oplus K_2^5)$；然后更新计数器 $\mathrm{N}_3[x_3] += \mathrm{N}_2[x_2]$。此步大体需要 $2^{56} \times 2^{17} \times 2^4$ 次内存访问。

(5)建立计数器 $\mathrm{N}_4[x_4]$，且全部置为零。取 $x_4 = L_{13}^{1,2,3,6,7} \| R_{13}^{3,4,6,7} \| I_1 \| I_2$；穷举 4 bit 轮子密钥 K_{12}^5，对于 $i = 1,2,6,7$，计算并更新 $L_{13}^i = L_{13}^i \oplus S(R_{13}^5 \oplus K_{12}^5)$；然后更新计数器 $\mathrm{N}_4[x_4] += \mathrm{N}_3[x_3]$。此步大体需要 $2^{48} \times 2^{21} \times 2^4$ 次内存访问。

(6)建立计数器 $\mathrm{N}_5[x_5]$，且全部置为零。取 $x_5 = L_{13}^{1,2,3,6,7} \| R_{13}^{3,4,7} \| I_1 \| I_2$；穷举 4 bit 轮子密钥 K_{12}^6，对于 $i = 1,3,6,7$，计算并更新 $L_{13}^i = L_{13}^i \oplus S(R_{13}^6 \oplus K_{12}^6)$；然后更新计数器 $\mathrm{N}_5[x_5] += \mathrm{N}_4[x_4]$。此步大体需要 $2^{44} \times 2^{25} \times 2^4$ 次内存访问。

(7)建立计数器 $\mathrm{N}_6[x_6]$，且全部置为零。取 $x_6 = L_{13}^{1,2,3,6,7} \| R_{13}^{3,4} \| I_1 \| I_2$;穷举 4 bit 轮子密钥 K_{12}^7，对于 i=2,3,7，计算并更新 $L_{13}^i = L_{13}^i \oplus S(R_{13}^7 \oplus K_{12}^7)$；然后更新计数器 $\mathrm{N}_6[x_6] += \mathrm{N}_5[x_5]$。此步大体需要 $2^{40} \times 2^{29} \times 2^4$ 次内存访问。

(8)建立计数器 $\mathrm{N}_7[x_7]$，且全部置为零。取 $x_7 = L_{13}^{1,2,3,6,7} \| R_{13}^3 \| I_1 \| I_2$；穷举 4 bit 轮子密钥 K_{12}^4，对于 $i = 1,2$，计算更新 $L_{13}^i = L_{13}^i \oplus S(R_{13}^4 \oplus K_{12}^4)$；然后更新计数器 $\mathrm{N}_7[x_7] += \mathrm{N}_6[x_6]$。此步需要 $2^{36} \times 2^{33} \times 2^4$ 次内存访问。

(9)建立计数器 $\mathrm{N}_8[x_8]$，且全部置为零。取 $x_8 = L_{13}^{3,6,7} \| I_1 \| I_2 \| I_3$，穷举 4 bit 轮子密钥 K_{12}^3，并且计算 $K_{11}^{1,2}$。计算并更新 $I_2 = I_2 \oplus S(R_{13}^3 \oplus K_{12}^3)$；$I_3 = S(L_{13}^1 \oplus S(R_{13}^3 \oplus K_{12}^3) \oplus K_{11}^1) \oplus S(L_{13}^2 \oplus K_{11}^2)$，然后更新计数器 $\mathrm{N}_8[x_8] += \mathrm{N}_7[x_7]$。此步大体需要 $2^{32} \times 2^{37} \times 2^4$ 次内存访问。

(10)建立计数器 $\mathrm{N}_9[x_9]$，且全部置为零。取 $x_9 = L_{13}^{6,7} \| I_1 \| I_2 \| I_3$；穷举 3 bit 轮子密钥 $K_{11}^3[1-3]$，并且计算 $K_{11}^3[0]$。计算并更新 $I_3 = I_3 \oplus S(R_{13}^3 \oplus K_{12}^3)$；然后更新计数器 $\mathrm{N}_9[x_9] += \mathrm{N}_8[x_8]$。此步大体需要 $2^{24} \times 2^{41} \times 2^3$ 次内存访问。

(11)建立计数器 $\mathrm{N}_{10}[x_{10}]$，且全部置为零。取 $x_{10} = L_{13}^7 \| I_1 \| I_2 \| I_3$；穷举 4 bit 轮子密钥 K_{11}^6，计算并更新 $I_3 = I_3 \oplus S(L_{13}^6 \oplus K_{11}^6)$；然后更新计数器 $\mathrm{N}_{10}[x_{10}] += \mathrm{N}_9[x_9]$。此步大体需要 $2^{20} \times 2^{44} \times 2^4$ 次内存访问。

(12)建立计数器 $\mathrm{N}_{11}[x_{11}]$，且全部置为零。取 $x_{11} = I_1 \| I_2$；穷举 7 bit 轮子密钥 K_{11}^7，$K_{10}^3[1-3]$，推出 $K_{10}^3[0]$，计算并更新 $I_2 = I_2 \oplus S(I_3 \oplus S(L_{13}^7 \oplus K_{11}^7) \oplus K_{10}^3)$；然后更新计数器 $\mathrm{N}_{11}[x_{11}] += \mathrm{N}_{10}[x_{10}]$。此步大体需要 $2^{16} \times 2^{48} \times 2^7$ 次内存访问。

(13)建立计数器 $\mathrm{N}[\boldsymbol{z}]$，且全部置为零。其中 z 是 8 bit 向量。对于 8 个长度为 8 bit 的基础向量 $\boldsymbol{z}_0, \cdots, \boldsymbol{z}_7$，比如 $\boldsymbol{z}_i$ 是第 $i+1$ 个 bit 为 1，其他 bit 为 0。计算 $\boldsymbol{z}[i] = z_i \cdot x_{11}$，$0 \le i \le 7$，计算出 $\boldsymbol{z}$，并且更新计数器 $\mathrm{N}[\boldsymbol{z}] += \mathrm{N}_{11}[x_{11}]$。根据式(1)，计算统计值 T，如果 $T < \tau$，则猜测轮子密钥为正确密钥。

(14)以上过程中猜测了 55 bit 密钥，然后穷举并验证其他 25 bit 密钥。

复杂度估计　在攻击过程中，我们设 $\alpha = 2^{-2.7}$，$\beta = 2^{-10}$。则 $z_{1-\alpha} \approx 1$，$z_{1-\beta} \approx 3.09$。又因为 $n = 64$，$\ell = 2^8$，则由式(2)可知大体需要 $2^{62.1}$ 个明密文对。判断的临界值为 $\tau \approx 2^{5.5}$。在本文中，我们假设访问一次内存的代价大约是一轮 MIBS 加密。由于步骤(2)~步骤(9)的复杂度占复杂度的主要部分，而它们复杂度之和不超过 $2^{77} \times 3 \div 13 \approx 2^{74.9}$ 次 13 轮 MIBS 加密。需要存储主要是第一步，需要大约 2^{84} Byte。

5 11 轮 MIBS-80 的积分分析

文献[12]揭示了零相关线性逼近和积分区分器之间的关系，并且利用推导出的积分区分器对 31 轮 Skipjack-BABABABA 密码算法做了积分分析。它们之间的数学联系可以概括成下面的定理。

定理 2[12] 设 $F_2^m=\{0,1\}^m$，函数 f 是 F_2^{4n} 上的向量布尔函数，并且 $a\in F_2^n$，$b\in F_2^{4n}$ 以及 n 是正整数，则下面两个条件等价。

(1)对于任何 $(a,0,a,0)\in F_2^{4n}$，

$$\mathrm{Cor}_{(x_1,x_2,x_3,x_4)}((a,0,a,0)\cdot(x_1,x_2,x_3,x_4)\oplus b\cdot f(x_1,x_2,x_3,x_4))=0$$

(2)对于任何常数 $c\in F_2^n$，

$$\mathrm{Cor}_{(x_1,x_2,x_1\oplus c,x_4)}(b\cdot f(x_1,x_2,x_1\oplus c,x_4))=0$$

推论 2 对于任何一个半字节常数 c，若第 r 轮的输入各个半字节分别跑遍并且满足 $c=\oplus_{i=0,1,3,4,5,7}L_{r-1}^i$，则 2^{60} 个选择明文经过 8 轮迭代之后输出的 4 bit $\oplus_{i=1,2,4,5,6}R_{r+7}^i$ 的 2^4 个状态恰好各出现 2^{56} 次。

若是随机情况下，推论 2 出现的概率为

$$\left(C_{2^{60}}^{2^{56}}\times C_{2^{60}-2^{56}}^{2^{56}}\times\cdots\times C_{2^{57}}^{2^{56}}\times C_{2^{56}}^{2^{56}}\right)\Big/\left(2^4\right)^{2^{60}}$$

是一个比 2^{-80} 小很多值。这也就是说在积分攻击过程中，猜测密钥如果是错误密钥，则不可能出现 2^4 个状态恰好各出现 2^{56} 次的情况。

下面我们给出 11 轮 MIBS-80 的积分分析。见图 3，其中 A 表示相应的半字节跑遍所有状态。具体攻击过程如下。

(1)建立 8 bit 的计数器 $\mathrm{V}_0[y_0]$，且全部置为零。取 $y_0=L_{11}^{1,2,3,6,7}\|R_{11}^{0,1,2,3,4,5,6,7}\|I_1$；选择 2^{60} 个明文及对应的密文满足各个半字节分别跑遍并且满足 $c=\oplus_{i=0,1,3,4,5,7}L_0^i$，并计 $I_1=\oplus_{i=1,2,4,5,6}R_{11}^i$；相应的计数器 $\mathrm{V}_0[y_0]$ 加 1。2^{60} 个明文密文对分成 2^{56} 个状态，所以计数器 8 bit 够用。

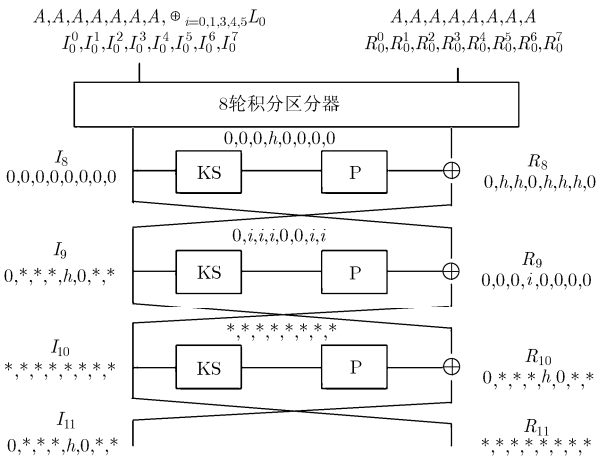

图 3 11 轮 MIBS-80 的积分分析

(2)建立计数器 $\mathrm{V}_1[y_1]$，且全部置为零。取 $y_1=L_{11}^{1,2,3,6,7}\|R_{11}^{1,2,3,4,5,6,7}\|I_1$；穷举 4 bit 轮子密钥 K_{10}^0，对于 i=2,6,7，计算并更新 L_{11}^i=$L_{11}^i\oplus S(R_{11}^0\oplus K_{10}^0)$；然后更新计数器 $\mathrm{V}_1[y_1]+=\mathrm{V}_0[y_0]$。此步大体需要 $2^{56}\times2^4$ 次内存访问。

(3)建立计数器 $\mathrm{V}_2[y_2]$，且全部置为零。取 $y_2=L_{11}^{1,2,3,6,7}\|R_{11}^{2,3,4,5,6,7}\|I_1$；穷举 4 bit 轮子密钥 K_{10}^1，对于 $i=1,2,3,6$，计算并更新 $L_{11}^i=L_{11}^i\oplus S(R_{11}^1\oplus K_{10}^1)$；然后更新计数器 $\mathrm{V}_2[y_2]+=\mathrm{V}_1[y_1]$。此步大体需要 $2^{52}\times2^4\times2^4$ 次内存访问。

(4)建立计数器 $\mathrm{V}_3[y_3]$，且全部置为零。取 $y_3=L_{11}^{1,2,3,6,7}\|R_{11}^{3,4,5,6,7}\|I_1$；穷举 4 bit 轮子密钥 K_{10}^2，对于 $i=1,2,3,6$，计算并更新 $L_{11}^i=L_{11}^i\oplus S(R_{11}^2\oplus K_{10}^2)$；然后更新计数器 $\mathrm{V}_3[x_3]+=\mathrm{V}_2[x_2]$。此步大体需要 $2^{48}\times2^8\times2^4$ 次内存访问。

(5)建立计数器 $\mathrm{V}_4[y_4]$，且全部置为零。取 $y_4=L_{11}^{1,2,3,6,7}\|R_{11}^{3,4,6,7}\|I_1$；穷举 4 bit 轮子密钥 K_{10}^4，对于 $i=1,2$，计算并更新 $L_{11}^i=L_{11}^i\oplus S(R_{11}^4\oplus K_{10}^4)$;然后更新计数器 $\mathrm{N}_4[x_4]+=\mathrm{N}_3[x_3]$。此步需要 $2^{44}\times2^{12}\times2^4$ 次内存访问。

(6)建立计数器 $\mathrm{V}_5[y_5]$，且全部置为零。取 $y_5=L_{11}^{1,2,3,6,7}\|R_{11}^{3,6,7}\|I_1$；穷举 4 bit 轮子密钥 K_{10}^5，对于 $i=1,2,6,7$，计算并更新 $L_{11}^i=L_{11}^i\oplus S(R_{11}^5\oplus K_{10}^5)$；然后更新计数器 $\mathrm{V}_5[y_5]+=\mathrm{V}_4[y_4]$。此步需要 $2^{40}\times2^{16}\times2^4$ 次内存访问。

(7)建立计数器 $\mathrm{V}_6[y_6]$，且全部置为零。取 $y_6=L_{11}^{1,2,3,6,7}\|R_{11}^{3,7}\|I_1$；穷举 4 bit 轮子密钥 K_{10}^6，对于 $i=1,3,6,7$，计算并更新 $L_{11}^i=L_{11}^i\oplus S(R_{11}^6\oplus K_{10}^6)$；然后更新计数器 $\mathrm{V}_6[y_6]+=\mathrm{V}_5[y_5]$。此步需要 $2^{36}\times2^{20}\times2^4$ 次内存访问。

(8)建立计数器 $\mathrm{V}_7[y_7]$，且全部置为零。$y_7=L_{11}^{1,2,3,6,7}\|R_{11}^3\|I_1$；穷举 4 bit 轮子密钥 K_{10}^7，对于 $i=2,3,7$，计算并更新 $L_{11}^i=L_{11}^i\oplus S(R_{11}^7\oplus K_{10}^7)$；更新计数器 $\mathrm{V}_7[y_7]+=\mathrm{V}_6[y_6]$。此步需要 $2^{32}\times2^{24}\times2^4$ 次内存访问。

(9)建立计数器 $\mathrm{V}_8[y_8]$，且全部置为零。取 $y_8=L_{11}^{3,6,7}\|I_1\|I_2$；穷举 4 bit 轮子密钥 K_{10}^3，并且计算 K_9^2。计算并更新 $I_1=I_1+S(R_{11}^3\oplus K_{10}^3)$；$I_2=S(R_{11}^1\oplus K_{10}^1)\oplus S(L_{11}^2\oplus K_9^2)$，然后更新计数器 $\mathrm{V}_8[y_8]+=\mathrm{V}_7[y_7]$。此步大体需要 $2^{28}\times2^{28}\times2^4$ 次内存访问。

(10)建立计数器 $\mathrm{V}_9[y_9]$，且全部置为零。取

$y_9 = L_{11}^{6,7} \| I_1 \| I_2$；穷举 3 bit 轮子密钥 $K_9^3[1-3]$，并且计算 $K_9^3[0]$。计算并更新 $I_2 = I_2 \oplus S(L_{11}^3 \oplus K_9^3)$；然后更新计数器 $\mathrm{V}_9[y_9] += \mathrm{V}_8[y_8]$。此步需要 $2^{20} \times 2^{32} \times 2^3$ 次内存访问。

(11)建立计数器 $\mathrm{V}_{10}[y_{10}]$，且全部置为零。取 $y_{10} = L_{11}^7 \| I_1 \| I_2$；穷举 4 bit 轮子密钥 K_9^6，计算并更新 $I_2 = I_2 \oplus S(L_{11}^6 \oplus K_9^6)$；然后更新计数器 $\mathrm{V}_{10}[y_{10}] += \mathrm{V}_9[y_9]$。此步大体需要 $2^{16} \times 2^{35} \times 2^4$ 次内存访问。

(12)建立计数器 $\mathrm{V}_{11}[y_{11}]$，且全部置为零。取 $y_{11} = I_1 \| I_2$；穷举 4 bit 轮子密钥 K_9^7，计算并更新 $I_2 = I_2 \oplus S(L_{11}^7 \oplus K_9^7)$；然后更新计数器 $\mathrm{V}_{11}[y_{11}] += \mathrm{V}_{10}[y_{10}]$。此步大体需要 $2^{12} \times 2^{39} \times 2^4$ 次内存访问。

(13)建立计数器 $\mathrm{V}_{12}[y_{12}]$，且全部置为零。取 $y_{12} = I_1$；穷举 3 bit 轮子密钥 $K_8^3[1-3]$，推导 $K_8^3[0]$ 出计算并更新 $I_1 = I_1 \oplus S(I_2 \oplus K_8^3)$；然后更新计数器。此步大体需要 $2^8 \times 2^{43} \times 2^3$ 次内存访问。

(14)到此为止，一共猜测了 46 bit 密钥。若是存在 $y_{13} \in F_2^4$，使得 $\mathrm{V}_{13}[y_{13}] \neq 2^{56}$，则认为所猜测的密钥是错误密钥。然后穷举其他 34 bit 密钥。

复杂度估计　攻击过程需要 2^{60} 个选择明密文对。需要存储主要是第(1)步，需要大约 2^{56} 字节。我们仍然假设访问一次内存的代价大约是一轮 MIBS 加密。由于(2)–(9)步的复杂度占复杂度的主要部分，而它们复杂度之和不超过 $2^{60} \times 10 \div 11 \approx 2^{59.8}$ 次 11 轮 MIBS 加密。

6　结束语

本文主要评估了 MIBS-80 密码算法关于多维零相关方法和积分攻击方法的安全性。首先利用 MIBS 算法结构的特点和密钥扩展算法的特点，构造出了一些合适的 2^8 个 8 轮零相关线性逼近和揭示了一些轮子密钥之间的关系。结合部分和技术，我们对 13 轮的 MIBS-80 进行了多维零相关分析，结果显示比穷举具有 $80 - 74.5 = 5.5$ bit 的优势。另外，我们利用零相关线性逼近和积分区分器之间的关系，推导出一个积分区分器，并对 11 轮 MIBS-80 进行了积分攻击，将积分攻击的结果改进一轮。值得注意的是，本文选择的零相关线性逼近是最优的，这并不能保证转化过来的积分区分器最优，所以可能存在很好的积分区分器和更好的积分分析结果。另外，这两个方法对 MIBS-64 同样适用。单密钥下的主要分析结果，见表 1。尽管本文的两个结果都没有达到线性分析所能分析的 18 轮，但是它们从不同的角度反映密码设计的某些特点，也给出一个理解零相关分析和积分分析之间联系的例子。

表 1　单密钥 MIBS-80 的主要分析结果

分析方法	轮数	数据复杂度	时间复杂度	工作出处
差分分析	13	2^{62}CPs	2^{25}	文献[3]
线性分析	18	$2^{60.9}$KPs	$2^{76.1}$	文献[3]
积分分析	9	$2^{39.6}$CPs	$2^{68.4}$	文献[5]
积分分析	10	$2^{61.6}$CPs	2^{40}	文献[6]
积分分析	10	$2^{28.2}$CPs	$2^{53.2}$	文献[7]
积分分析	11	2^{60}CPs	$2^{59.8}$	本文
中间相遇分析	11	$2^{24.9}$CPs	$2^{66.2}$	文献[8]
不可能差分分析	12	2^{59}CPs	2^{63}	文献[4]
多维零相关分析	13	$2^{62.1}$KPs	$2^{74.9}$	本文

注：CPs 表示选择明文；KPs 表示已知明文。

参 考 文 献

[1] IZADI M, SADEGHIYAN B, SADEGHIANS, *et al*. MIBS: a new light-weight block cipher[C]. CANS 2009. Berlin: Springer, 2009: 334–348. doi: 10.1007/978-3-642-10433-6_22.

[2] 杨林, 王美琴. 简约轮的 MIBS 算法的差分分析[J]. 山东大学学报(理学版), 2010, 45(4): 12–15.
YANG L and WANG M. Differential cryptanalysis of reduced-round MIBS[J]. *Journal of Shandong University* (*Natural Science*), 2010, 45(4): 12–15.

[3] BAY A, NAKAJARA J, and VAUDENAY S. Cryptanalysis of reduced-round MIBS block cipher[C]. CANS 2010. Berlin: Springer, 2010: 1–19.

[4] 杜承航, 陈佳哲. 轻量级分组密码算法 MIBS 不可能差分分析[J]. 山东大学学报(理学版), 2012, 47(7): 55–58.
DU C and CHEN J. Impossible differential cryptanalysis of reduced round MIBS[J]. *Journal of Shandong University* (*Natural Science*), 2012, 47(7): 55–58.

[5] 王高丽, 王少辉. 对 MIBS 算法的 Integral 攻击[J]. 小型微型计算机系统, 2012, 33(4): 773–777. doi: 10.3969/j.issn.1000-1220.2012.04.020
WANG G and WANG S. Integral cryptanalysis of reduced round MIBS block ciphe[J]. *Journal of Chinese Computer Systems*, 2012, 33(4): 773–777. doi: 10.3969/j.issn.1000-1220.2012.04.020.

[6] 于晓丽, 吴文玲, 李艳俊. 低轮 MIBS 分组密码的积分分析[J]. 计算机研究与发展, 2013, 50(10): 2117–2125.
YU X, WU W, and LI Y. Integral attack of reduced-round MIBS block ciper[J]. *Journal of Computer Research and Development*, 2013, 50(10): 2117–2125.

[7] 潘志舒, 郭建胜, 曹进克, 等. MIBS 算法的积分攻击[J]. 通信学报, 2014, 35(7): 157–163.
PAN Z, GUO J, CAO J, *et al*. Integral attack on MIBS block

cipher[J]. *Journal on Communications,* 2014, 35(7): 157–163.

[8] 刘超, 廖福成, 卫宏儒. 对 MIBS 算法的中间相遇攻击[J]. 内蒙古大学学报(自然科学版), 2013, 44(3): 308–315.
LIU C, LIAO F, and WEI H. Meet-in-the-middle attacks on MIBS[J]. *Journal of Inner Mongolia University* (*Natural Science Edition*), 2013, 44(3): 308–315.

[9] 陈平, 廖福成, 卫宏儒. 对轻量级 MIBS 算法的相关密钥不可能差分攻击[J]. 通信学报, 2014, 35(2): 190–193.
CHEN P, LIAO F, and WEI H. Related-key impossible differential attack on a lightweight block cipher MIBS[J]. *Journal on Communications*, 2014, 35(2): 190–193.

[10] BOGDANOV A and RIJMEN V. Linear hulls with correlation zero and linear cryptanalysis of block ciphers[J]. *Designs*, *Codes and Cryptography*, 2014, 70(3): 369–383. doi: 10.1007/s10623-012-9697-z.

[11] BOGDANOV A and WANG M. Zero correlation linear cryptanalysis with reduced data complexity[C]. FSE 2012, Washington, DC, USA, 2012: 29–48. doi: 10.1007/978-3-642-34047-5_3.

[12] BOGDANOV A, LEANDER G, NYBERG K, *et al*. Integral and multidimensional linear distinguishers with correlation zero[C]. ASIACRYPT 2012, Beijing, China, 2012: 244–261. doi: 10.1007/978-3-642-34961-4_16.

[13] SOLEIMANY H and NYBERG K. Zero-correlation linear cryptanalysis of reduced-round LBlock[J]. *Designs, Codes and Cryptography*, 2014, 73(2): 683–698. doi: 10.1007/s10623-014-9976-y.

[14] WEN L, WANG M, and BOGDANOV A. Multidimensional zero-correlation linear cryptanalysis of E2[C]. AFRICACRYPT 2014, Marrakesh, Morocco, 2014: 147–164. doi: 10.1007/978-3-319-06734-6_10.

[15] BOGDANOV A, GENG H, WANG M,*et al*. Zero-correlation linear cryptanalysis with FFT and improved attacks on ISO standards Camellia and CLEFIA[C]. SAC 2013, Burnaby, BC, Canada, 2013: 306–323. doi: 10.1007/ 978-3-662-43414-7_16.

[16] BOGDANOV A, KNUDSEN L, LEANDER G, *et al*. PRESENT: an ultra-lightweight block cipher[C]. CHESS 2007, Vol. 4727: 450–466. doi: 10.1007/978-3-540-74735-2_31.

伊文坛： 男，1989 年生，博士生，研究方向为分组密码安全性分析.

鲁林真： 男，1985 年生，博士生，研究方向为分组密码安全性分析.

陈少真： 女，1967 年生，教授，研究方向为密码学与信息安全.