第 26 卷第 6 期 电 子 与 信 息 学 报 Vol.26No.6
2004 年 6 月 Journal of Electronics & Information Technology June 2004

Twofish 算法中密钥相关 S-盒的差分性质分析及其改进 [1]

周 旋* 李 超**

*(国防科技大学理学院数学与系统科学系 长沙 410073)
**(中国科学院软件研究所计算机科学重点实验室 北京 100080)

摘 要: 该文从理论上证明了 Twofish 算法中，密钥越长，密钥相关 S- 盒的差分概率就越小，提出了一种新的与密钥作用的方式来产生密钥相关 S-盒的方法，理论与测试结果表明新的 S-盒的“异或”差分概率和“模加”差分概率比原算法的差分概率要小.

关键词: Twofish, 差分分析, S-盒
中图分类号: TN918, O441 **文献标识码**: A **文章编号**: 1009-5896(2004)06-0912-05

Differential Analysis and Modification of the Key-Dependent S-Boxes of Twofish

Zhou Xuan Li Chao

*(*Dept. of Math. and System Sci., Nat. Univ. of Defense Tech., Changsha* 410073, *China*)

**(*Key Lab. of Computer Sci., Inst. of Software, CAS, Beijing* 100080, *China*)

Abstract This paper proves that the longer the key of Twofish is,the smaller the differential probabilities of the key-dependent S-boxes are. A new method is proposed to produce the key-dependent S-boxes. Theory and simulation results show that the XOR differential probability and modular addition differential probability of the modified S-boxes are smaller than those of the original S-boxes.

Key words Twofish, Differential analysis, S-boxes

1 引言

Twofish 算法是由 Bruce Schneier 等在 1998 年提出的一个算法 [1]，它是一个 128-bit 明文分组，密钥可变长的分组密码，采用了密钥相关 S-盒的设计方法. 本文中所提到的 Twofish 算法的密钥长度如不作说明均指 128-bit . Twofish 算法是按照 NIST(National Institute of Standard Technology) 所发布的征集 AES(Advanced Encryption Standard) 候选算法的要求进行设计的. 其基本思想是在 Blowfish 已经得到许多安全性分析的基础上，构造一个分组为 128-bit 的密钥可变长的分组密码以适应 NIST 提出的要求. Blowfish 算法是由 Bruce Schneier 在 1993 年提出的分组为 64-bit 的密钥可变长的分组密码 [2] . 事实表明，Twofish 是一个性能较好的分组密码算法，成为了 AES 最后一轮的五个候选算法之一. 它吸收了 Square 密码中的 MDS(Maximum Distance Separable) 码矩阵及 Safer 系列密码中的 PHT(Pseudo-Hadamard Transforms) 结构.

自从 DES(Data Encryption Standard) 算法中使用 S-盒以来，针对 S-盒在密码算法中所起的作用及如何对 S-盒进行分析，人们进行了比较多的讨论，其中 Biham 和 Shamir 在 1991 年提出的针对 DES 的差分密码分析主要集中在对 S-盒进行差分分析 [3] . 分组密码的差分性质已成为其安全性能的一个重要指标. 对大多数采用 S-盒的分组密码算法的差分性质进行分析主

[1] 2003-02-27 收到， 2003-08-08 改回
国防科技大学基础研究基金 (JC02-02-007) 和中国科学院软件研究所计算机科学重点实验室开放基金 (syskf0402) 资助课题

要是集中在对其 S-盒的差分性质进行分析．密钥相关 S-盒的设计增加了对 S-盒的各种性质进行分析的难度，因此对密码分析者来说，攻击将变得更加困难．

Twofish 算法中的 MDS 码矩阵运算对“异或”是线性的， PHT 运算对“模 2^{32} 加”是线性的．密钥相关 S-盒对两种运算均是非线性的，所以在本文中我们讨论 S-盒的“异或”差分性质与“模加”差分性质．并依据这两种差分性提出一种新构造方法，来降低差分概率．

2 符号与定义

本文除特别标记的外，使用文献 [1] 中的记号．运算均为域 $(F_{2^8},+,\bullet)$ 上的运算，即为相应的字节运算．

$\oplus$：两字节按比特进行“异或”； $+$：两字节相加，取模 256 的最小非负余数； $-$：两字节相减，取模 256 的最小非负余数．

定义 1　“异或”差分：有序对数据块 (X_i,X_i^*) 的“异或”差分为 $\Delta X_i=X_i\oplus X_i^*$

定义 2　“模加”差分：有序对数据块 (X_i,X_i^*) 的“模加”差分为 $\nabla X_i=X_i-X_i^*$[4]

$$P_i(a,b)=\Pr_x(q_i[x]\oplus q_i[x\oplus a]=b),\quad P_i^*(a,b)=\Pr_x(q_i[x]-q_i[(x-a)]=b),\ i=0,1$$
$$Ps_i(a,b)=\Pr_x(s_i[x]\oplus s_i[x\oplus a]=b),\quad P^*s_i(a,b)=\Pr_x(s_i[x]-s_i[(x-a)]=b),\ i=0,1,2,3$$
$$Ps_i^*(a,b)=\Pr_x(s_i^*[x]\oplus s_i^*[x\oplus a]=b),\quad P^*s_i^*(a,b)=\Pr_x(s_i^*[x]-s_i^*[(x-a)]=b),\ i=0,1,2,3$$

3 S-盒的改进

本文所提出的对 S-盒的改进如表 1 所示：

表 1　原 S-盒与改进 S-盒的对照表

原 S-盒		改进 S-盒	
s_0	$q_1[q_0[q_0[x]\oplus s_{0,0}]\oplus s_{1,0}]$	s_0^*	$q_1[q_0[q_0[x]\oplus s_{0,0}]+s_{1,0}]$
s_1	$q_0[q_0[q_1[x]\oplus s_{0,1}]\oplus s_{1,1}]$	s_1^*	$q_0[q_0[q_1[x]\oplus s_{0,1}]+s_{1,1}]$
s_2	$q_1[q_1[q_0[x]\oplus s_{0,2}]\oplus s_{1,3}]$	s_2^*	$q_1[q_1[q_0[x]\oplus s_{0,2}]+s_{1,3}]$
s_3	$q_0[q_1[q_1[x]\oplus s_{0,3}]\oplus s_{1,3}]$	s_3^*	$q_0[q_1[q_1[x]\oplus s_{0,3}]+s_{1,3}]$

在文献 [5] 中， X J. Lai 提到 IDEA 密码中的混乱是通过混合不相容的群运算来达到的．IDEA 中并没有使用 S-盒，但与 S-盒有着同样的作用． $\oplus$ 与 $+$ 之间没有明显的代数关系．这提示如果在 Twofish 算法中的密钥相关 S-盒中采用类似的方法将提高其密码性质．下面就其差分性质进行分析．

4 密钥相关 S-盒的差分性质分析

为方便，以下我们只对 4 个 S-盒中的 s_0 进行分析，其余 3 个类似．

定理 1　在置换 q_0, q_1 退化的情况下， s_0^* 的差分性质要优于 s_0 的差分性质．

证明　设 q_0, q_1 是恒等变换，则 $s_0(x)=q_1[q_0[q_0[x]\oplus s_{0,0}]\oplus s_{1,0}]=(x\oplus s_{0,0})\oplus s_{1,0}=x\oplus(s_{0,0}\oplus s_{1,0})$， $s_0^*(x)=q_1[q_0[q_0[x]\oplus s_{0,0}]+s_{1,0}]=(x\oplus s_{0,0})+s_{1,0}$ ．对“异或”差分：显然 $\forall x,y$ ，恒有 $s_0(x)\oplus s_0(y)=x\oplus y$ ；即当 $a=b$ 时， $Ps_0(a,b)=1$ ；当 $a\neq b$ 时， $Ps_0(a,b)=0$ ；而 $\forall a,b(a,b\neq 0,128)$ ，则 $Ps_0^*(a,b)\neq 1$ 且其最大值为 $1/2$ ．可见 s_0^* 的“异或”差分性质要好于 s_0 ．对“模加”差分性质： $s_0(x)-s_0(y)=(x\oplus(s_{0,0}\oplus s_{1,0}))-(y\oplus(s_{0,0}\oplus s_{1,0}))$， $s_0^*(x)-s_0^*(y)=((x\oplus s_{0,0})+s_{1,0})-((y\oplus s_{0,0})+s_{1,0})=x\oplus s_{0,0}-y\oplus s_{0,0}$ ．当 $s_{0,0}$ 与 $s_{1,0}$ 是均匀分布时，这两个等式等价，因此 s_0 与 s_0^* 的模加差分性质完全一样．综上所述，结论成立．

证毕

引理 1[6]　设 $s(x)=q_i[q_j[x]\oplus \mathrm{key}]$ ，记 $P_s(a,b)=P_{ij}(a,b)$ 为： s 中的输入 x 与密钥 key 均匀分布时，输入“异或”差分为 a ，输出“异或”差分为 b 时的概率．则有 $P_{ji}(a,b)=\Sigma_d P_i(a,d)P_j(d,b)$ ．

定理 2 在引理 1 的条件下，有 $\min_{a\neq 0,b\neq 0} P_{ji}(a,b) \leq \min(\max_{a\neq 0,b\neq 0} P_j(a,b), \max_{a\neq 0,b\neq 0} P_i(a,b))$.

证明 由于 q_i 和 q_j 是置换，显然对任意的 a 和 b ，有 $\Sigma_d P_i(a,d)=1, \Sigma_d P_j(d,b)=1$. 且对 $b\neq 0$, $P_i(0,b)=0$. 对 $a\neq 0$, $P_i(a,0)=0$. 由引理 1 ，对 $a\neq 0, b\neq 0$ ，有：

$P_{ji}(a,b)=\Sigma_d P_i(a,d)P_j(d,b) \leq \Sigma_d(\max_{a\neq 0,b\neq 0} P_i(a,d))P_j(d,b)=(\max_{a\neq 0,b\neq 0} P_i(a,d))$ $\Sigma_d P_j(d,b)=\max_{a\neq 0,b\neq 0} P_i(a,d)$.

由 a, b, d 的任意性，知 $\max_{a\neq 0,b\neq 0} P_{ji}(a,b) \leq \max_{a\neq 0,b\neq 0} P_i(a,b)$.

同理 $\max_{a\neq 0,b\neq 0} P_{ji}(a,b) \leq \max_{a\neq 0,b\neq 0} P_j(a,b)$. 故结论成立. 证毕

引理 2 设 $s^*(x)=q_i[q_j[x]+\text{key}]$ ，记 $P^*_{s^*}(a,b)=P^*_{ij}(a,b)$ ，表示 s^* 中当输入 x 与密钥 key 均匀分布时，输入“模加”差分为 a ，输出“模加”差分为 b 时的概率，则有 $P^*_{ji}(a,b)=\Sigma_d P^*_i(a,d)P^*_j(d,b)$.

定理 3 在引理 2 的条件下，有 $\max_{a\neq 0,b\neq 0} P^*_{ji}(a,b) \leq \min(\max_{a\neq 0,b\neq 0} P^*_j(a,b), \max_{a\neq 0,b\neq 0} P^*_i(a,b))$.

证明 与定理 2 的证明类似，略.

用 $P(s_0,N)$ 表示 Twofish 算法的密钥长度为 Nbit 时所产生的密钥相关 S-盒 s_0 的最大“异或”差分概率.

推论 1 $P(s_0,256)<P(s_0,192)<P(s_0,128)$.

证明 记 $P_{s_0\_N}(a,b)$ 表示密钥长度为 Nbit 时，s_0 盒的“异或”输入差分为 a ，“异或”输出差分为 b 的概率. 将引理 1 中的 P_{ij} 的定义扩展至多层得 $P_{ij\cdots m}(a,b)=\Sigma_d P_i(a,d)P_{j\cdots m}(d,b)$, 则 $P_{s_0\_256}(a,b)=P_{11001}(a,b)$, $P_{s_0\_192}(a,b)=P_{1001}(a,b)$, $P_{s_0\_128}(a,b)=P_{001}(a,b)$ ，由定理 2 ，$\max_{a\neq 0,b\neq 0} P_{11001}(a,b) \leq \max_{a\neq 0,b\neq 0} P_{1001}(a,b) \leq \max_{a\neq 0,b\neq 0} P_{1001}(a,b)$. 事实上，由定理 2 的证明过程可以看出，等号成立等价于当 $P_i(a,d)$ 不取最大值时，对任意的 b 有 $P_j(d,b)=0$ ；或当 $P_j(d,b)$ 不取最大值时，对任意的 a 有 $P_i(a,d)=0$. 而这是不可能的，因为对任意的替换 S-盒来说，差分总是不均匀的，总存在 $a_0\neq 0$, $d_0\neq 0$ 使得 $P_i(a_0,d_0)<\max_{a\neq 0,b\neq 0} P_i(a,b)$ ，若对任意 b 有 $P_j(d_0,b)=0$, 则 $\Sigma_b P_j(d_0,b)=0$ 这与 P_j 是置换有 $\Sigma_b P_j(d_0,b)=1$ 矛盾. 因此等号不成立. 证毕

显然推论 1 的结果对于另外 3 个 S 盒也同样成立.

在 S-盒的构造中所用到的 q_0, q_1 最大“异或”差分概率均为 10/256 ，最大“模加”差分概率均为 8/256 . 这样的差分概率虽比较小，但未达到 8×8 S-盒的最佳差分情形 ($\delta\leq 2^{-6}$) . 在通过各种长度的密钥作用后， s_0 的“异或”差分概率均达到了很好的界，接近 2^{-8} . 本质上，当 $s_{0,0}$ 与 $s_{1,0}$ 分别遍历每个字节时，原 q_0, q_1 中差分概率较大的对分散开了，从而从总体上降低其最大差分概率.

定理 4 设 $P0$, $P1$ 是两个置换， $P(x)=P1\circ P0(x)$ ，则 P 的差分概率分布不能仅由 $P0$, $P1$ 的差分概率分布确定，而且还与 $P0$, $P1$ 本身有关.

证明 设 $s(x)=q_j(q_i(x)\oplus y)$ ，给定一个 y 将产生一个对应的 S-盒. 记 $P0_y(x)=q_i(x)\oplus y$, $P1(x)=q_j(x)$. 当 $y1\neq y2$ 时， $P0_{y1}$ 与 $P0_{y2}$ 是两个不同的置换，但它们显然有相同的“异或”差分概率分布 (与 q_i 相同) . 而 $P1\circ P0_{y1}$ 与 $P1\circ P0_{y2}$ 之间有不同的“异或”差分概率分布. 对“模加”差分的情况证明类似. 故结论成立. 证毕

由定理 4 知，对于固定的两个置换，要求它们结合后产生的新置换的差分概率分布是不可能的. 文献 [5] 中有结果 $\oplus$ 与 $+$ 不满足广义结合律与分配律. 因此在对 $s(x)=q_j(q_i(x)\oplus y)$ 求其“模加”差分概率分布时没有引理 2 中那样的结果，同样对 $s^*(x)=q_j(q_i(x)+y)$ ，求其“异或”差分分布时没有引理 1 中那样的结果. 因此我们对改动的细节进行全面统计，然后在此基础上对新的 S-盒的性质进行分析. 统计方法是：一个密钥 key 将产生一个新的 S-盒，对此 S-盒

进行差分概率分布统计，256 个 key 值 (0~255) 对应 256 个差分概率分布，将对应位置的值求和，再平均，得到整个 S-盒的差分概率分布表，从中算出最大值即为此 S-盒的最大差分概率。结果如表 2 所示

表 2　改进环节与原环节的最大差分概率对照表

s	最大“异或”差分概率 $(\times 2^{-16})$	最大“模加”差分概率 $(\times 2^{-16})$
$s_{00}[x] = q_0[q_0[x] \oplus \text{key}]$	440	354
$s^*_{00}[x] = q_0[q_0[x] + \text{key}]$	368	335
$s_{01}[x] = q_0[q_1[x] \oplus \text{key}]$	420	368
$s^*_{01}[x] = q_0[q_1[x] + \text{key}]$	354	325
$s_{10}[x] = q_1[q_0[x] \oplus \text{key}]$	416	328
$s^*_{10}[x] = q_1[q_0[x] + \text{key}]$	372	327
$s_{11}[x] = q_1[q_1[x] \oplus \text{key}]$	428	348
$s^*_{11}[x] = q_1[q_1[x] + \text{key}]$	364	328

表 2 中的 key 指均匀分布的随机密钥。从表 3 可以看出改进后的环节要比原有环节具有更好的差分性质。

推论 2　改进后的 S-盒将以较大的概率具有更好的差分性质。

证明　令 $s_0[x] = s_{100}[x] = s_{10}[q_0[x] \oplus \text{key}]$，$s^*_0[x] = s^*_{100}[x] = s^*_{10}[q_0[x] \oplus \text{key}]$. 类似推论 1 中所用的记号，用 $P(s)$ 表示 S-盒 s 的最大“异或”方差概率。由定理 2，$\max(P(s_{100})) \le \min(\max(P_0), \max(P(s_{10}))) = \max(P(s_{10}))$. $\max(P(s^*_{100})) \le \min(\max(P_0), \max(P(s^*_{10}))) = \max(P(s^*_{10}))$. 而由表 3 可以看出 $\max(P(s^*_{10})) < \max(P(s_{10}))$. 而且改进的环节中采用了模加运算，这对于异或差分的传播更快。因此 $\max(P(s^*_{001}))$ 将会以较大的概率小于 $\max(P(s_{001}))$. 即：s^*_0 的“异或”差分性质优于 s_0 的异或差分性质。　　证毕

对于其余的 3 个 S-盒情况类似。更进一步，可以将情况推广到 192-bit 与 256-bit 的情况，结论依然成立。

但对模加差分性质的分析较为困难，我们通过统计进行补充说明。

给定一个密钥，相应产生一个密钥相关 S-盒，对每一个密钥相关 S-盒计算其最大差分概率，然后对 65536 个密钥相关 S-盒按照其最大差分概率进行分类，并统计出每一类密钥相关 S-盒的个数，结果如表 3 所示

表 3　改进 S-盒 S^*_i 与原 S-盒 S_i 对应不同最大差分概率的个数分布统计表

		s_0	s^*_0	s_1	s^*_1	s_2	s^*_2	s_3	s^*_3
最大“异或”差分概率 $(\times 2^{-8})$	8	1	3	3	3	2	2		1
	10	26178	25845	26020	25694	25937	25987	26021	25912
	12	35106	35443	35322	35541	35277	35292	35321	35387
	14	3968	3973	3919	4023	4058	3994	3908	3984
	16	269	258	258	266	245	248	269	239
	18	14	13	14	9	17	13	17	13
	20		1						
最大“异或”差分概率 $(\times 2^{-8})$	6	4304	4292	4299	4335	4289	4284	4344	4337
	7	42253	42330	42174	42154	42199	42270	42228	42183
	8	16505	16499	16654	16739	16612	16545	16537	16570
	9	2205	2181	2139	2064	2154	2179	2191	
	10	245	217	243	219	252	262	226	233
	11	23	14	24	24	19	19	16	
	12	1	3	3	1	1	2	2	6
	13							1	

在表 3 的基础上，我们定义了密钥相关 S-盒的平均差分概率，即将所有密钥相关 S-盒的最大差分概率求和然后平均，结果如表 4 所示。

另一种统计方法是对每一个密钥相关 S-盒计算出其差分概率分布表，再将所有密钥相关 S-盒的差分概率分布表对应位置的值求和，然后平均．我们称这样的结果为密钥相关 S-盒的差分概率分布表，其中的最大值即为密钥相关 S-盒的最大差分概率．对于原算法中的密钥相关 S-盒，这种方法统计出的结果与引理 1 推出的结果完全一致．结果如表 4 所示．

表 3 和表 4 的结果表明，改进后的 S-盒的“异或”差分与“模加”差分性质都比原 S-盒的差分性质好．

表 4 改进 S-盒 S_i^* 与原 S-盒 S_i 差分概率统计表

	s_0	s_0^*	s_1	s_1^*	s_2	s_2^*	s_3	s_3^*
平均“异或”差分概率 ($\times 2^{-8}$)	11	11	11	11	11	11	11	11
平均“模加”差分概率 ($\times 2^{-8}$)	7	7	7	7	7	7	7	7
密钥相关 S-盒的最大“异或”差分概率 ($\times 2^{-8}$)	1.0649	1.0380	1.0566	1.0387	1.0533	1.0410	1.0538	1.0385
密钥相关 S-盒的最大“模加”差分概率 ($\times 2^{-8}$)	1.0276	1.0238	1.0285	1.0220	1.0459	1.0213	1.0296	1.0223

5 结论

本文中提出了 Twofish 算法中密钥相关 S-盒的一种改进方法，从理论上初步探索了差分性质变好的本质原因，并给出了改进后的 S-盒的差分概率，结果显示改进后的 S-盒降低了差分概率，说明改进后的 S-盒提高了抗差分攻击的能力，并由于采用了混合不同群运算，进一步提高了抗未知攻击方法的能力．沿着这个思路下去，可以将“模加”操作从 8-bit 扩展至 32-bit ．这样将会把密钥相关 S-盒由 4 个 8×8 的 S-盒变为一个 32×32 的 S-盒，从而增加安全性．另外，在 pentium 处理器中，可以采用 LEA 指令实现寻址与运算并行，因此可以提高运算速度，但并不增加开销．从安全与速度等方面来看，我们的改进工作都是有意义的．

参 考 文 献

[1] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall. Twofish: A 128-bit block cipher. http://www.counterpane.com/twofish.html.

[2] Schneier B. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish), Fast Software Encryption, Cambridge Security Workshop Proceedings, (December 1993), Springer-Verlag, 1994: 191–204.

[3] Eli Biham, Adi Shamir. Differential cryptanalysis of the DES-like cryptosystems. *Journal of Cryptology*, 1991, 3(1): 72.

[4] 冯登国，吴文玲．分组密码的设计与分析．北京：清华大学出版社， 2000, 9: 23.

[5] Lai X J. On the Design and Security of Block Ciphers. ETH Series in Information Processing, Konstanz: Hartung-Gorre Verlag, 1992, Vol.1, Ch.3: 25–33.

[6] Niels Ferguson. Twofish Technical Report #1, Upper bounds on differential characteristics in Twofish. http://www.counterpane.com/twofish.html.

周 旋： 男， 1976 年生，硕士生，主要研究方向为：编码密码理论及其应用．

李 超： 男， 1966 年生，教授，博士生导师，主要研究方向为：编码密码理论及其应用．