分组密码最小活跃 S 盒个数快速搜索算法

刘正斌[1]，李永强[2]，朱朝熹[1]
（1. 保密通信重点实验室，四川 成都 610041；2. 中国科学院信息工程研究所，北京 100093）

摘 要：为了解决密码设计中最小活跃 S 盒个数的快速计算问题，研究了扩散层的差分和掩码传播性质，提出了一种计算最大距离可分（MDS）矩阵和二元域矩阵的差分/掩码模式分布表的方法，并证明了所提方法计算复杂度的下界。基于扩散矩阵的差分/掩码模式分布表，提出了一种快速搜索分组密码最小活跃 S 盒个数的算法，将其用于代入置换网络（SPN）型分组密码，找到了 LED、SKINNY、CRAFT 和 FIDES 的全轮最小活跃 S 盒个数。

关键词：分组密码；差分密码分析；线性密码分析；活跃 S 盒；自动化搜索
中图分类号：TN918
文献标志码：A
DOI: 10.11959/j.issn.1000–436x.2023022

Fast algorithm to search for the minimum number of active S-boxes of block cipher

LIU Zhengbin[1], LI Yongqiang[2], ZHU Chaoxi[1]
1. Science and Technology on Communication Security Laboratory, Chengdu 610041, China
2. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Abstract: To solve the problem of fast calculation of the minimum number of active S-boxes in cryptographic design, the difference and mask propagation of the diffusion layer were investigated, and a method was proposed to compute the difference (resp. mask) pattern distribution table of MDS (maximum distance separable) matrices and binary matrices. A lower bound on the computation complexity of the proposed method was also given. Based on the difference (resp. mask) pattern distribution table of diffusion matrix, a fast algorithm to search for the minimum number of active S-boxes of block cipher was proposed. The proposed algorithm is applied to some SPN (substitution permutation network) block ciphers, and finds the minimum number of active S-boxes for the full round of LED, SKINNY, CRAFT and FIDES.

Keywords: block cipher, differential cryptanalysis, linear cryptanalysis, active S-box, automatic search

0 引言

差分密码分析[1]和线性密码分析[2]是密码分析中最有效的 2 种分析方法，抵抗差分密码分析和线性密码分析是现代分组密码最基本的一项设计准则。评估分组密码抵抗差分密码分析和线性密码分析的主要方法包括计算最大差分/线性特征的概率和计算最小活跃 S 盒个数。对于代入置换网络（SPN, substitution permutation network）型分组密码，根据宽轨迹设计策略[3]，可以得到抵抗差分密码分析和线性密码分析的可证明安全界。通过计算最小活跃 S 盒个数，并结合 S 盒的最大差分概率和线性概率，设计者可以计算出最大差分特征概率和线性特征概率的上界。目前，学术界已经提出了一些自动化搜索算法来辅助评估分组密码的安全性。

1994 年，Matsui[4]提出一种分支定界搜索算法来自动化搜索分组密码 DES（data encryption standard）的最优差分特征和线性特征，该算法也称 Matsui 算法。尽管该算法能够在有限时间内找

收稿日期：2022–08–05；**修回日期**：2022–10–30
基金项目：国家自然科学基金资助项目（No.61772517）
Foundation Item: The National Natural Science Foundation of China (No.61772517)

到 DES 的 16 轮最优差分特征和线性特征，但是对于其他一些分组密码，它的效率却非常低。随后，Ohta 等[5]和 Aoki 等[6]分别改进了 Matsui 算法，找到了分组密码 FEAL（fast data encipherment algorithm）的最优差分特征和线性特征。虽然 Matsui 算法可以扩展到搜索 SPN 型分组密码的最优差分特征和线性特征，但是其线性层的快速扩散性质和雪崩效应严重降低了搜索效率。对于使用比特置换层的 SPN 型分组密码，Arnaud 等[7]优化了 Matsui 算法并找到了分组密码 PRESENT、PUFFIN 和 ICEBERG 的全轮最优差分特征。Ji 等[8]则通过引入 3 种加速方法来提高 Matsui 算法的搜索效率，并找到了 DESL 和 GIFT 约减轮数的最优差分特征和线性特征。Kim 等[9]改进了 Matsui 算法来搜索 SPN 型分组密码的最优差分特征和线性特征。

2011 年，Mouha 等[10]提出了一种基于混合整数线性规划（MILP, mixed integer linear programming）的方法来搜索 SPN 型分组密码的最小活跃 S 盒个数。Sun 等[11]将基于 MILP 的方法扩展到使用比特置换层的分组密码，提出了 2 种能够精确刻画 S 盒的差分和掩码传播的方法，即逻辑条件模型和凸包计算，并给出了约减不等式组的贪婪算法。Abdelkhalek 等[12]将逻辑条件模型中约束条件的生成问题转化成布尔函数的和积的最小化问题，并使用 Quine-McCluskey 算法[13-14]和 Espresso 算法[15]来建立 8 bit S 盒的差分分布表的比特模型。为了提高基于 MILP 方法的效率，Zhang 等[16]在 MILP 模型中引入了分支定界策略来减少约束条件，降低搜索空间；Zhou 等[17]则使用分而治之的方法来优化模型求解。近几年，基于 MILP 的方法被广泛应用于分组密码的设计与分析中[18-30]。

除了算法模型的优化外，基于 MILP 方法的效率主要由求解器的效率决定，如 CPLEX、Gurobi 等。对于轮数较长或者轮函数较复杂的分组密码，由于需要更多的变量和不等式来描述差分和掩码的传播，因此求解器通常需要执行非常长的时间来得到最优解。对于攻击一个已知的分组密码，花费十几天甚至几个月的时间是有意义的。然而，从分组密码设计的角度，为了优化密码部件（S 盒、扩散矩阵等）的选择以及确定合理的迭代轮数，设计者通常需要进行多次安全性评估。因此，一个快速搜索最小活跃 S 盒个数的算法对于评估分组密码抵抗差分密码分析和线性密码分析的安全性具有重要意义，在分组密码设计过程中具有重要实用价值。

本文主要的研究工作如下。

1) 提出了扩散矩阵的差分/掩码模式分布表的概念。对于二元域矩阵，证明了通过遍历输入差分/掩码方式来构造差分/掩码模式分布表的计算复杂度的下界。基于该下界，给出了构造差分/掩码模式分布表的快速方法。

2) 对于任意阶最大距离可分（MDS, maximum distance separable）矩阵，证明了满足分支数条件的差分/掩码模式一定能够实例化，即对于特定的差分/掩码模式，一定存在对应的差分/掩码传播。基于分支数关系，给出了任意阶 MDS 矩阵的差分/掩码模式的传播集合。

3) 基于差分/掩码模式分布表，提出了一种自动化搜索分组密码最小活跃 S 盒个数的快速算法。针对 SPN 型分组密码，通过实验验证了该算法的有效性。实验结果表明，本文算法效率比目前常用的基于 MILP 的方法高。

1　差分/掩码模式分布表

1.1　差分/掩码模式分布表定义

定义 1　S 盒的差分分布表[1]。设 $S:\mathbb{F}_2^m \to \mathbb{F}_2^m$ 是一个 m bit S 盒，它的差分分布表（DDT, difference distribution table）是一个 $2^m \times 2^m$ 的表，其中，每一行对应 S 盒的输入差分，每一列对应 S 盒的输出差分。给定输入差分 $\alpha \in \mathbb{F}_2^m$、输出差分 $\beta \in \mathbb{F}_2^m$，差分分布表中第 α 行、第 β 列的元素为

$$\mathrm{DDT}_S(\alpha,\beta)=\#\{x \in \mathbb{F}_2^m \mid S(x) \oplus S(x \oplus \alpha)=\beta\}$$

定义 2　S 盒的线性近似表[2]。设 $S:\mathbb{F}_2^m \to \mathbb{F}_2^m$ 是一个 m bit S 盒，它的线性近似表（LAT, linear approximation table）是一个 $2^m \times 2^m$ 的表，其中，每一行对应 S 盒的输入掩码，每一列对应 S 盒的输出掩码。给定输入掩码 $\mu \in \mathbb{F}_2^m$、输出掩码 $\nu \in \mathbb{F}_2^m$，线性近似表中第 μ 行、第 ν 列的元素为

$$\mathrm{LAT}_S(\mu,\nu)=\#\{x \in \mathbb{F}_2^m \mid \mu x=\nu S(x)\}-2^{m-1}$$

定义 3　差分模式和掩码模式。设 $x \in \mathbb{F}_2^m$、$X \in \mathbb{F}_2$，定义

$$X=\delta(x)=\begin{cases}0, & x=0\\ 1, & \text{其他}\end{cases}$$

令 $\Delta \boldsymbol{x}, \Gamma \boldsymbol{x} \in \mathbb{F}_2^m$ 分别表示 x 的差分和掩码，则

$\Delta \boldsymbol{X}=\delta(\Delta \boldsymbol{x})$ 定义为 x 的差分模式，$\Gamma \boldsymbol{X}=\delta(\Gamma \boldsymbol{x})$ 定义为 x 的掩码模式。

对于差分向量 $\Delta \boldsymbol{x}=(\Delta x_1,\cdots,\Delta x_n)\in(\mathbb{F}_2^m)^n$，对应的差分模式定义为

$$\Delta \boldsymbol{X}=\delta(\Delta \boldsymbol{x})=(\Delta X_1,\cdots,\Delta X_n)\in \mathbb{F}_2^n$$

其中

$$\Delta X_i=\delta(\Delta x_i)=\begin{cases}0, & \Delta x_i=0\\ 1, & \text{其他}\end{cases}$$

类似地，对于掩码向量 $\Gamma \boldsymbol{x}=(\Gamma x_1,\cdots,\Gamma x_n)\in(\mathbb{F}_2^m)^n$，对应的掩码模式定义为

$$\Gamma \boldsymbol{X}=\delta(\Gamma \boldsymbol{x})=(\Gamma X_1,\cdots,\Gamma X_n)\in \mathbb{F}_2^n$$

其中

$$\Gamma X_i=\delta(\Gamma x_i)=\begin{cases}0, & \Gamma x_i=0\\ 1, & \text{其他}\end{cases}$$

对于双射 S 盒，设 $\Delta \boldsymbol{X}$ 和 $\Delta \boldsymbol{Y}$ 分别表示 S 盒的输入差分模式和输出差分模式，则 $\Delta \boldsymbol{Y}=0$ 当且仅当 $\Delta \boldsymbol{X}=0$，$\Delta \boldsymbol{Y}=1$ 当且仅当 $\Delta \boldsymbol{X}=1$，反之亦然。

对于线性变换 $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$，$\boldsymbol{A}$ 是一个 n 阶矩阵，设输入差分为 $\Delta \boldsymbol{x}=(\Delta x_1,\cdots,\Delta x_n)$，则输出差分为 $\Delta \boldsymbol{y}=\boldsymbol{A}\Delta \boldsymbol{x}$。令 $\Delta \boldsymbol{X}=(\Delta X_1,\cdots,\Delta X_n)$ 表示输入差分模式，$\Delta \boldsymbol{Y}=(\Delta Y_1,\cdots,\Delta Y_n)$ 表示输出差分模式，那么 $(\Delta \boldsymbol{X},\Delta \boldsymbol{Y})$ 就称为线性变换的一个差分模式传播。

定义 4 差分模式分布表。线性变换的差分模式分布表（DPDT, difference pattern distribution table）是一个表示其输入差分模式和输出差分模式的分布表。对于给定的输入差分模式 $\Delta \boldsymbol{X}=(\Delta X_1,\cdots,\Delta X_n)\in \mathbb{F}_2^n$ 和输出差分模式 $\Delta \boldsymbol{Y}=(\Delta Y_1,\cdots,\Delta Y_n)\in \mathbb{F}_2^n$，令 $\alpha=\Delta X_1\|\cdots\|\Delta X_n$，$\beta=\Delta Y_1\|\cdots\|\Delta Y_n$，那么差分模式分布表中第 α 行、第 β 列元素 $\mathrm{DPDT}(\alpha,\beta)=1$ 表示一定存在与 $(\Delta \boldsymbol{X},\Delta \boldsymbol{Y})$ 对应的差分传播 $(\Delta \boldsymbol{x},\Delta \boldsymbol{y})$，满足 $\Pr(\Delta \boldsymbol{x}\to\Delta \boldsymbol{y})\neq 0$，其中，$\Pr(\cdot)$表示概率。

与差分模式分布表对应，可以定义线性变换 $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$ 的掩码模式分布表。记输入掩码为 $\Gamma \boldsymbol{x}=(\Gamma x_1,\cdots,\Gamma x_n)$，那么输出掩码为 $\Gamma \boldsymbol{y}=(\boldsymbol{A}^{\mathrm{T}})^{-1}\Gamma \boldsymbol{x}$。令 $\Gamma \boldsymbol{X}=(\Gamma X_1,\cdots,\Gamma X_n)$ 表示输入掩码模式，$\Gamma \boldsymbol{Y}=(\Gamma Y_1,\cdots,\Gamma Y_n)$ 表示输出掩码模式，那么 $(\Gamma \boldsymbol{X},\Gamma \boldsymbol{Y})$ 就称为线性变换的一个掩码模式传播。

定义 5 掩码模式分布表。线性变换的掩码模式分布表（MPDT, mask pattern distribution table）是表示其输入掩码模式和输出掩码模式的分布表。对于给定的输入掩码模式 $\Gamma \boldsymbol{X}=(\Gamma X_1,\cdots,\Gamma X_n)\in \mathbb{F}_2^n$ 和输出差分模式 $\Gamma \boldsymbol{Y}=(\Gamma Y_1,\cdots,\Gamma Y_n)\in \mathbb{F}_2^n$，令 $\mu=\Gamma X_1\|\cdots\|\Gamma X_n$，$\nu=\Gamma Y_1\|\cdots\|\Gamma Y_n$，那么掩码模式分布表中第 μ 行、第 ν 列元素 $\mathrm{MPDT}(\mu,\nu)=1$ 表示一定存在与 $(\Gamma \boldsymbol{X},\Gamma \boldsymbol{Y})$ 对应的掩码 $(\Gamma \boldsymbol{x},\Gamma \boldsymbol{y})$，满足 $\Pr(\Gamma \boldsymbol{x}\cdot \boldsymbol{x}=\Gamma \boldsymbol{y}\cdot \boldsymbol{y})\neq \frac{1}{2}$。

接下来，针对分组密码中最常用的 2 种扩散矩阵——MDS 矩阵和二元域矩阵，给出其差分模式分布表和掩码模式分布表的构造。

1.2 MDS 矩阵差分/掩码模式分布表构造

首先证明 MDS 矩阵的差分/掩码模式传播与其分支数之间的等价关系，然后基于分支数来构造 MDS 矩阵的差分/掩码模式分布表。

设 m 为正整数，$\mathbb{F}_{2^m}$ 为包含 2^m 个元素的有限域。$\mathbb{GL}_n(\mathbb{F}_{2^m})$ 表示元素取自有限域 $\mathbb{F}_{2^m}$ 上的 n 阶可逆矩阵的集合。对于 $\mathbb{F}_q$ 上的 MDS 码，其码字分布有如下结果。

定理 1[31] 设 $\mathcal{C}$ 为 $\mathbb{F}_q$ 上参数为 $[n,k,d=n-k+1]$ 的 MDS 码，则 $\mathcal{C}$ 中重量为 w 的码字个数为 $A_w=\binom{n}{w}(q-1)\sum_{j=0}^{w-d}(-1)^j\binom{w-1}{j}q^{w-d-j}$。

定理 2 设 $\boldsymbol{A}\in\mathbb{GL}_n(\mathbb{F}_{2^m})$ 为 MDS 矩阵，且满足 $2^m>2n$，$\boldsymbol{u}=(u_1,\cdots,u_n)\in\mathbb{F}_2^n$ 为非零向量，那么对于任意非零向量 $\boldsymbol{v}=(v_1,\cdots,v_n)\in\mathbb{F}_2^n$，且 $n+1-\mathrm{Hw}(\boldsymbol{u})\leqslant \mathrm{Hw}(\boldsymbol{v})\leqslant n$，一定存在向量 $(x_1,\cdots,x_n)\in\mathbb{F}_{2^m}^n$，使 $(\delta(x_1),\cdots,\delta(x_n))=\boldsymbol{u}$，$(\delta(y_1),\cdots,\ \delta(y_n))=\boldsymbol{v}$，其中，$(y_1,\cdots,y_n)^{\mathrm{T}}=\boldsymbol{A}(x_1,\cdots,x_n)^{\mathrm{T}}$，$\mathrm{HW}(\cdot)$表示汉明重量。

证明 设 $\boldsymbol{A}\in\mathbb{GL}_n(\mathbb{F}_{2^m})$ 为 MDS 矩阵，则 $[\boldsymbol{I},\boldsymbol{A}]$ 为 $\mathbb{F}_{2^m}$ 上参数为 $[2n,n,n+1]$ 的 MDS 码 $\mathcal{C}$ 的生成矩阵。对于任意 $(x_1,\cdots,x_n)\in\mathbb{F}_{2^m}^n$，令 $(y_1,\cdots,y_n)^{\mathrm{T}}=\boldsymbol{A}\cdot(x_1,\cdots,x_n)^{\mathrm{T}}$，那么 $(x_1,\cdots,x_n,y_1,\cdots,y_n)$ 为 $\mathcal{C}$ 中码字。因此，只需证明对于任意 $n+1\leqslant w\leqslant 2n$，$\mathcal{C}$ 中均存在重量为 w 的码字。

令 $q=2^m$，$d=n+1$，根据上述结果可知 $A_w=\binom{2n}{w}(q-1)\sum_{j=0}^{w-d}(-1)^j\binom{w-1}{j}q^{w-d-j}$。

如果存在 i，使 $w-d=2i+1$，则

$$A_w=\binom{2n}{w}(q-1)\sum_{j=0}^{2i+1}(-1)^j\binom{w-1}{j}q^{w-d-j}=$$

$$\binom{2n}{w}(q-1)\sum_{j=0}^{i}\left((-1)^{2j}\binom{w-1}{2j}q^{w-d-2j}+(-1)^{2j+1}\binom{w-1}{2j+1}q^{w-d-2j-1}\right)=\binom{2n}{w}$$

$$(q-1)\sum_{j=0}^{i}q^{w-d-2j-1}\left(\binom{w-1}{2j}q-\frac{w-1-2j}{2j+1}\binom{w-1}{2j}\right)=$$

$$\binom{2n}{w}(q-1)\sum_{j=0}^{i}q^{w-d-2j-1}\binom{w-1}{2j}\left(q-\frac{w}{2j+1}+1\right)\geqslant$$

$$\binom{2n}{w}(q-1)\sum_{j=0}^{i}q^{w-d-2j-1}(q+1-2n)>0$$

如果存在 i，使 $w-d=2i$，此时根据上述证明情况可得

$$A_w=\binom{2n}{w}(q-1)\sum_{j=0}^{2i}(-1)^j\binom{w-1}{j}q^{w-d-j}=$$

$$\binom{2n}{w}(q-1)\sum_{j=0}^{2(i-1)+1}(-1)^j\binom{w-1}{j}q^{w-d-j}+$$

$$\binom{n}{w}(q-1)(-1)^{2i}\binom{w-1}{2i}q^{w-d-2i}>0$$

综上，定理 2 证毕。

根据定理 2，MDS 矩阵的差分/掩码模式分布表可以直接根据其分支数来构造。该方法只需要遍历差分/掩码模式，极大降低了差分/掩码模式分布表的计算复杂度，对于 $\mathbb{F}_{2^m}$ 上 n 阶 MDS 矩阵，计算复杂度从 $\mathcal{O}(2^{mn})$ 降为 $\mathcal{O}(2^n)$。

1.3　二元域矩阵差分/掩码模式分布表构造

对于二元域矩阵，首先证明通过遍历输入差分/掩码来构造差分/掩码模式分布表的计算复杂度的下界，然后给出差分/掩码模式分布表的构造方法。

对任意 $\boldsymbol{A}\in\mathbb{GL}_n(\mathbb{F}_2)$ 以及 $(x_1,\cdots,x_n)\in\mathbb{F}_{2^k}^n$，令 $(y_1,\cdots,y_n)^{\mathrm{T}}=\boldsymbol{A}(x_1,\cdots,x_n)^{\mathrm{T}}$，$\Theta_k(\boldsymbol{A})$ 为如下集合 $\{(\delta(x_1),\cdots,\delta(x_n),\delta(y_1),\cdots,\delta(y_n))\,|\,(x_1,\cdots,x_n)\in\mathbb{F}_{2^k}^n\}$。

引理 1　设 R 为有限集合，T 为 R 的子集，$\overline{T}$ 为 T 的补集，即 $\overline{T}=\frac{R}{T}$。记 $R_1,\cdots,R_{m-1}$ 以及 R_m 为集合 R 的不同子集，那么 $\#\bigcap_{i=1}^{m}\overline{R}_i=\#R+\sum_{t=1}^{m}(-1)^t\cdot\sum_{1\leqslant i_1<\cdots<i_t\leqslant m}\#\bigcap_{j=1}^{t}R_{i_j}$。

定理 3　设 $\boldsymbol{A}\in\mathbb{GL}_n(\mathbb{F}_2)$，$d$ 为使 $2^d>2n-1$ 成立的最小正整数，那么对于任意 $m\geqslant d$，$\Theta_m(\boldsymbol{A})=\Theta_d(\boldsymbol{A})$。

证明　由于 $\boldsymbol{A}\in\mathbb{GL}_n(\mathbb{F}_2)$，定理 3 只涉及有限域的加法运算，从而可以将 $\mathbb{F}_{2^d}$ 视为 $\mathbb{F}_{2^m}$ 的加法子环。因此，$\Theta_d(\boldsymbol{A})\subseteq\Theta_m(\boldsymbol{A})$。

下面只需证明 $\Theta_m(\boldsymbol{A})\subseteq\Theta_d(\boldsymbol{A})$，即对任意 $\boldsymbol{r}\in\Theta_m(\boldsymbol{A})$，都有 $\boldsymbol{r}\in\Theta_d(\boldsymbol{A})$。

设非零向量 $\boldsymbol{r}=(i_1,\cdots,i_n,j_1,\cdots,j_n)\in\Theta_m(\boldsymbol{A})$，$y_t=l_t(x_1,\cdots,x_n)(1\leqslant t\leqslant n)$ 是 $x_1,\cdots,x_n$ 的线性函数，则下述关系式在 $\mathbb{F}_{2^m}$ 上有解。

$$\begin{cases}\delta(x_t)=i_t,1\leqslant t\leqslant n\\ \delta(y_t)=\delta(l_t(x_1,\cdots,x_n))=j_t,1\leqslant t\leqslant n\end{cases}$$

不失一般性，令 l_n 不是常值函数，则有下述 2 种情况。

情形 1　$j_n=1$，即 $\delta(y_n)=1$，考虑式(1)所示方程组。

$$\begin{cases}x_1=\mathrm{or}\neq 0\\ \vdots\\ x_n=\mathrm{or}\neq 0\\ y_1=l_1(x_1,\cdots,x_n)=\mathrm{or}\neq 0\\ \vdots\\ y_{n-1}=l_{n-1}(x_1,\cdots,x_n)=\mathrm{or}\neq 0\\ y_n=l_n(x_1,\cdots,x_n)\neq 0\end{cases}\tag{1}$$

其中，前 $2n-1$ 个关系式取等号还是不等号由 $\boldsymbol{r}$ 中相应的 $i_1,\cdots,i_n,j_1,\cdots,j_{n-1}$ 的值所确定。只需证明 $\delta(y_n)$ 在 $\mathbb{F}_{2^d}$ 上也能取到 1，即式(1)在 $\mathbb{F}_{2^d}^n$ 中有解。

令 s 为式(1)的前 $2n-1$ 个关系式中线性无关的等式个数，即前 $2n-1$ 个关系式中等于 0 的线性无关的等式个数，那么 $0\leqslant s\leqslant n-1$（$s\geqslant n$ 可得 $x_1=0,\cdots,x_n=0$，与 $\boldsymbol{r}$ 为非零向量矛盾）。将 s 个线性无关的等式化简之后得到 $n-s$ 个变量，以及最多只剩下 $2n-1-s$ 个不等式的方程组。注意到，在消去变量后，$l_n(x_1,\cdots,x_n)$ 不可能是 0 函数。因为 $\delta(y_n)$ 在 $\mathbb{F}_{2^m}$ 上可以等于 1，并且由于 $\boldsymbol{A}\in\mathbb{GL}_n(\mathbb{F}_2)$，因此消去过程与域的选取无关。

记 $q=2^d$，注意到每个不等式在 $\mathbb{F}_{2^d}^{n-s}$ 上有且仅有 $q^{n-s}-q^{n-s-1}$ 个点。于是，这 $2n-s-1$ 个不等式在 $\mathbb{F}_{2^d}^{n-s}$ 上至多有 $q^{n-s}-(2n-s-1)q^{n-s-1}+1$ 个点。

但是，等式 $l_n(x_1,\cdots,x_n)=0$ 在 $\mathbb{F}_{2^d}$ 上的零点个数只有 q^{n-s-1} 个。由于 $q=2^d>2n-1$ 和 $q^{n-s}-(2n-s-1)\cdot q^{n-s-1}+1=q^{n-s-1}(q-(2n-1)+s)>q^{n-s-1}$ 对 $0\leqslant s\leqslant n-1$ 都成立。因此，在 $\mathbb{F}_{2^d}^n$ 上满足式(1)的前 $2n-1$ 个关系式的所有点中，至少有一个点使

$l_n(x_1,\cdots,x_n)\neq 0$，即 $\delta(y_n)$ 在 $\mathbb{F}_{2^d}$ 上可以取到1。

情形 2 $j_n=0$，即 $\delta(y_n)=0$。下面证明 $\delta(y_n)$ 在 $\mathbb{F}_{2^d}$ 上也可以取到 0。考虑式(2)所示方程组。

$$\begin{cases} x_1=\text{or}\neq 0 \\ \vdots \\ x_n=\text{or}\neq 0 \\ y_1=l_1(x_1,\cdots,x_n)=\text{or}\neq 0 \\ \vdots \\ y_{n-1}=l_{n-1}(x_1,\cdots,x_n)=\text{or}\neq 0 \\ y_n=l_n(x_1,\cdots,x_n)=0 \end{cases} \tag{2}$$

只需证明式(2)在 $\mathbb{F}_{2^d}^n$ 中仍然有解。将 $l_n(x_1,\cdots,x_n)=0$ 代入前 $2n-1$ 个关系式，消去一个变量。类似情形 1，假设前 $2n-1$ 个关系式有 s $(s\leqslant n-2)$ 个线性无关的等式，由引理 1 可知至少有 $q^{n-s-1}+\sum_{i=1}^{n-s-1}(-1)^i\binom{2n-s-1}{i}q^{n-1-s-i}>0$ 个点，使替换后的式(2)成立（大于 0 的证明与定理 2 中 MDS 矩阵的证明一致）。于是，$\delta(y_n)$ 在 $\mathbb{F}_{2^d}$ 上也能为 0。

综上，定理 3 证毕。

根据定理 3，对于分组密码中常用的任意 4 阶和 8 阶二元域矩阵，构造其差分/掩码模式分布表的计算复杂度分别为 $\mathcal{O}(2^{12})$ 和 $\mathcal{O}(2^{32})$，因此可以在计算机上非常快速地构造。然而，构造 16 阶二元域矩阵的差分/掩码模式分布表需要 $\mathcal{O}(2^{80})$ 的计算量，因此无法通过这种方式直接构造。二元域矩阵差分模式分布表的构造如算法 1 所示。在算法 1 中，对于 4 阶和 8 阶二元域矩阵，d 的值分别为 3 和 4。

算法 1 二元域矩阵差分模式分布表构造算法

for $i=0$ to 2^n-1 do
  $\text{DPDT}[i]=\phi$
end for
for $\Delta x_1,\cdots,\Delta x_n=0$ to 2^d-1 do
  $(\Delta y_1,\cdots,\Delta y_n)^{\mathrm{T}}=\boldsymbol{A}(\Delta x_1,\cdots,\Delta x_n)^{\mathrm{T}}$
  for $i=1$ to n do
    $\Delta X_i=\delta(\Delta x_i)$
    $\Delta Y_i=\delta(\Delta y_i)$
  end for
  $\Delta\boldsymbol{X}=\Delta X_1\|\cdots\|\Delta X_n$
  $\Delta\boldsymbol{Y}=\Delta Y_1\|\cdots\|\Delta Y_n$
  $\text{DPDT}[\Delta\boldsymbol{X}]=\text{DPDT}[\Delta\boldsymbol{X}]\cup\{\Delta\boldsymbol{Y}\}$
end for

对于差分模式分布表的每一行，按照汉明重量从小到大的顺序对输出差分模式进行排序。掩码模式分布表的构造是类似的，只需要将其中的矩阵 $\boldsymbol{A}$ 替换为矩阵 $(\boldsymbol{A}^{\mathrm{T}})^{-1}$。

2 SPN 型分组密码活跃 S 盒个数搜索算法

2.1 分支定界搜索算法

Matsui 算法对差分特征和线性特征执行递归搜索，它根据已知的 i 轮最优特征概率 B_i（$1\leqslant i\leqslant r-1$）和 r 轮最优特征概率 B_r 的初始估计值 $\overline{B_r}$ 来计算 B_r。对于任意 $\overline{B_r}$，只要 $\overline{B_r}\leqslant B_r$，Matsui 算法一定可以得到 r 轮最优特征概率 B_r。对于 Feistel 密码，其轮函数的差分传播过程如图 1 所示，其中，F 表示轮函数中的一个变换。Matsui 算法搜索最优差分特征的伪代码如算法 2 所示。

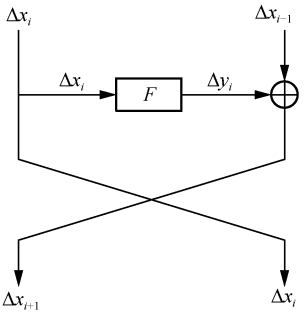


图 1 Feistel 密码轮函数的差分传播过程

算法 2 Matsui 算法搜索最优差分特征

Procedure Round-1
for each candidate for Δx_1 do
  $p_1=\max_{\Delta y}\Pr(\Delta x_1\to\Delta y)$
  if $p_1B_{r-1}\geqslant\overline{B}_r$ do
  call Procedure Round-2
  end if
end for
return to the upper procedure
Procedure Round-2
for each candidate for Δx_2 and Δy_2 do
  $p_2=\Pr(\Delta x_2\to\Delta y_2)$
  if $p_1p_2B_{r-2}\geqslant\overline{B}_r$ do
    call Procedure Round-3
  end if
end for
return to the upper procedure
Procedure Round-*i* $(3\leqslant i\leqslant r-1)$

for each candidate for Δy_i do

$\Delta x_i = \Delta x_{i-2} \oplus \Delta y_{i-1}$

$p_i = \Pr(\Delta x_i \to \Delta y_i)$

if $p_1 p_2 \cdots p_i B_{r-i} \geqslant \overline{B}_r$ do

call Procedure Round-(i+1)

end if

end for

return to the upper procedure

Procedure Round-*r*

$\Delta x_r = \Delta x_{r-2} \oplus \Delta y_{r-1}$

$p_r = \max_{\Delta y} \Pr(\Delta x_r \to \Delta y)$

if $p_1 p_2 \cdots p_r \geqslant \overline{B}_r$ do

$\overline{B}_r = p_1 p_2 \cdots p_r$

end if

return to the upper procedure

2.2　SPN 型分组密码最小活跃 S 盒个数搜索算法

SPN 结构是现代分组密码最常用的一种密码结构，其轮函数包含一个非线性替换层和一个线性扩散层。非线性替换层又称 S 盒层，它使用一组并行的 S 盒；线性扩散层是一个线性函数，通常包含一个基于字的置换和一个基于矩阵乘法的线性变换。SPN 型分组密码最著名的实例是高级加密标准（AES, advanced encryption standard），它的轮函数包含字节置换（SubBytes）、行移位（ShiftRows）、列混淆（MixColumns）、轮密钥加（AddRoundKey）。

在轻量级密码领域，研究者提出了若干 SPN 型轻量级分组密码。为了适用于资源受限环境，S 盒层通常使用 4 bit S 盒，列混淆层则使用轻量级 MDS 矩阵或者二元域矩阵。不失一般性，本文仍然沿用 AES 的轮函数结构，使用符号 SB 表示 S 盒层，SR 表示线性置换层，MC 表示列混淆层，AK 表示轮密钥加。SPN 型分组密码的轮函数如图 2 所示，其中，S 表示 S 盒变换。

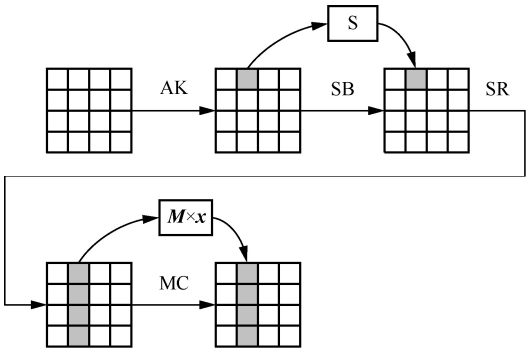


图 2　SPN 型分组密码的轮函数

设 $\Delta \boldsymbol{X}^i = (\Delta X_1, \cdots, \Delta X_w) \in \mathbb{F}_2^w$ 为第 i 轮输入差分模式，那么 $\Delta \boldsymbol{Y}^i = \mathrm{SR}(\Delta \boldsymbol{X}^i)$ 为列混淆层的输入差分模式。假设列混淆层并行使用 m 个 n 阶矩阵，则 $w = mn$。对于 $\Delta \boldsymbol{Y}^i = (\Delta Y_1, \cdots, \Delta Y_w)$，定义 $\Delta Y_j^i = \Delta Y_{nj-n+1} \| \cdots \| \Delta Y_{nj}$ $(1 \leqslant j \leqslant m)$，则可以将 $\Delta \boldsymbol{Y}^i$ 表示为 $\Delta \boldsymbol{Y}^{\boldsymbol{i}} = (\Delta Y_1^i, \cdots, \Delta Y_m^i)$。列混淆层的输出差分模式，即第 $i+1$ 轮的输入差分模式为

$$\Delta \boldsymbol{X}^{i+1} = (\Delta X_1, \cdots, \Delta X_w) = \left(\Delta X_1^{i+1}, \cdots, \Delta X_m^{i+1}\right) = (\mathrm{DPDT}(\Delta Y_1^i), \cdots, \mathrm{DPDT}(\Delta Y_m^i))$$

类似地，对于列混淆层的输入掩码模式 $\Gamma \boldsymbol{Y}^i = (\Gamma Y_1^i, \cdots, \Gamma Y_m^i) \in (\mathbb{F}_2^n)^m$，其输出掩码模式为

$$\Gamma \boldsymbol{X}^{i+1} = (\Gamma X_1, \cdots, \Gamma X_w) = \left(\Gamma X_1^{i+1}, \cdots, \Gamma X_m^{i+1}\right) = (\mathrm{MPDT}(\Gamma Y_1^i), \cdots, \mathrm{MPDT}(\Gamma Y_m^i))$$

由于差分活跃 S 盒和线性活跃 S 盒的搜索算法本质上是相同的，接下来仅讨论差分活跃 S 盒的搜索算法。最小差分活跃 S 盒个数的搜索算法如算法 3 所示，这里使用 B_i $(1 \leqslant i \leqslant r)$ 表示 i 轮最小活跃 S 盒个数，$\overline{B_r}$ 表示 B_r 的初始值。由于 Matsui 算法（算法 2）使用 B_i $(1 \leqslant i \leqslant r-1)$ 和 $\overline{B_r}$ 来计算 B_r 的值，$\overline{B_r}$ 的取值对于算法效率有很大影响，$\overline{B_r}$ 越大，效率越高，因此提高 Matsui 算法效率的关键是尽可能地提前停止。

因为每一轮至少有一个活跃 S 盒，所以算法 3 在搜索起始阶段将 $\overline{B_r}$ 设置为 $B_{r-1}+1$，然后开始搜索具有固定活跃 S 盒个数 $\overline{B_r}$ 的截断差分特征，如果没有找到，则逐次增加活跃 S 盒个数（即 $\overline{B_r}$ 值每次增加 1）继续搜索。一旦找到一条截断差分特征，则搜索结束。这种搜索策略消除了 $\overline{B_r}$ 对搜索效率的影响，并且尽可能降低了搜索空间。

算法 3　最小差分活跃 S 盒个数搜索算法

Procedure Main

begin the program

let $\overline{B}_r = B_{r-1} + 1$ and $B_r = 0$

while $\overline{B}_r \neq B_r$ do

call Procedure Round-1

end while

exit the program

Procedure Round-1

for each candidate $\Delta \boldsymbol{X}^1$ with $\mathrm{Hw}(\Delta \boldsymbol{X}^1)$ from 1 to $\overline{B}_r - B_{r-1}$ do

$w_1 = \mathrm{Hw}(\Delta \boldsymbol{X}^1)$
$\Delta \boldsymbol{Y}^1 = \mathrm{SR}(\Delta \boldsymbol{X}^1)$
for each candidate in $\mathrm{DPDT}[\Delta Y_1^1]$ do
$\Delta X_1^2 = \mathrm{DPDT}[\Delta Y_1^1]$
$v_1 = \mathrm{Hw}(\Delta X_1^2)$
if $w_1 + v_1 + B_{r-2} > \overline{B}_r$ do
break
else
…
for each candidate in $\mathrm{DPDT}[\Delta Y_m^1]$ do
$\Delta X_m^2 = \mathrm{DPDT}[\Delta Y_m^1]$
$v_m = \mathrm{Hw}(\Delta X_m^2)$
$v = v_1 + \cdots + v_m$
if $w_1 + v + B_{r-2} > \overline{B}_r$ do
break
else
call Procedure Round-2
end if
end for
…
end if
end for
end for
return to the upper procedure

Procedure Round-*i* $(2 \leqslant i \leqslant r-1)$
$w_i = \mathrm{Hw}(\Delta \boldsymbol{X}^i)$
if $w_1 + \cdots + w_i + B_{r-i} \leqslant \overline{B}_r$ do
$\Delta \boldsymbol{Y}^i = \mathrm{SR}(\Delta \boldsymbol{X}^i)$
for each candidate in $\mathrm{DPDT}[\Delta Y_1^i]$ do
$\Delta X_1^{i+1} = \mathrm{DPDT}[\Delta Y_1^i]$
$v_1 = \mathrm{Hw}(\Delta X_1^{i+1})$
if $w_1 + \cdots + w_i + v_1 + B_{r-i-1} > \overline{B}_r$ do
break
else
…
for each candidate in $\mathrm{DPDT}[\Delta Y_m^i]$ do
$\Delta X_m^{i+1} = \mathrm{DPDT}[\Delta Y_m^i]$
$v_m = \mathrm{Hw}(\Delta X_m^{i+1})$
$v = v_1 + \cdots + v_m$
if $w_1 + \cdots + w_i + v + B_{r-i-1} > \overline{B}_r$ do
break
else
call Procedure Round-(i+1)
end if
end for
…
end if
end for
end if
return to the upper procedure

Procedure Round-*r*
$w_r = \mathrm{Hw}(\Delta \boldsymbol{X}^r)$
if $w_1 + w_2 + \cdots + w_r = \overline{B}_r$ do
$B_r = \overline{B}_r$
end if
return to the upper procedure

为了进一步提高搜索效率，算法 3 引入了一些优化策略。

首先，按照汉明重量从小到大的顺序遍历明文差分模式，一旦某个汉明重量的差分模式不满足搜索条件，即 $n_1 + B_{r-1} > \overline{B}_r$，就停止搜索，不需要遍历更高汉明重量的明文差分模式。这种策略使搜索空间非常小——只包含低汉明重量的差分模式，极大地提高了算法效率。

另外，由于差分模式分布表的输出差分模式按照汉明重量从小到大的顺序排列，对于给定的输入差分模式，每次查表都得到当前汉明重量最小的输出差分模式，即下一轮的活跃 S 盒个数，可以更有效地排除不满足条件的差分模式。一旦某个输出差分模式不满足搜索条件，就可以提前停止，不需要继续搜索下一轮。

3　本文算法应用

将本文算法应用于 SPN 型分组密码 LED（Light encryption device）[32]、SKINNY[27]、CRAFT[28]以及认证加密算法 FIDES[29]，找到了其全轮最小活跃 S 盒个数的下界。对于随机选择的 S 盒，该下界是紧致的，即对于一个特定的 S 盒，可以构造一条有效的差分/线性特征。实验结果如表 1～表 4 所示，其中，$\#S_D$ 和 $\#S_L$ 分别表示差分活跃 S 盒个数和线性活跃 S 盒个数，T_{OurA} 和 T_{MILP} 分别表示本文算法和基于 MILP 方法的运行时间，—表示未给出或者未找到相应轮数的结果。与基于 MILP 的方法相比，本文算法效率更高。

表 1　LED 最小活跃 S 盒个数

轮数	MILP		本文算法	
	$\#S_D$	T_{MILP}/s	$\#S_D$	T_{OurA} /s
1	1	—	1	0
2	5	—	5	0.01
3	9	—	9	0.01
4	25	—	25	16.62
5	26	—	26	0.01
6	30	—	30	0.01
7	34	—	34	0.01
8	50	—	50	16.59
9	51	—	51	0.01
10	55	—	55	0.01
11	59	—	59	0.01
12	75	—	75	17.02
13	76	—	76	0.02
14	80	—	80	0.02
15	84	—	84	0.02
16	100	—	100	16.55
17	—	—	101	0.02
18	—	—	105	0.02
19	—	—	109	0.02
20	—	—	125	16.55
21	—	—	126	0.02
22	—	—	130	0.02
23	—	—	134	0.02
24	—	—	150	16.53
25	—	—	151	0.02
26	—	—	155	0.02
27	—	—	159	0.02
28	—	—	175	17.01
29	—	—	176	0.02
30	—	—	180	0.02
31	—	—	184	0.02
32	—	—	200	16.58
33	—	—	201	0.02
34	—	—	205	0.02
35	—	—	209	0.02
36	—	—	225	16.57
37	—	—	226	0.03
38	—	—	230	0.03
39	—	—	234	0.03
40	—	—	250	16.55
41	—	—	251	0.03
42	—	—	255	0.04
43	—	—	259	0.04
44	—	—	275	16.93
45	—	—	276	0.03
46	—	—	280	0.03
47	—	—	284	0.03
48	—	—	300	16.55

表 2　SKINNY 最小活跃 S 盒个数

轮数	MILP			本文算法			
	$\#S_D$	$\#S_L$	T_{MILP} / s	$\#S_D$	T_{OurA} /s	$\#S_L$	T_{OurA} /s
1	1	1	—	1	0	1	0
2	2	2	—	2	0	2	0.01
3	5	5	—	5	0.01	5	0.01
4	8	8	—	8	0.01	8	0.01
5	12	13	—	12	0.01	13	0.01
6	16	19	—	16	0.01	19	0.03
7	26	25	—	26	0.18	25	0.05
8	36	32	—	36	2.67	32	0.2
9	41	38	—	41	0. 39	38	0. 21
10	46	43	—	46	0.79	43	0.16
11	51	48	—	51	0.82	48	0.2
12	55	52	—	55	0.3	52	0.1
13	58	55	—	58	0.1	55	0.02
14	61	58	—	61	0.03	58	0.02
15	66	64	—	66	0.05	64	0.03
16	75	70	—	75	0.39	70	0.07
17	82	76	—	82	0.61	76	0.11
18	88	80	—	88	0.46	80	0.02
19	92	85	—	92	0.06	85	0.03
20	96	90	—	96	0.07	90	0.05
21	102	96	—	102	0.28	96	0.15
22	108	102	—	108	0. 60	102	0.47
23	114	107	—	112	0.21	107	0.22
24	116	110	—	116	0.19	110	0.03
25	124	118	—	124	0.77	115	0.05
26	132	122	—	128	0.06	121	0.1
27	138	128	—	132	0.04	127	0.17
28	136	136	—	136	0.02	130	0.03
29	148	141	—	142	0.04	135	0.05
30	158	143	—	148	0.06	141	0.1
31	—	—	—	154	0.16	147	0.42
32	—	—	—	160	0.21	153	0.59
33	—	—	—	164	0.04	157	0.09
34	—	—	—	168	4	160	0.04
35	—	—	—	172	0.03	166	0.09
36	—	—	—	176	0.03	172	0.13
37	—	—	—	182	0.04	177	0.09
38	—	—	—	188	0.05	180	0.04
39	—	—	—	194	0.06	186	0.09
40	—	—	—	200	0.07	192	0.33

本文中所有实验都是在计算机上运行的，所用的实验平台是 Intel Core i7-6700 CPU @3.4 GHz，16 GB RAM，软件采用 VS2010，C 语言编程。

3.1　LED 分组密码

LED 是 Guo 等[32]在 CHES 2011 会议上提出的一组 64 bit 分组密码，它包含 2 个版本，即 LED-64

和 LED-128，对应的密钥长度分别为 64 bit 和 128 bit。由于 LED 的列混淆层使用 MDS 矩阵，根据宽轨迹策略，可以证明 LED 任意连续 4 轮的最小活跃 S 盒个数为 25 个。

本文通过自动化搜索程序找到了 LED-64 和 LED-128 全轮的最小活跃 S 盒个数。对于 LED-64，找到差分和线性活跃 S 盒一共需要 134 s。对于 LED-128，找到差分和线性活跃 S 盒一共需要 201 s。实验结果如表 1 所示，由于差分活跃 S 盒与线性活跃 S 盒个数相同，表 1 仅列出了最小差分活跃 S 盒个数。表 1 中 r 轮的时间是指在已知 $r-1$ 轮最小活跃 S 盒个数的条件下，搜索 r 轮最小活跃 S 盒个数的时间，因此总的搜索时间是累积前 r 轮的时间。

3.2 SKINNY 分组密码

在 CRYPTO 2016 会议上，Beierle 等[27]提出了一簇可调轻量级分组密码 SKINNY，它包含 2 种分组长度：64 bit 和 128 bit，分别记为 SKINNY-64 和 SKINNY-128。为了评估 SKINNY 抵抗差分密码分析和线性密码分析的安全性，设计者使用基于 MILP 的方法来搜索最小活跃 S 盒个数，并找到了 22 轮最小差分活跃 S 盒个数，以及 23 轮最小线性活跃 S 盒个数。对于更多轮数，由于 MILP 求解器的运行时间太长，他们只提供了最小活跃 S 盒个数的上界。

本文找到了 SKINNY 全轮的最小活跃 S 盒个数，其中，搜索 40 轮最小差分活跃 S 盒个数需要 10 s，最小线性活跃 S 盒个数需要 5 s，与基于 MILP 的方法相比效率非常高。实验结果如表 2 所示。由于 SKINNY-64 和 SKINNY-128 使用相同的二元域矩阵（分别作用在 $\mathbb{F}_2^4$ 和 $\mathbb{F}_2^8$ 上），根据定理 3，SKINNY-64 和 SKINNY-128 的差分/掩码模式分布表相同，因此它们具有相同的活跃 S 盒个数。

3.3 CRAFT 分组密码

CRAFT 是 Beierle 等[28]提出的一个可调轻量级分组密码，其分组长度为 64 bit，密钥长度为 128 bit，迭代轮数为 31 轮。它的设计目标是实现对差分故障攻击的有效防护，以及以很少的额外代价同时实现加密和解密。为此，CRAFT 使用了对合的密码部件，并且没有使用密钥扩展算法。在算法分析方面，设计者同时使用 Matsui 算法和基于 MILP 的方法来评估其抵抗差分密码分析和线性密码分析的安全性，得到了 17 轮最小差分活跃 S 盒个数的下界。

本文找到了 CRAFT 全轮的最小活跃 S 盒个数，算法运行时间约为 2 s，实验结果如表 3 所示。由于 CRAFT 使用对合的二元域矩阵，差分模式分布表和掩码模式分布表相同，因此差分活跃 S 盒个数与线性活跃 S 盒个数相同，表 3 只列出了最小差分活跃 S 盒个数。另外，与算法设计者直接使用 Matsui 算法相比，本文的优化策略对于 Matsui 算法的效率提升非常明显。

表 3 CRAFT 最小活跃 S 盒个数

轮数	MILP		本文算法	
	$\#S_D$	T_{MILP} / s	$\#S_D$	T_{OurA} /s
1	1	—	1	0
2	2	—	2	0.01
3	4	—	4	0.02
4	6	—	6	0.03
5	10	—	10	0.03
6	14	—	14	0.04
7	20	—	20	0.04
8	26	—	26	0.05
9	32	—	32	0.05
10	36	—	36	0.06
11	40	—	40	0.06
12	44	—	44	0.06
13	48	—	48	0.06
14	52	—	52	0.06
15	56	—	56	0.08
16	60	—	60	0.08
17	64	—	64	0.09
18	—	—	68	0.14
19	—	—	72	0.11
20	—	—	76	0.1
21	—	—	80	0.11
22	—	—	84	0.11
23	—	—	88	0.13
24	—	—	92	0.1
25	—	—	96	0.1
26	—	—	100	0.1
27	—	—	104	0.11
28	—	—	108	0.12
29	—	—	112	0.12
30	—	—	116	0.12
31	—	—	120	0.12

3.4 FIDES 认证加密算法

FIDES 是面向硬件的轻量级认证加密算法[29]，

采用类似 Duplex Sponge 结构和专门设计的内部置换。FIDES 包含 2 个版本：FIDES-80 和 FIDES-96，其内部状态分别为 160 bit 和 192 bit。FIDES 的加密算法采用 SPN 结构，S 盒分别使用 5 bit 的 AB（almost bent）函数和 6 bit 的 APN（almost perfect nonlinear）函数，其初始化和生成标签的过程分别执行 16 次轮函数迭代。

根据宽轨迹设计策略，FIDES 的任意连续 4 轮至少有 16 个活跃 S 盒。为了得到更好的安全界，设计者使用基于 MILP 的方法搜索最小活跃 S 盒个数，并找到了 8 轮最小活跃 S 盒个数。然而，该 MILP 模型并没有精确刻画列混淆矩阵的差分/掩码模式传播，因此无法保证 5～8 轮最小活跃 S 盒个数的下界是紧致的。

本文找到了 FIDES 全轮的最小活跃 S 盒个数，由于本文通过遍历列混淆矩阵的输入差分/掩码来计算其所有可能的差分/掩码模式传播，因此得到了最小活跃 S 盒个数的紧致下界，实验结果如表 4 所示。由于列混淆层使用对合二元域矩阵，其差分模式分布表和掩码模式分布表相同，因此差分活跃 S 盒个数与线性活跃 S 盒个数相同，表 4 只列出了最小差分活跃 S 盒个数。另外，根据定理 3，FIDES-80 和 FIDES-96 的差分/掩码模式分布表相同，因此具有相同的活跃 S 盒个数。

表 4　　FIDES 最小活跃 S 盒个数

轮数	MILP		本文算法	
	$\#S_D$	T_{MILP} / s	$\#S_D$	T_{OurA} /s
1	1	—	1	0
2	4	—	4	0.01
3	7	—	7	0.01
4	16	—	16	8.97
5	22	—	25	21.94
6	32	—	36	229.86
7	42	—	47	430.37
8	48	—	60	0.83×3 600
9	—	—	66	8.44
10	—	—	72	16.42
11	—	—	77	0.31
12	—	—	86	12.31
13	—	—	95	24.28
14	—	—	104	12.27
15	—	—	114	55.99
16	—	—	124	57.96

4　结束语

本文研究了分组密码常用的 MDS 矩阵和二元域矩阵的差分和掩码传播性质，提出了差分模式分布表和掩码模式分布表的概念，证明了构造差分/掩码模式分布表的计算复杂度的下界，并给出了快速构造方法。基于 Matsui 算法，提出了一种自动化搜索 SPN 型分组密码最小活跃 S 盒个数的快速算法，通过引入一些优化策略，极大提高了 Matsui 算法的效率。针对 SPN 型分组密码 LED、SKINNY、CRAFT 以及认证加密算法 FIDES，给出了其全轮最小活跃 S 盒个数。实验结果表明，本文算法的效率比基于 MILP 的方法高。

参考文献：

[1] BIHAM E, SHAMIR A. Differential cryptanalysis of DES-like cryptosystems[J]. Journal of Cryptology, 1991, 4(1): 3-72.

[2] MATSUI M. Linear cryptanalysis method for DES cipher[C]//Proceedings of International Conference on the Theory and Application of Cryptographic Techniques. Berlin: Springer, 1993: 386-397.

[3] DAEMEN J, RIJMEN V. The design of Rijndael: AES - the advanced encryption standard[M]. Berlin: Springer, 2002.

[4] MATSUI M. On correlation between the order of S-boxes and the strength of DES[C]//Proceedings of International Conference on the Theory and Application of Cryptographic Techniques. Berlin: Springer, 1994: 366-375.

[5] OHTA K, MORIAI S, AOKI K. Improving the search algorithm for the best linear expression[C]//Proceedings of 15th Annual International Cryptology Conference. Berlin: Springer, 1995: 157-170.

[6] AOKI K, KOBAYASHI K, MORIAI S. Best differential characteristic search of FEAL[C]//Proceedings of International Conference on Fast Software Encryption. Berlin: Springer, 1997: 41-53.

[7] ARNAUD B, NICOLAS B, ERIC F. Automatic search for a maximum probability differential characteristic in a substitution-permutation network[C]//Proceedings of the 48th Hawaii International Conference on System Sciences. Piscataway: IEEE Press, 2015: 5165-5174.

[8] JI F L, ZHANG W T, DING T Y. Improving Matsui's search algorithm for the best differential/linear trails and its applications for DES, DESL and GIFT[J]. The Computer Journal, 2020, 64(4): 610-627.

[9] KIM S, HONG D, SUNG J, et al. Accelerating the best trail search on AES-like ciphers[J]. IACR Transactions on Symmetric Cryptology, 2022, 2022(2): 201-252.

[10] MOUHA N, WANG Q, GU D, et al. Differential and linear cryptanalysis using mixed-integer linear programming[C]//Proceedings of International Conference on Information Security and Cryptology. Berlin: Springer, 2011: 57-76.

[11] SUN S W, HU L, WANG P, et al. Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers[C]//Proceedings of the 20th International Conference on the

Theory and Application of Cryptology and Information Security. Berlin: Springer, 2014: 158-178.

[12] ABDELKHALEK A, SASAKI Y, TODO Y, et al. MILP modeling for (large) S-boxes to optimize probability of differential characteristics[J]. IACR Transactions on Symmetric Cryptology, 2017, 2017(4): 99-129.

[13] QUINE W V. The problem of simplifying truth functions[J]. The American Mathematical Monthly, 1952, 59(8): 521-531.

[14] MCCLUSKEY E J J. Minimization of Boolean functions[J]. Bell System Technical Journal, 1956, 35(6): 1417-1444.

[15] BRAYTON R K, HACHTEL G D, MCMULLEN C T, et al. Logic minimization algorithms for VLSI synthesis[M]. Berlin: Springer, 1984.

[16] ZHANG Y, SUN S, CAI J, et al. Speeding up MILP aided differential characteristic search with Matsui's strategy[C]//Proceedings of International Conference on Information Security. Berlin: Springer, 2018: 101-115.

[17] ZHOU C N, ZHANG W T, DING T Y, et al. Improving the MILP-based security evaluation algorithm against differential/linear cryptanalysis using a divide-and-conquer approach[J]. IACR Transactions on Symmetric Cryptology, 2019, 2019(4):438-469.

[18] FU K, WANG M Q, GUO Y H, et al. MILP-based automatic search algorithms for differential and linear trails for speck L[C]//Proceedings of International Conference on Fast Software Encryption. Berlin: Springer, 2016: 268-288.

[19] XIANG Z J, ZHANG W T, BAO Z Z, et al. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers[C]//Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2016: 648-678.

[20] SASAKI Y, TODO Y. New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers[C]//Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2017: 185-215.

[21] BOURA C, COGGIA D. Efficient MILP modelings for Sboxes and linear layers of SPN ciphers[J]. IACR Transactions on Symmetric Cryptology, 2020, 2020(3): 99-129.

[22] BAO Z, DONG X, GUO J, et al. Automatic search of meet-in-the-middle preimage attacks on AES-like hashing[C]//Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2021: 771-804.

[23] UDOVENKO A. Convexity of division property transitions: theory, algorithms and compact models[C]//Proceedings of the 27th International Conference on the Theory and Application of the Cryptology and Information Security. Berlin: Springer, 2021: 332-361.

[24] WANG Q, HAO Y, TODO Y, et al. Improved division property based cube attacks exploiting algebraic properties of superpoly[C]//Proceedings of the 38th Annual International Cryptology Conference. Berlin: Springer, 2018: 275-305.

[25] GUO H, SUN S W, SHI D P, et al. Differential attacks on CRAFT exploiting the involutory S-boxes and tweak additions[J]. IACR Transactions on Symmetric Cryptology, 2020, 2020(3): 119-151.

[26] DERBEZ P, LAMBIN B. Fast MILP models for division property[J]. IACR Transactions on Symmetric Cryptology, 2022, 2022(2):: 289-321.

[27] BEIERLE C, JEAN J, KÖLBL S, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS[C]//Proceedings of the 36th Annual International Cryptology Conference. Berlin: Springer, 2016:123-153.

[28] BEIERLE C, LEANDER G, MORADI A, et al. CRAFT: lightweight tweakable block cipher with efficient protection against DFA attacks[J]. IACR Transactions on Symmetric Cryptology, 2019, 2019(1): 5-45.

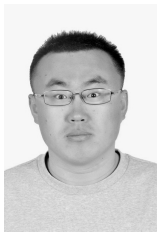
[29] BILGIN B, BOGDANOV A, KNEZEVIC M, et al. Fides: lightweight authenticated cipher with side-channel resistance for constrained hardware[C]//Proceedings of International Conference on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2013: 142-158.

[30] 王念平, 郭祉成. 动态密码结构抵抗差分密码分析能力评估[J]. 通信学报, 2021, 42(8): 70-79.
WANG N P, GUO Z C. Security evaluation against differential cryptanalysis for dynamic cryptographic structure[J]. Journal on Communications, 2021, 42(8): 70-79.

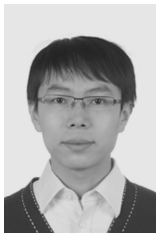
[31] MACWILLIAMS F, SLOANE N. The theory of error-correcting codes[M]. Amsterdam: North-Holland Publishing Company, 1981.

[32] GUO J, PEYRIN T, POSCHMANN A, et al. The LED block cipher[C]//Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2011: 326-341.

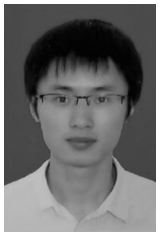
[作者简介]



刘正斌（1985– ），男，山东青岛人，博士，保密通信重点实验室高级工程师，主要研究方向为对称密码算法设计、密码算法自动化分析等。



李永强（1982– ），男，吉林集安人，博士，中国科学院信息工程研究所副研究员，主要研究方向为对称密码算法、布尔函数等。



朱朝熹（1992– ），男，重庆人，博士，保密通信重点实验室工程师，主要研究方向为序列密码的设计与分析等。