SHA-3置换函数的差分转移概率分析

高晓东[1,2] 杨亚涛[1] 李子臣[1]
(北京电子科技学院 北京 100070)[1] (西安电子科技大学通信工程学院 西安 710071)[2]

摘 要 通过对SHA-3算法置换函数Keccak-f的分析,提出三维数组的循环移位方法。根据置换函数Keccak-f每一步变换的结构,构造出输出差分的布尔函数表达式。通过研究输出差的差分布尔函数表达式,证明了Keccak-f每一步变换的输入输出差分通过循环移位后,其差分转移概率不变。在此基础上,通过分析得出,当Keccak-f每一步变换的两个输入差分之间和对应输出差分之间均满足相同循环移位特性时,整个置换函数Keccak-f的输入输出差分在循环移位后,其差分转移概率不变。

关键词 SHA-3,循环移位,差分分析,差分转移概率
中图法分类号 TP393.08 **文献标识码** A

Differential Transition Probability Analysis of SHA-3 Permutation Function

GAO Xiao-dong[1,2] YANG Ya-tao[1] LI Zi-chen[1]
(Beijing Electronic Science and Technology Institute, Beijing 100070, China)[1]
(Communication Engineering Institute, Xidian University, Xi'an 710071, China)[2]

Abstract By analyzing the permutation function Keccak-f of SHA-3, cycle shift method of three-dimensional array was proposed. According to structure of every step transform in Keccak-f, the boolean expression of the output difference was structured. By analyzing the boolean expressions of the output difference, to the every step transform of Keccak-f, it was proved that the differential transition probability about cycle shift is unchanged. On this basis, by analyzing, it was obtained that when cycle shift properties of two input difference and two corresponding output difference are same, the differential transition probability of the whole permutation function Keccak-f about cycle shift is unchanged.

Keywords SHA-3, Cycle shift, Differential analysis, Differential transition probability

1 引言

2012年10月2日,美国国家标准技术研究所(NIST)宣布Keccak算法为新的Hash标准算法——SHA-3[1]。Keccak算法是由意法半导体的Guido Bertoni、Joan Daemen等设计开发的[2]。经过三轮的筛选[3-5],Keccak算法以其独特的结构和高的运算效率摘得了第三代Hash标准的王牌[6]。

差分攻击[7]是攻击Hash数比较有效的攻击方法,对SHA-1和SHA-2有不同程度的攻击,然而这种攻击方法是否对SHA-3有效,是现在科学界比较关注的问题。SHA-3使用的基本结构是Sponge结构[8],Keccak-f是SHA-3中的非线性置换函数,利用在3维数组上的5步迭代排列构成。Itai Dinur、Orr Dunkelman和Adi Shamir以复杂度2^{24}找到了Keccak-224/256四轮的实际碰撞和五轮的近似碰撞,Keccak-224五轮近似碰撞的汉明距离为5,复杂度为2^{57};Keccak-256五轮近似碰撞的汉明距离为10,复杂度为2^{59}[9]。本文对Keccak-f每一步的差分转移概率进行分析,推出Keccak-f的差分转移概率特性,该结论有助于研究整个SHA-3算法的抗差分攻击强度。该方法有别于通常的差分攻击方法,从置换函数的特性出发,研究SHA-3置换函数满足的一些差分特性,可以把这看作是研究整个SHA-3算法的前期准备,寻找到的差分特性越多,就越容易研究SHA-3的差分特性和抗差分强度等。

2 SHA-3算法

SHA-3算法使用基于Sponge构造的Sponge函数[10],Keccak的状态更新使用的是作用在3维数组上的5步迭代排列。

2.1 Keccak-f置换

Keccak中用到的置换Keccak-f,用Keccak-f[b]表示,其中$b=25\times 2^l$,l表示0到6的数。Keccak-f[b]在$s\in Z_2^b$上,对s的比特流从0到$b-1$进行编号,称b为置换的宽度。SHA-3算法中使用Keccak-f[1600]。

置换Keccak-f[b]被描述在状态a上,a表示一个在$GF(2)$上的三维数组,表示为:$a[5][5][w]$,$w=2^l$。用$x,y\in Z_5$和$z\in Z_w$表示$a[x][y][w]$,(x,y,z)表示比特位置。

到稿日期:2013-03-05 返修日期:2013-07-25 本文受国家自然科学基金项目(61070219),中央高校基本科研业务费专项资金资助课题,北京电子科技学院信息安全重点实验室资助课题资助。

高晓东(1984—),男,硕士,主要研究方向为信息安全;杨亚涛(1978—),男,博士,讲师,主要研究方向为无线通信安全、密码学;李子臣(1965—),男,博士,教授,主要研究方向为信息安全、密码学。

Keccak-f[b]是一个迭代置换，该迭代置换由 n_r 轮 R 运算组成，$R=\iota\circ\chi\circ\pi\circ\rho\circ\theta$。一轮 R 运算一共有 5 步。其中每一步的具体算法表示如下：

θ: $a[x][y][z]\leftarrow a[x][y][z]+\sum_{y'}^{4}a[x-1][y'][z]+\sum_{y'}^{4}a[x+1][y'][z-1]$;

ρ: $a[x][y][z]\leftarrow a[x][y][z-(t+1)(t+2)/2]$,

t 满足 $0\leqslant t<24$ 并且 $\begin{bmatrix}0 & 1\\ 2 & 3\end{bmatrix}^t\begin{bmatrix}1\\ 0\end{bmatrix}=\begin{bmatrix}x\\ y\end{bmatrix}$ 在 $GF(5)^{2\times 2}$ 上，当 $x=y=0$ 时 $t=-1$；

π: $a[x][y]\leftarrow a[x'][y']$，其中 $\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}0 & 1\\ 2 & 3\end{bmatrix}\begin{bmatrix}x'\\ y'\end{bmatrix}$;

χ: $a[x]\leftarrow a[x]+(a[x+1]+1)a[x+2]$;

ι: $a\leftarrow a+RC[i_r]$。

以上各式中，加法和乘法都是基于 $GF(2)$ 域的运算，上面式子中出现的 $RC[i_r]$ 是指循环常数。当 $0\leqslant j\leqslant l$ 时，$RC[i_r][0][0][2^j-1]=rc[j+7i_r]$，其余的值为零。$rc[t]\in GF(2)$ 是一个二元线性反馈移位寄存器的输出。$rc[t]=(x^t \bmod x^8+x^6+x^5+x^4+1) \bmod x$ 在 $GF(2)[x]$ 上。n_r 由置换的宽度决定，$n_r=12+2l$。

2.2 Keccak 海绵函数

Keccak 使用基于海绵构造（Sponge construction）的 Sponge 函数[11]，Sponge 函数表示为 Keccak[r,c]。r、c 表示两个输入参数。在函数中 $\|$ 表示连接符号，$\oplus$ 表示按位异或，$\lfloor s\rfloor_n$ 表示将字符串 s 截短，留前 n 比特。

Keccak[r,c]函数表示如下：

Input $M\in Z_2^*$;
Output $Z\in Z_2^*$
P=pad(P,r)
Let $P=P_0\|P_1\|\cdots\|P_{|P|-1}$, with $P_i\in Z_2^r$
$s=0^{r+c}$
For i=0 to $|P|-1$
{$s=s\oplus(P_i\|0^c)$
  $s=$Keccak-f[r+c](s)
}
Z=''
While out is requested do
{$Z=Z\|\lfloor s\rfloor_r$
  $s=$Keccak-f[r+c](s)
}

Keccak-f 作用在 3 维数组上的 5 步迭代排列，详见上节。Keccak[r,c]函数中使用了两个函数 $pad(M,n)$。$pad(M,n)$ 表示在 M 后添加一比特“1”，紧接着添加最少的“0”，最后再添加一比特“1”，使得添加后的长度是 n 的整数倍。

2.3 Keccak 算法

Keccak 候选算法根据产生摘要的长度分为如下 4 个：

(1)SHA3-224: $\lfloor Keccak[r=1152,c=448]\rfloor_{224}$

(2) SHA3-256: $\lfloor Keccak[r=1088,c=512]\rfloor_{256}$

(3) SHA3-384: $\lfloor Keccak[r=832,c=768]\rfloor_{384}$

(4) SHA3-512: $\lfloor Keccak[r=576,c=1024]\rfloor_{512}$

3 基本运算定义

定义 1 设$(X,+)$，$(Y,+)$是二元域上的有限交换群，$Y=f(X)$，$\alpha\in X$，$\beta\in Y$，则称输入差为 α、输出差为 β 时的差分转移概率[12] $p_f(\alpha\to\beta)$ 为：$\frac{1}{|X|}\#\{x\in X: f(x+\alpha)-f(x)=\beta\}$。

定义 2 $\mathrm{mod}_b(a)$表示 a 模 b 的值，即当 $\mathrm{mod}_b(a)=d$ 时，$d=a \bmod b$。$div_b(a)$表示 a 除以 b 的整数商，即若 $div_b(a)=d$，则 $d=a/b$。

定义 3 设 $X,Y\in Z_2^{5\times 5\times 64}$，$0\leqslant n<1600$，$n\in Z$，$0\leqslant i<5$，$0\leqslant j<5$，$0\leqslant k<64$。定义循环左移 $Y=(X\ll n)$，则 X 与 Y 之间的关系可表示为：$y_{[i][j][k]}=x_{([i][j][k]+n)}=x_{[\alpha][\beta][\gamma]}$，其中 α 和 β 在此满足：$\alpha=\mathrm{mod}_5(i+div_5(j+div_{64}(k+n)))$，$\beta=\mathrm{mod}_5(j+div_{64}(k+n))$，$\gamma=\mathrm{mod}_{64}(k+n)$；定义循环右移 $Y=(X>>n)$，则 X 与 Y 的关系为：$y_{[i][j][k]}=x_{([i][j][k]-n)}$ 即 $y_{([i][j][k]+n)}=y_{[\alpha][\beta][\gamma]}=x_{[i][j][k]}$，其中：$\alpha=\mathrm{mod}_5(i+div_5(j+div_{64}(k+n)))$，$\beta=\mathrm{mod}_5(j+div_{64}(k+n))$，$\gamma=\mathrm{mod}_{64}(k+n)$。

4 Keccak-f 差分转移概率分析

对于 $f:X\to Y$，$x,x',\alpha,\alpha',\beta,\beta'\in Z_2^{5\times 5\times 64}$ 并且 $\beta=f(x)\oplus f(x\oplus\alpha)$，$\alpha'=(\alpha\ll n)$，$\beta'=(\beta\ll n)$，$0\leqslant n<1600$，$n\in Z$，推导 Keccak-f 置换的差分转移概率 $p_f(\alpha'\to\beta')$ 与 $p_f(\alpha\to\beta)$ 之间的关系。Keccak-f 置换函数由 5 步组成，分别是 $\theta,\rho,\pi,\chi,\iota$，下面对每一步分别进行分析。

定理 1 对于 Keccak-f 置换函数的第一步运算 θ，对应的映射为 $f:X\to Y$，则差分转移概率 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$。

证明：对于 Keccak-f 的第一步 θ，$f:X\to Y$，当 $n=1$ 时，$Y=f(X)$对应的布尔函数表达式为：

$$y_{[i][j][k]}=x_{[i][j][k]}\oplus\sum_{j'=0}^{4}x_{[i-1][j'][k]}\oplus\sum_{j'=0}^{4}x_{[i+1][j'][k-1]}$$

其中，$0\leqslant i<5$，$0\leqslant j<5$，$0\leqslant k<64$。输出差 β,β' 的差分变换布尔函数如下：

$$\begin{aligned}\beta_{[i][j][k]}&=x_{[i][j][k]}\oplus\sum_{j'=0}^{4}x_{[i-1][j'][k]}\oplus\sum_{j'=0}^{4}x_{[i+1][j'][k-1]}\oplus\\&\quad(x_{[i][j][k]}\oplus\alpha_{[i][j][k]})\oplus\sum_{j'=0}^{4}(x_{[i-1][j'][k]}\oplus\\&\quad\alpha_{[i-1][j'][k]})\oplus\sum_{j'=0}^{4}(x_{[i+1][j'][k-1]}\oplus\alpha_{[i+1][j'][k-1]})\\&=x_{[i][j][k]}\oplus\sum_{j'=0}^{4}x_{[i-1][j'][k]}\oplus\sum_{j'=0}^{4}x_{[i+1][j'][k-1]}\oplus\\&\quad\alpha_{[i][j][k]}\oplus\sum_{j'=0}^{4}\alpha_{[i-1][j'][k]}\oplus\sum_{j'=0}^{4}\alpha_{[i+1][j'][k-1]}\end{aligned}$$

$$\begin{aligned}\beta'_{[i][j][k]}&=x_{([i][j][k]+1)}\oplus\sum_{j'=0}^{4}x_{([i-1][j'][k]+1)}\oplus\sum_{j'=0}^{4}\\&\quad x_{([i+1][j'][k-1]+1)}\oplus\alpha_{([i][j][k]+1)}\oplus\sum_{j'=0}^{4}\\&\quad\alpha_{([i-1][j'][k]+1)}\oplus\sum_{j'=0}^{4}\alpha_{([i+1][j'][k-1]+1)}\end{aligned}$$

记 $L_\beta^\alpha=\{x\mid f(x)\oplus f(x\oplus\alpha)=\beta\}$，$L_{\beta'}^{\alpha'}=\{x'\mid f(x')\oplus f(x'\oplus\alpha')=\beta'\}$。当 $x\in L_\beta^\alpha$ 时：

$$\begin{aligned}\beta'_{[i][j][k]}&=x_{([i][j][k]+1)}\oplus\sum_{j'=0}^{4}x_{([i-1][j'][k]+1)}\oplus\sum_{j'=0}^{4}\\&\quad([i+1][j'][k-1]+1)\oplus\alpha_{([i][j][k]+1)}\oplus\sum_{j'=0}^{4}\\&\quad\alpha_{([i-1][j'][k]+1)}\oplus\sum_{j'=0}^{4}\alpha_{([i+1][j'][k-1]+1)}\\&=\beta_{([i][j][k]+1)}\end{aligned}$$

当 $\alpha'=(\alpha\ll 1)$，$x'=(x\ll 1)$时，β'与 β 满足 $\beta'=(\beta\ll 1)$，根据上式可以得出：$f(x\ll 1)\oplus f((x\ll 1)\oplus(\alpha\ll 1))=(\beta\ll 1)=\beta'$即满足此式：$f(x')\oplus f(x'\oplus\alpha')=(\beta\ll 1)=\beta'$，因此有 $x'\in L_{\beta'}^{\alpha'}$。记 $x'=\varphi(x)=(x\ll 1)$，其中：$x\in L_\beta^\alpha$，$x'\in L_{\beta'}^{\alpha'}$。下面证明

$x'=\varphi(x)$为一一映射。先证 $x'=\varphi(x)$满足单射。选取两个不同的数 $x_1,x_2\in L_\beta^\alpha$，则我们可以得到 $x_1'=\varphi(x_1)=(x_1\ll 1),x_2'=\varphi(x_2)=(x_2\ll 1)$，可知 $x_1'\neq x_2'$，因此 $x'=\varphi(x)$为单射。

下面证明 $x'=\varphi(x)$为满射。对于任意的 $x'\in L_{\beta'}^{\alpha'}$，存在 $x=(x'>>1)$，只须证明 $x\in L_\beta^\alpha$ 即可。因为 $\beta'=f(x')\oplus f(x'\oplus\alpha')=f(x\ll 1)\oplus f((x\ll 1)\oplus(\alpha\ll 1))$，对应的布尔函数表达式如下式：

$$\begin{aligned}\beta'_{[i][j][k]}=&x_{([i][j][k]+1)}\oplus\sum_{j'=0}^{4}x_{([i-1][j'][k]+1)}\oplus\sum_{j'=0}^{4}x_{([i+1][j'][k-1]+1)}\oplus\alpha_{([i][j][k]+1)}\oplus\sum_{j'=0}^{4}\alpha_{([i-1][j'][k]+1)}\oplus\sum_{j'=0}^{4}\alpha_{([i+1][j'][k-1]+1)}\\=&\beta_{([i][j][k]+1)}\end{aligned}$$

可得：

$$\begin{aligned}\beta'_{([i][j][k]-1)}=&x_{[i][j][k]}\oplus\sum_{j'=0}^{4}x_{[i-1][j'][k]}\oplus\sum_{j'=0}^{4}x_{[i+1][j'][k-1]}\oplus\alpha_{[i][j][k]}\oplus\sum_{j'=0}^{4}\alpha_{[i-1][j'][k]}\oplus\sum_{j'=0}^{4}\alpha_{[i+1][j'][k-1]}\\=&\beta_{[i][j][k]}\end{aligned}$$

因此满足：$f(x)\oplus f(x\oplus\alpha)=\beta$，则 x 满足 $x\in L_\beta^\alpha$。说明对任意的 $x'\in L_{\beta'}^{\alpha'}$，存在 $x\in L_\beta^\alpha$ 使得 $x=(x'\gg 1)$。

根据以上证明可知当 $x\in L_\beta^\alpha,x'\in L_{\beta'}^{\alpha'}$ 时，$x'=\varphi(x)$为一一映射。可以得出 $|L_\beta^\alpha|=|L_{\beta'}^{\alpha'}|$。根据差分转移概率的定义得：$p_f(\alpha\to\beta)=|L_\beta^\alpha|/2^n$，$p_f(\alpha'\to\beta')=|L_{\beta'}^{\alpha'}|/2^n$，因此可以得出：$p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$。

假设 $n=m-1,1\leqslant m<1600,m\in Z$ 时，$p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$成立。当 $n=m$ 时，相当于在 $n=m-1$ 的基础上继续向左循环一位，这种情况跟前面证明的 $n=1$ 时的情形相同，因此 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$成立。

定理 2 对于 Keccak-f 置换函数的第二步运算 ρ，对应的映射为 $f:X\to Y$，则差分转移概率 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$。

证明：对于 Keccak-f 的第二步 ρ，$f:X\to Y$，即 $Y=f(X)$，$y_{[i][j][k]}=x_{[i][j][k-\theta(i,j)]}$，其中 $0\leqslant i<5,0\leqslant j<5,0\leqslant k<64$，$\theta(i,j)$表示与 i,j 有关的函数，它的值是由 i,j 决定的。

输出差 β,β' 的差分变换布尔函数如下：

$$\begin{aligned}\beta_{[i][j][k]}=&x_{[i][j][k-\theta(i,j)]}\oplus(x_{[i][j][k-\theta(i,j)]}\oplus\alpha_{[i][j][k-\theta(i,j)]})\\=&\alpha_{[i][j][k-\theta(i,j)]}\end{aligned}$$

又由 $\beta'=(\beta\ll n)$，我们可以得到 $\beta'_{[i][j][k]}=\alpha_{([i][j][k-\theta(i,j)]+n)}$，因为 $\alpha'=(\alpha\ll n)$，所以可得 $\beta'_{[i][j][k]}=\alpha'_{[i][j][k-\theta(i,j)]}$。即 $\alpha'=(\alpha\ll n)$时，有 $\beta'=(\beta\ll n)$。

差分输出 β,β' 只与对应的差分输入 α,α' 有关，而与 X 无关，因此对于任意的 X，均满足当差分输入为 α,α' 时对应的差分输出为 β,β'，则 $|L_\beta^\alpha|=|L_{\beta'}^{\alpha'}|=2^n$。根据差分转移概率的定义可得对应的差分转移概率为 $p_f(\alpha\to\beta)=|L_\beta^\alpha|/2^n=1$，$p_f(\alpha'\to\beta')=|L_{\beta'}^{\alpha'}|/2^n=1$。因此得出结果 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')=1$。

定理 3 对于 Keccak-f 置换函数的第三步运算 π，对应的映射为 $f:X\to Y$，则差分转移概率 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$。

证明：对于 Keccak-f 的第三步 π，$f:X\to Y$，即 $Y=f(X)$，对应的布尔函数表达式为：

$$y_{[i][j][k]}=x_{[\frac{1}{2}j-\frac{3}{2}i][i][k]}$$

其中，$0\leqslant i<5,0\leqslant j<5,0\leqslant k<64$。输出差 β,β' 的差分变换布尔函数如下：

$$\begin{aligned}\beta_{[i][j][k]}=&x_{[\frac{1}{2}j-\frac{3}{2}i][i][k]}\oplus(x_{[\frac{1}{2}j-\frac{3}{2}i][i][k]}\oplus\alpha_{[\frac{1}{2}j-\frac{3}{2}i][i][k]})\\=&\alpha_{[\frac{1}{2}j-\frac{3}{2}i][i][k]}\end{aligned}$$

由 $\beta'=(\beta\ll n)$得：

$$\beta'_{[i][j][k]}=\alpha_{([\frac{1}{2}j-\frac{3}{2}i][i][k]+n)}$$

由 $\alpha'=(\alpha\ll n)$得：

$$\beta'_{[i][j][k]}=\alpha'_{[\frac{1}{2}j-\frac{3}{2}i][i][k]}$$

$\alpha'=(\alpha\ll n)$时，有 $\beta'=(\beta\ll n)$。差分输出 β,β' 只与对应的差分输入 α,α' 有关，而与 X 无关，因此对于任意的 X，均满足当差分输入为 α,α' 时对应的差分输出为 β,β'，则 $|L_\beta^\alpha|=|L_{\beta'}^{\alpha'}|=2^n$。根据差分转移概率的定义可得对应的差分转移概率为 $p_f(\alpha\to\beta)=|L_\beta^\alpha|/2^n=1$，$p_f(\alpha'\to\beta')=|L_{\beta'}^{\alpha'}|/2^n=1$。因此得出结果 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')=1$。

定理 4 对于 Keccak-f 置换函数的第四步运算 χ，对应的映射为 $f:X\to Y$，则差分转移概率 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$。

证明：对于 Keccak-f 的第四步 χ，$f:X\to Y$，当 $n=1$ 时，$Y=f(X)$，$y_{[i][j][k]}=x_{[i][j][k]}\oplus x_{[i+1][j][k]}x_{[i+2][j][k]}\oplus x_{[i+2][j][k]}$，其中：$0\leqslant i<5,0\leqslant j<5,0\leqslant k<64$。输出差 β,β' 的差分变换布尔函数如下：

$$\begin{aligned}\beta_{[i][j][k]}=&x_{[i][j][k]}\oplus x_{[i+1][j][k]}x_{[i+2][j][k]}\oplus x_{[i+2][j][k]}\oplus(x_{[i][j][k]}\oplus\alpha_{[i][j][k]})\oplus(x_{[i+1][j][k]}\oplus\alpha_{[i+1][j][k]})(x_{[i+2][j][k]}\oplus\alpha_{[i+2][j][k]})\oplus(x_{[i+2][j][k]}\oplus\alpha_{[i+2][j][k]})\\=&x_{[i+1][j][k]}\alpha_{[i+2][j][k]}\oplus x_{[i+2][j][k]}\alpha_{[i+1][j][k]}\oplus\alpha_{[i][j][k]}\oplus\alpha_{[i+1][j][k]}\alpha_{[i+2][j][k]}\oplus\alpha_{[i+2][j][k]}\\\beta'_{[i][j][k]}=&x_{([i+1][j][k]+1)}\alpha_{([i+2][j][k]+1)}\oplus x_{([i+2][j][k]+1)}\alpha_{([i+1][j][k]+1)}\oplus\alpha_{([i][j][k]+1)}\oplus\alpha_{([i+1][j][k]+1)}\alpha_{([i+2][j][k]+1)}\oplus\alpha_{([i+2][j][k]+1)}\end{aligned}$$

记 $L_\beta^\alpha=\{x|f(x)\oplus f(x\oplus\alpha)=\beta\}$，$L_{\beta'}^{\alpha'}=\{x'|f(x')\oplus f(x'\oplus\alpha')=\beta'\}$。当 $x\in L_\beta^\alpha$ 时：

$$\beta'_{[i][j][k]}=x_{([i+1][j][k]+1)}\alpha_{([i+2][j][k]+1)}\oplus x_{([i+2][j][k]+1)}\alpha_{([i+1][j][k]+1)}\oplus\alpha_{([i][j][k]+1)}\oplus\alpha_{([i+1][j][k]+1)}\alpha_{([i+2][j][k]+1)}\oplus\alpha_{([i+2][j][k]+1)}=\beta_{([i][j][k]+1)}$$

$\alpha'=(\alpha\ll 1)$，$x'=(x\ll 1)$时，有 $\beta'=(\beta\ll 1)$，根据上式可得：$f(x\ll 1)\oplus f((x\ll 1)\oplus(\alpha\ll 1))=(\beta\ll 1)=\beta'$，从而可得出：$f(x')\oplus f(x'\oplus\alpha')=\beta'$，因此 $x'\in L_{\beta'}^{\alpha'}$。

记 $x'=\varphi(x)=(x\ll 1)$，其中：$x\in L_\beta^\alpha,x'\in L_{\beta'}^{\alpha'}$，下面证明 $x'=\varphi(x)$为一一映射。先证 $x'=\varphi(x)$满足单射。选取两个不同的数 $x_1,x_2\in L_\beta^\alpha$，则 $x'_1=\varphi(x_1)=(x'_1\ll 1),x_2'=\varphi(x_2)=(x_2\ll 1)$，可知 $x'_1\neq x'_2$，因此 $x'=\varphi(x)$为单射。

下面证明 $x'=\varphi(x)$为满射。对于任意的 $x'\in L_{\beta'}^{\alpha'}$，存在 $x=(x'>>1)$，只须证明 $x\in L_\beta^\alpha$ 即可。由 $\beta'=f(x')\oplus f(x'\oplus\alpha')=f(x\ll 1)\oplus f((x\ll 1)\oplus(\alpha\ll 1))$，得对应的布尔函数表达式如下：

$$\begin{aligned}\beta'_{[i][j][k]}=&x_{([i+1][j][k]+1)}\alpha_{([i+2][j][k]+1)}\oplus x_{([i+2][j][k]+1)}\alpha_{([i+1][j][k]+1)}\oplus\alpha_{([i][j][k]+1)}\oplus\alpha_{([i+1][j][k]+1)}\alpha_{([i+2][j][k]+1)}\oplus\alpha_{([i+2][j][k]+1)}\\=&\beta_{([i][j][k]+1)}\end{aligned}$$

可得：

$$\begin{aligned}\beta'_{([i][j][k]-1)}=&x_{[i+1][j][k]}\alpha_{[i+2][j][k]}\oplus x_{[i+2][j][k]}\alpha_{[i+1][j][k]}\oplus\alpha_{[i][j][k]}\oplus\alpha_{[i+1][j][k]}\alpha_{[i+2][j][k]}\oplus\alpha_{[i+2][j][k]}\end{aligned}$$

因此满足：$f(x)\oplus f(x\oplus\alpha)=\beta$，则 x 满足 $x\in L_\beta^a$。说明对任意的 $x'\in L_\beta^{a'}$，存在 $x\in L_\beta^a$ 使得 $x=(x'>>1)$。

根据以上证明可知当 $x\in L_\beta^a$，$x'\in L_\beta^{a'}$ 时，$x'=\varphi(x)$ 为一一映射，则 $|L_\beta^a|=|L_\beta^{a'}|$。根据差分转移概率的定义得 $p_f(\alpha\to\beta)=|L_\beta^a|/2^n$，$p_f(\alpha'\to\beta')=|L_\beta^{a'}|/2^n$，因此可以得出：$p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$。

假设 $n=m-1$，$1\leqslant m<1600$，$m\in Z$ 时，$p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$成立。当 $n=m$ 时，相当于在 $n=m-1$ 的基础上继续向左循环一位，这种情况跟前面证明的 $n=1$ 时的情形相同，因此 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$成立。

定理 5 对于 Keccak-f 置换函数的第五步运算 ι，对应的映射为 $f:X\to Y$，则差分转移概率 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$。

证明：对于 Keccak-f 的第五步 ι，$f:X\to Y$，即 $Y=f(X)$，$y_{[i][j][k]}=x_{[i][j][k]}\oplus r_{[i][j][k]}$，其中 $0\leqslant i<5$，$0\leqslant j<5$，$0\leqslant k<64$，$r_{[i][j][k]}$ 为常数。输出差 β,β' 的差分变换布尔函数如下：

$$\begin{aligned}\beta_{[i][j][k]}&=x_{[i][j][k]}\oplus r_{[i][j][k]}\oplus((x_{[i][j][k]}\oplus\alpha_{[i][j][k]})\oplus r_{[i][j][k]})\\&=\alpha_{[i][j][k]}\end{aligned}$$

由 $\beta'=(\beta\ll n)$ 得 $\beta'_{[i][j][k]}=\alpha_{([i][j][k]+n)}$；因为 $\alpha'=(\alpha\ll n)$，所以 $\beta'_{[i][j][k]}=\alpha'_{[i][j][k-\theta(i,j)]}$。即若 $\alpha'=(\alpha\ll n)$ 时，有 $\beta'=(\beta\ll n)$。

差分输出 β,β' 只与对应的差分输入 α,α' 有关，而与 X 无关，因此对于任意的 X，均满足当差分输入为 α,α' 时对应的差分输出为 β,β'，则我们可以得出 $|L_\beta^a|=|L_\beta^{a'}|=2^n$。根据差分转移概率的定义可得对应的差分转移概率为 $p_f(\alpha\to\beta)=|L_\beta^a|/2^n=1$，$p_f(\alpha'\to\beta')=|L_\beta^{a'}|/2^n=1$。因此得出结果 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')=1$。

定理 6 如果 Keccak-f 每一步的两个输入差分之间和对应的两个输出差分之间均满足相同循环移位特性，整个 Keccak 算法的置换函数 Keccak-f 对于输入输出差分通过循环移位后，其差分转移概率 $p_f(\alpha\to\beta)=p_f(\alpha'\to\beta')$。

证明：根据之前的证明，Keccak 算法的置换函数 Keccak-f 每一步运算的输入和对应的输出差分均循环移位相同位数后的循环移位概率与循环移位之前的差分转移概率相同。只要 Keccak-f 每一步的两个输入差分之间和对应的两个输出差分之间均满足相同循环移位特性，就可以保证 Keccak-f 每一步对输入输出差分进行循环移位后，其差分转移概率不变。置换函数 Keccak-f 的每一步运算的输入是前一步的输出。根据前面证明的结论可以得出，如果 Keccak-f 每一步的两个输入差分之间和对应的两个输出差分之间均满足相同循环移位特性，整个 Keccak 算法的置换函数 Keccak-f 对输入输出差分进行循环移位后，其差分转移概率相同。

结束语 差分分析方法是分析 Hash 函数最有效的方法之一。王小云教授利用差分攻击方法对 MD4、MD5、SHA、SHA-1 进行了有效的分析，找到了这些算法的多对碰撞。今年十月，NIST 公布了最新的 Hash 函数标准，Keccak 算法将作为今后一段时间的 Hash 函数标准算法 SHA-3。Keccak 算法将成为研究热点。对以往 Hash 函数的分析结果有助于产生更加安全的 Hash，如同 SHA-3 算法的产生过程一样，通过对提交的候选算法不断分析淘汰和改进而产生。本文在研究 SHA-3 算法的基础上，提出了三维数组的循环移位方法。根据 Keccak 置换函数结构特点，构造出每一步运算的输出差分布尔函数表达式，对每一步运算进行分析研究，证明了对 SHA-3 非线性置换函数的每一步的输入输出差分进行循环移位后，其差分转移概率不变的结论。最后推出 SHA-3 置换函数的每一步变换的两个输入差分和对应的两个输出差分之间均满足相同循环移位特性时，整个置换函数的输入输出差分通过循环移位后，其差分转移概率相同。该结论有助于研究整个 SHA-3 算法的抗差分攻击强度。

参 考 文 献

[1] NIST Tech Beat. NIST Selects Winner of Secure Hash Algorithm(SHA-3) Competition[EB/OL]. http://www.nist.gov/itl/csd/sha-100212.cfm,2012-10-02

[2] Bertoni G,Daemen J,Peeters M,et al. The Keccak SHA-3 submission[EB/OL]. http:// keccak.noekeon.org/,2011-01-14

[3] NIST. Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA-3) family[J]. Federal RegisterNotices,2007,72(212):62212-62220

[4] Andrew R,Ray P,Chang S J. Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition[R]. Information Technology Laboratory National Institute of Standards and Technology,Gaithersburg,2009

[5] Meltem S T,Ray P,Lawrence E B,et al. Status Report on the Second Round of the SHA-3 Cryptographic Hash Algorithm Competition. Computer Security Division[R]. Information Technology Laboratory National Institute of Standards and Technology,Gaither-sburg,2011

[6] NIST. The SHA-3 Finalists candidates U S department of commerce national information service[EB/OL]. http://csrc.nist.gov./groups/ST/hash/sha-3/Round3/submissions_round3.html,2011

[7] Wang X Y, Yin Yi-qun, Yu H B. Finding collisions in the full SHA-1[C]// Shoup(ed). CRYPTO 2005,LNCS 3621. Berlin: Springer-Verlag,2005:17-36

[8] Bertoni G,Daemen J,Peeters M,et al. The Keccak sponge function family[EB/OL]. http:// keccak.noekeon.org/news.Html,2012-04

[9] Dinur I,Dunkelman O,Shamir A. New Attacks on Keccak-224 and Keccak-256[C]// International Association for Cryptologic Research 2012. FSE 2012,LNCS 7549. 2012:42-461

[10] Maurer U,Renner R,Holenstein C. Indifferentiability,impossibility results on reductions,and applications to the random oracle methodology[C]// Naor,ed. TCC'04,LNCS 2951. Berlin: Springer-Verlag,2004:21-39

[11] Bertoni G, Daemen J, Peeters M, et al. Cryptographic sponge functions [EB/OL]. http://sponge.noekeon.org/,2011-01

[12] 李倩男,李云强,蒋淑静,等. Keccak 类非线性变换的差分性质研究[J]. 通信学报,2012,33(9):140-146