一种基于完全性的不可能差分区分器构造方法

李俊志* 关 杰
(信息工程大学电子技术学院 郑州 450001)

摘 要：基于混合运算的密码算法(MOC)以安全性高、软硬件实现效率高等特点受到人们的广泛关注。完全性指输出的每一比特都包含有输入每一比特的信息，达到完全性是密码算法设计的一个基本原则。该文提出针对 MOC 算法完全性分析的通用算法，并在此基础上提出利用完全性寻找 MOC 算法的不可能差分区分器的方法，此构造方法可直接给出 MOC 算法高重量的不可能差分区分器且搜索效率高，为 MOC 算法不可能差分区分器的实际构造提供了理论指导和技术支持。应用此方法找到了 SIMON 系列算法全部现有的最长不可能差分区分器，并找到了 SPECK 系列算法更多的不可能差分区分器。

关键词：基于混合运算密码算法；不可能差分区分器；完全性；SIMON；SPECK
中图分类号： TN918.1 **文献标识码**： A **文章编号**：1009-5896(2018)02-0430-08
DOI: 10.11999/JEIT170422

A Method of Constructing Impossible Differential Distinguishers Based on Completeness

LI Junzhi GUAN Jie
(*College of Electronic Technology*, *Information Engineering University*, *Zhengzhou* 450001, *China*)

Abstract: Mixed Operation based Ciphers (MOC) attract cryptographers owing to their high security and high efficiency on both software and hardware platforms. As a basic principle of cryptosystem design, completeness refers to that every output bit contains the information of every input bit. This paper presents a universal algorithm of completeness analysis against MOC. Based on the algorithm, a method of constructing impossible differential distinguishers utilizing completeness is proposed. This method constructs heavy weight impossible differential distinguishers directly with high efficiency. The method can provide theory and technology direction for the construction of impossible differential distinguishers. Then, this paper analysis SIMON and SPECK with this method and introduces all the longest impossible differential distinguishers of SIMON currently public and new impossible differential distinguishers of SPECK.

Key words: Mixed Operation based Cipher (MOC); Impossible differential distinguishers; Completeness; SIMON; SPECK

1 引言

不可能差分分析[1]由 Biham, Biryukov 和 Shamir 于 1999 年首次提出，并应用于 Skipjack[2] 算法的分析中，同时作者给出了不可能差分攻击的方法和原理，随后不可能差分分析作为一种重要的密码分析方法广泛应用于大量的分组密码算法中，如 AES[3], Camellia[4], MIBS[5], 3D[6]等，并取得了较好的分析效果。

不可能差分分析的基本思想是利用概率为 0 的差分来排除错误的候选密钥，恢复正确的密钥。目前寻找不可能差分路径的主要思想为中间相错，即从加密方向和解密方向寻找概率为 1 的差分传递链，如果中间存在矛盾则将前后连接起来即得到一条不可能差分传递链。

基于混合运算的密码算法[7](Mixed Operation based Cipher, MOC)是一类仅基于简单运算(模加运算、循环移位运算、异或加运算、移位运算、与、或、非等比特布尔运算)的算法，它们以安全性高、软硬件实现效率高等特点受到人们的广泛关注，其典型算法为 HIGHT[8], SIMON[9], SPECK[9]等。目前对此类算法的安全性研究是密码领域的热点方向，分析方法较多，其中不可能差分分析是一种比较有效的方法。

目前对基于 MOC 算法的不可能差分构造方法主要有：文献[10]中提出的利用算法弱扩散性构造不

收稿日期：2017-05-08；改回日期：2017-09-26；网络出版：2017-11-01
*通信作者：李俊志 lijunzhi1998@163.com
基金项目：国家自然科学基金(61572516, 61272041, 61272488)
Foundation Items: The National Natural Science Foundation of China (61572516, 61272041, 61272488)

可能差分的方法，该方法找到的不可能差分个数较少且有时找到的区分器轮数较小；文献[7]中提出的模式搜索算法，主要思想是搜索单比特的不可能差分，再组合成高重量的不可能差分区分器并验证其正确性，其组合及验证过程繁琐并容易出现遗漏和错误；文献[11,12]中将搜索差分的问题转化为混合整数线性规划(MILP)问题进行求解的自动搜索方法，该方法在进行求解时会耗费大量时间，特别是当算法分组规模较大时进行求解的代价巨大。

完全性指经过若干轮迭代后，输出的每一比特都包含有输入的每一比特的信息，它是分组密码安全性的一个基本要求，当算法没有达到完全性时可以构造概率为 1 的差分链。文献[13]提出一种面向非线性移存器型序列密码算法的比特级完全性通用算法，该通用算法对算法内部状态信息进行细致的刻画进而从理论上给出算法的完全性分析。

本文扩展了文献[13]的完全性通用算法，提出了针对 MOC 的完全性通用算法，并在此基础上给出了利用完全性寻找 MOC 的不可能差分区分器的新方法，可直接给出算法高重量的不可能差分区分器且搜索效率高，为不可能差分区分器的实际构造提供了理论指导和技术支持。应用此方法找到了 SIMON 算法全部现有的最长不可能差分区分器，并找到了 SPECK 算法更多的不可能差分区分器，此方法提高了搜索不可能差分区分器的效率，但是没有增加对这些算法不可能差分攻击的轮数，对它们的安全性并不构成威胁。

2　针对 MOC 模型的完全性通用算法

本文在文献[13]的面向非线性移存器型序列密码的完全性通用算法的基础上，给出模 2^n 加、模 2^n 减运算的完全性运算规则，提出了针对包含异或、逻辑与、移位及循环移位、模 2^n 加、模 2^n 减等运算的 MOC 算法的完全性通用算法。

完全性[14]指输出的每一比特都包含有输入每一比特的信息，即输出的每一比特都应该是关于明文和密钥的函数，且应使该函数足够复杂才能保证算法的安全性。

完全性通用算法的主要思想是将算法中间状态和输出比特看成是关于密钥及明文的表达式，对线性部分给出具体的表达式，对非线性部分给出所包含信息的集合，当某比特的非线性部分包含所有密钥及明文(IV)的信息时认为该比特完全。本文主要研究了基于 MOC 的分组密码，故以下只给出了关于密钥及明文的完全性通用算法。

设密码算法密钥长度 m 比特，明文长度 l 比特，记密钥比特为 $k_t(0\le t<m)$，明文比特为 $p_t(0\le t<l)$，第 i 轮的第 j 比特内部状态为 s_j^i，令 $s_j^i=f_{ij}(K,P)$，令 f_{ij} 的线性部分和非线性部分分别为 $L\left(f_{ij}\right)$ 和 $N\left(f_{ij}\right)$，将内部状态表示为

$$\left(L\left(s_j^i\right),N\left(s_j^i\right)\right)=\left(\bigcup_{k_t\in L(f_{ij})}k_t\cup\bigcup_{p_t\in L(f_{ij})}p_t,\ \bigcup_{k_t\in N(f_{ij})}k_t\cup\bigcup_{p_t\in N(f_{ij})}p_t\right)\quad(1)$$

其中，$L\left(s_j^i\right)$ 表示内部状态 s_j^i 的线性部分所包含的明文和密钥比特所组成的集合，$N\left(s_j^i\right)$ 表示 s_j^i 的非线性部分所包含的明文和密钥比特所组成的集合，我们称集合 $\left(L\left(s_j^i\right),N\left(s_j^i\right)\right)$ 为 s_j^i 的完全性表达式。

2.1 对 MOC 运算的完全性运算法则

模 2 加和逻辑与是两个基本运算，对这两个运算的完全性法则，本文直接采用文献[13]中的完全性运算法则。移位和循环移位的完全性运算法则比较简单，直接将内部状态的线性部分和非线性部分作移位即可。其它较复杂的运算的完全性法则均建立在模 2 加和逻辑与这两个运算完全性法则的基础上。下面引用文献[13]中的模 2 加和逻辑与的运算法则，然后主要给出模 2^n 加、模 2^n 减的完全性运算法则。

(1)模 2 加和逻辑与：　模 2 加的完全性运算如表 1 所示，逻辑与的完全性运算如表 2 所示。

表 1 模 2 加的完全性运算

算法 1　模 2 加的完全性运算[13]
输入：$s_1=(L(s_1),N(s_1)),\ s_2=(L(s_2),N(s_2))$
输出：$s_3=s_1\oplus s_2=(L(s_3),N(s_3))$
步骤 1　将 $L(s_1)$ 和 $L(s_2)$ 看成是线性多项式，进行多项式的模 2 加，得到的结果即为 $L(s_3)$；
步骤 2　$N(s_3)=N(s_1)\cup N(s_2)$，输出 $s_3=(L(s_3),N(s_3))$。

表 2 逻辑与的完全性运算

算法 2　逻辑与的完全性运算[13]
输入：$s_1=(L(s_1),N(s_1)),\ s_2=(L(s_2),N(s_2))$
输出：$s_3=s_1\ \&\ s_2=(L(s_3),N(s_3))$
步骤 1　若 s_1 和 s_2 中有一个为常数，则进入步骤 2，否则进入步骤 3；
步骤 2　若 s_1 或 s_2 为常数 0，则输出 $s_3=\varnothing\,|\,\varnothing$ 并退出算法；若 s_1 为常数 1，则输出 $s_3=(L(s_2),N(s_2))$ 并退出算法；若 s_1 为常数 1，则输出 $s_3=(L(s_1),N(s_1))$ 并退出算法；
步骤 3　将 $L(s_1)$ 和 $L(s_2)$ 看成是线性多项式，进行多项式的相乘，得到的结果的线性部分即为 $L(s_3)$，非线性部分为 $N_1(s_3)$；
步骤 4　$N(s_3)=N(s_1)\cup N(s_2)\cup N_1(s_3)$，输出 $s_3=(L(s_3),N(s_3))$。

(2)模 2^n 加(减)： 模 2^n 加(减)可分解为比特级的运算。令，$A=a_{n-1}a_{n-2}\cdots\ a_0$, $B=b_{n-1}b_{n-2}\cdots\ b_0$, c_i 为第 i 位的进位 $(0\le i<n)$，若记

$$D=A\boxplus B=d_{n-1}d_{n-2}\cdots\ d_0$$

$$(D=A\boxminus B=d_{n-1}d_{n-2}\cdots\ d_0) \qquad (2)$$

则模 2^n 加运算的逐比特数学表达式如下：对于 $0\le i<n$，有

$d_i=a_i\oplus b_i\oplus c_i$; $c_0=0$

$c_{i+1}=a_i\ \&\ b_i\oplus a_i\ \&\ c_i\oplus b_i\ \&\ c_i$

而模 2^n 减运算的逐比特数学表达式如下：对于 $0\le i<n$，有

$d_i=a_i\oplus b_i\oplus c_i$; $c_0=0$

$c_{i+1}=(a_i\oplus 1)\ \&\ b_i\oplus(a_i\oplus 1)\ \&\ c_i\oplus b_i\ \&\ c_i$

根据上述转化关系，可以得出模 2^n 加(减)的完全性运算法则如表 3 和表 4 所示。

以上几个运算涵盖了MOC的大部分基础运算，由它们可以组成具体算法的完全性推演规则。

2.2 MOC 模型的完全性通用算法描述

分组密码一般由加脱密算法和子密钥扩展算法构成，下面给出 MOC 类分组密码的完全性通用算法如表 5 所示。

算法 5 的若干性质。

当算法输出的某个比特关于明文不完全时，即非线性部不包含某些明文比特时，则可按照下述方式构造概率为 1 的差分传递链。

对某个输出比特 y_i，记 $y_i=(L(y_i),N(y_i))$，可根据其完全性表达式式(1)考察输入 x 的情况。

情况 1：输入 $x_j\notin N(y_i)$ 且 $x_j\in L(y_i)$。

情况 2：输入 $x_j\notin N(y_i)$ 且 $x_j\notin L(y_i)$。

易知，若 x_j 满足情况 2，无论 x_j 的差分 0 或者 1，y_i 的差分均为 0；

当有偶数个满足情况 1 的 x_j 的差分为 1，其它输入的差分为 0 时，y_i 的差分为 0；

当有奇数个满足情况 1 的 x_j 的差分为 1，其它输入的差分为 0 时，y_i 的差分为 1。

为了方便描述，给出如下符号说明。

对输出的第 i 个比特 y_i，令

$A_0^i=\{\boldsymbol{X}=(x_0,x_1,\cdots,x_{n-1})|x_j=1$，当 x_j 满足情况 1；$x_k=0$，当 x_k 不满足情况 1；且 $W(\boldsymbol{X})$ 为偶数$\}$，其中 $W(\boldsymbol{X})$ 表示 X 的汉明重量(下同)；

$A_1^i=\{\boldsymbol{X}=(x_0,x_1,\cdots,x_{n-1})|x_j=1$，当 x_j 满足情况 1；$x_k=0$，当 x_k 不满足情况 1；且 $W(\boldsymbol{X})$ 为奇数$\}$；

$B^i=\{\boldsymbol{X}=(x_0,x_1,\cdots,x_{n-1})|x_j$ 任取，当 x_j 满足情况 2；$x_k=0$，当 x_k 不满足情况 2$\}$；

$D_0^i=\{\alpha\oplus\beta\mid\alpha\in A_0^i,\beta\in B^i\}$；

$D_1^i=\{\alpha\oplus\beta\mid\alpha\in A_1^i,\beta\in B^i\}$。

根据以上定义可以给出下面两种类型的概率为 1 的差分对应：

对于输入差 $\alpha\in D_0^i\bigcup B^i$，有以下形式的差分对应：

表 3 模 2^n 加的完全性运算

算法 3 模 2^n 加的完全性运算

输入：$A=a_{n-1}a_{n-2}\cdots\ a_0$, $B=b_{n-1}b_{n-2}\cdots\ b_0$, $a_i=(L(a_i),N(a_i))$, $b_i=(L(b_i),N(b_i)),(0\le i<n)$

输出：$D=A\boxplus B=d_{n-1}d_{n-2}\cdots\ d_0$

令 $c_0=(\varnothing,\varnothing)$；

For $i=0$ to $n-1$ do

(1)调用算法 1 计算 $d_i=a_i\oplus b_i\oplus c_i$；

(2)调用算法 1 和算法 2 计算 $c_{i+1}=a_i\ \&\ b_i\oplus a_i\ \&\ c_i\oplus b_i\ \&c_i$；

输出 $D=d_{n-1}d_{n-2}\cdots\ d_0$。

表 4 模 2^n 减的完全性运算

算法 4 模 2^n 减的完全性运算

输入：$A=a_{n-1}a_{n-2}\cdots\ a_0$, $B=b_{n-1}b_{n-2}\cdots\ b_0$, $a_i=(L(a_i),N(a_i))$, $b_i=(L(b_i),N(b_i)),(0\le i<n)$

输出：$D=A\boxplus B=d_{n-1}d_{n-2}\cdots\ d_0$

令 $c_0=(\varnothing,\varnothing)$；

For $i=0$ to $n-1$ do

(1)调用算法 1 计算 $d_i=a_i\oplus b_i\oplus c_i$；

(2)调用算法 1 和算法 2 计算 $c_{i+1}=(a_i\oplus 1)\ \&\ b_i\oplus(a_i\oplus 1)\ \&c_i\oplus b_i\ \&\ c_i$；

输出 $D=d_{n-1}d_{n-2}\cdots\ d_0$。

表 5 MOC 类分组密码的完全性能用算法

算法 5 MOC 型分组密码算法的完全性通用算法

输入：算法参数，加密轮数 R，密钥，明文

输出：R 轮输出各比特的线性部分和非线性部分

For $i=1$ to R

步骤 1 将子密钥扩展算法中的输入 k_i 用其完全性表示 $(k_i,\varnothing)(0\le i<m)$ 代替，将密钥扩展算法中的运算(异或、逻辑与、移位及循环移位、模 2^n 加、模 2^n 减)用相应的完全性运算代替，运行子密钥扩展算法并得到 R 轮子密钥的各比特的线性部分和非线性部分 $\left(L\left(sk_j^i\right),N\left(sk_j^i\right)\right)(1\le i\le R,0\le j<M)$，其中 M 为每轮输出的子密钥长度；

步骤 2 将加密算法中的输入 p_i 用其完全性表示 $(p_i,\varnothing)$ $(0\le i<l)$ 代替，将加密算法中的运算(异或、逻辑与、移位及循环移位、模 2^n 加、模 2^n 减)用相应的完全性运算代替，将 $\left(L\left(sk_j^i\right),N\left(sk_j^i\right)\right)(1\le i\le R,0\le i<M)$ 代入，运算并得到 R 轮输出各比特的线性部分和非线性部分 $\left(L\left(s_j^i\right),N\left(s_j^i\right)\right)(1\le i\le R,0\le j<l)$；

输出 $\left(L\left(s_j^i\right),N\left(s_j^i\right)\right)(1\le i\le R,0\le j<l)$。

差分类型 I： $\alpha \to \mathbf{0}_i$

其中，$\mathbf{0}_i$ 表示输出差分在第 i 位差分为 0，其它位置的差分暂时不能确定。

对于输入差 $\beta \in D_1^i$，有以下形式的差分对应：

差分类型 II： $\beta \to \mathbf{1}_i$

其中，$\mathbf{1}_i$ 表示输出差分在第 i 位差分为 1，其它位置的差分暂时不能确定。

下一节将利用这两类差分传递链构造不可能差分区分器。

3　利用完全性构造 MOC 算法的不可能差分区分器

利用完全性构造不可能差分区分器的基本思想是中间相错，分别从加密方向和解密方向运行完全性通用算法，在中间相遇的位置对每个比特进行分析，若根据某一比特 y_i 的完全性，从加密方向上可以概率 1 构造 I 型(II 型)差分特征；而在解密方向可以概率 1 构造 II 型(I 型)差分特征，将这两个特征链接起来，在 y_i 这一比特必然存在矛盾，则选取合适的输入差分和输出差分后，将构造出一个不可能差分区分器，其形式如图 1 所示。

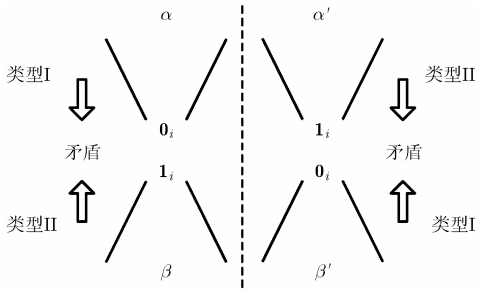

图 1　MOC模型不可能差分区分器的构造

下面给出非相关密钥条件下，即密钥不改变(差分为 0)时利用完全性构造不可能差分区分器的算法。

3.1 基本构造方法

MOC 算法的不可能差分区分器的基本构造算法利用完全性提供的概率为 1 的差分传递链构造单比特矛盾，并可以根据算法完全性特点给出算法的高重量(汉明重量大于 1)不可能差分区分器(如图 1 所示类型 I 和类型 II)。

算法 6(见表 6)的最大复杂度为 $O\left(R^2N^3\right)$，其中 R 为算法轮数，N 为算法分组长度，该搜索算法为多项式时间算法，该算法可以给出 MOC 类算法不可差分区分器的具体形式及不可能差分区分器轮数的选取准则，为不可能差分区分器的构造提供了理论依据。

表 6 利用完全性构造 MOC 不可能差分区分器算法

算法 6　利用完全性构造 MOC 不可能差分区分器算法

输入：MOC 型分组密码算法

输出：r 轮不可能差分区分器的集合 A

初始化：$A=\varnothing$

步骤 1　从加密方向运行算法 5*，直到每个输出比特的非线性部分均包含所有的信息，设此时算法运行 r_1 轮，运行时存储每轮输出的完全性 $s_j^i=\left(L\left(s_j^i\right),N\left(s_j^i\right)\right)$，$(0\le i<r_1,0\le j<l)$；

步骤 2　从解密方向运行算法 5*，直到每个输出比特的非线性部分均包含所有的信息，设此时算法运行 r_2 轮，运行时存储每轮输出的完全性 $s_j^{'i}=\left(L\left(s_j^{'i}\right),N\left(s_j^{'i}\right)\right)$，$\left(0\le i<r_2,0\le j<l\right)$；

步骤 3　For $t=r_1+r_2-2$ to 1 do

For $a=1$ to r_1-1 do

$b=t-a$;

If $b\ge r_2$, continue;

For $j=0$ to $l-1$ do

根据 s_j^a 的完全性分别构造 B^a, D_0^a 和 D_1^a；

根据 $s_j^{'a}$ 的完全性分别构造 $B^{'b}$,$D^{'b}_0$ 和 $D^{'b}_1$；

For 所有 $\alpha\in D_0^a\cup B^a$ do

对所有 $\beta\in D_1^{'b}$，令 A=A∪{$\alpha\to\beta$}

For 所有 $\alpha\in D_1^a$ do

对所有 $\beta\in D_0^{'b}\cup B^{'b}$，令 A=$A\cup\{\alpha\to\beta\}$

If $A\ne\varnothing$，令 $r=t$，输出不可能差分区分器的集合 A 及轮数 r 并终止算法。

If $A=\varnothing$，输出未找到不可能差分并终止算法。

*算法 5*是指在运行算法 5 时不进行步骤 1 及步骤 2 中的子密钥参与环节，这里由于此时密钥差分为 0，如果若考虑相关密钥不可能差分区分器的构造，则运行算法 5 即可

3.2 扩展轮数的区分器构造方法

为了增加不可能差分区分器的轮数，我们在算法 6 构造的不可能差分区分器的基础上，以概率 1 向前或向后链接若干轮差分特征，从而找到尽可能长的不可能差分区分器。构造的关键问题是，如何找到以概率 1 成立，且满足原不可能差分对应的结构特点的差分对应。我们依然可以用完全性通用算法来寻找这样的差分对应。

为了对算法 6 进行扩展，我们在运行完全性通用算法时将和算法 6 有所区别。我们将采用逆向搜索技术，即向前链接时，将运行解密完全性通用算法；向后链接时，将运行加密完全性通用算法。这样可保证链接成功后的不可能差分特征一定不包含在算法 6 中。

对某个 $n$bit 输入 $n$bit 输出的可逆函数

$$F:(y_0,y_1,\cdots,y_{n-1})=F(x_0,x_1,\cdots,x_{n-1}) \tag{3}$$

其中，$y_i=(L(y_i),N(y_i))$。

函数 F 的逆函数为

$$\widetilde{F}:(x_0,x_1,\cdots,x_{n-1})=\widetilde{F}(y_0,y_1,\cdots,y_{n-1}) \tag{4}$$

将联合考察输出 $n$bit 的完全性，考虑如下情况：

情况 3：输入 x_j在所有 y_i的非线性部分均不出现，即 $x_j \notin \bigcup_{0\le i<n} N(y_i)$。则有以下结论：

结论 1 对于可逆函数 $F,(y_0,y_1,\cdots,y_{n-1})=F(x_0,x_1,\cdots,x_{n-1})$，若 $\overline{\bigcup_{0\le i<n} N(y_i)}\neq\varnothing$，不妨令 $x_j\notin\bigcup_{0\le i<n} N(y_i)$，$A_{x_j}=\bigcup_{x_j\in L(y_i)} y_i$，记 $\boldsymbol{e}_i$ 为 n 维单位向量则

$$p_F\left\{\left(\boldsymbol{e}_j\right)\to\left(\mathop{\oplus}_{y_i\in A_{x_j}}\boldsymbol{e}_i\right)\middle| x_j\notin\bigcup_{0\le i<n}N(y_i)\right\}=1$$

且

$$p_{\widetilde{F}}\left\{\left(\mathop{\oplus}_{y_i\in A_{x_j}}\boldsymbol{e}_i\right)\to\left(\boldsymbol{e}_j\right)\middle| x_j\notin\bigcup_{0\le i<n}N(y_i)\right\}=1$$

证明 对于 F 函数，由于 $x_j\notin\bigcup_{0\le i<n}N(y_i)$，因此对于任一 y_i $(0\le i<n)$，x_j 一定满足情况 1 或情况 2 其中之一，而 $A_{x_j}=\bigcup_{x_j\in L(y_i)} y_i$ 即为使 x_j 满足情况 1 的 y_i 组成的集合，故当输入差分为 $\boldsymbol{e}_j$ 时，以概率 1 发生如下事件，$y_i\in A_{x_j}$ 的差分为 1，$y_i\notin A_{x_j}$ 的差分为 0，则对于 $x_j\notin\bigcup_{0\le i<n}N(y_i)$，对任意 $\boldsymbol{X}\in\{0,1\}^n$，一定有

$$F(\boldsymbol{X})\oplus F\left(\boldsymbol{X}\oplus\left(\boldsymbol{e}_j\right)\right)=\left(\mathop{\oplus}_{y_i\in A_{x_j}}\boldsymbol{e}_i\right) \tag{5}$$

即

$$p_F\left\{\left(\boldsymbol{e}_j\right)\to\left(\mathop{\oplus}_{y_i\in A_{x_j}}\boldsymbol{e}_i\right)\middle| x_j\notin\bigcup_{0\le i<n}N(y_i)\right\}=1 \tag{6}$$

可逆函数 $\widetilde{F}$ 作用到式(5)，可得

$$\widetilde{F}\left(F(\boldsymbol{X})\oplus\left(\mathop{\oplus}_{y_i\in A_{x_j}}\boldsymbol{e}_i\right)\right)=\widetilde{F}\left(F\left(\boldsymbol{X}\oplus\left(\boldsymbol{e}_j\right)\right)\right)=\boldsymbol{X}\oplus(\boldsymbol{e}_j) \tag{7}$$

令 $\boldsymbol{Y}=F(\boldsymbol{X})$，有

$$\widetilde{F}\left(\boldsymbol{Y}\oplus\left(\mathop{\oplus}_{y_i\in A_{x_j}}\boldsymbol{e}_i\right)\right)\oplus\widetilde{F}(\boldsymbol{Y})=\left(\boldsymbol{e}_j\right) \tag{8}$$

由于 F 函数为双射，因此当 $\boldsymbol{X}$ 取遍 $\{0,1\}^n$ 时 $\boldsymbol{Y}$ 也取遍 $\{0,1\}^n$，故

$$p_{\widetilde{F}}\left\{\left(\mathop{\oplus}_{y_i\in A_{x_j}}\boldsymbol{e}_i\right)\to\left(\boldsymbol{e}_j\right)\middle| x_j\notin\bigcup_{0\le i<n}N(y_i)\right\}=1$$

证毕

为了方便描述，给出如下定义。令

$$A=\left\{\left(\boldsymbol{e}_j\right)\to\left(\mathop{\oplus}_{y_i\in A_{x_j}}\boldsymbol{e}_i\right)\middle| x_j\notin\bigcup_{0\le i<n}N(y_i)\right\}$$

$$\widetilde{A}=\{(\beta\to\alpha)\,|\,(\alpha\to\beta)\in A\}$$

记 $|A|=\left|\widetilde{A}\right|=T$，若 $T\ge1$，为了得到高重量的差分链，可将 A 中输入差及相应的输出差做同样的非零线性组合得到如下集合：

$$E=\left\{\left(\mathop{\oplus}_{\boldsymbol{v}\in\{0,1\}^T\text{且}\boldsymbol{v}\neq\boldsymbol{0}}\boldsymbol{v}_t\cdot\alpha_t\right)\to\left(\mathop{\oplus}_{\boldsymbol{v}\in\{0,1\}^T\text{且}\boldsymbol{v}\neq\boldsymbol{0}}\boldsymbol{v}_t\cdot\beta_t\right)\middle|(\alpha_t\to\beta_t)\in A\right\}$$

（$\boldsymbol{v}_t$ 表示 t 维向量 $\boldsymbol{v}$ 的第 t 位）

$$\widetilde{E}=\left\{\left(\mathop{\oplus}_{\boldsymbol{v}\in\{0,1\}^T\text{且}\boldsymbol{v}\neq\boldsymbol{0}}\boldsymbol{v}_t\cdot\alpha_t\right)\to\left(\mathop{\oplus}_{\boldsymbol{v}\in\{0,1\}^T\text{且}\boldsymbol{v}\neq\boldsymbol{0}}\boldsymbol{v}_t\cdot\beta_t\right)\middle|(\alpha_t\to\beta_t)\in\widetilde{A}\right\}$$

由结论 1 易证：E 和 $\widetilde{E}$ 中的差分对应均以概率 1 成立。

这样就可利用完全性通用算法得到加(解)密方向的概率为 1 的差分对应 $(\alpha_2\to\beta_2)$。

若想链接成功，还需要做以下判断：

对于不可能差分区分器 $(\alpha_1\to\beta_1)$，若想使用概率为 1 的差分对应 $(\alpha_2\to\beta_2)$，将不可能差分区分器延长若干轮，则可以采用向前链接和向后链接的方式(如图 2 所示)。

向前链接时，若 β_2 满足 α_1 的结构特点，则存在不可能差分区分器 $(\alpha_2\to\beta_1)$；

向后链接时，若 α_2 满足 β_1 的结构特点，则存在不可能差分器 $(\alpha_1\to\beta_2)$。

为了方便描述，有以下定义。

定义 1 若两个输入(输出)差分 $\alpha_1=(x_1,x_2,\cdots,x_n)$ 和 $\alpha_2=\left(x_1',x_2',\cdots,x_n'\right)$，其中 $x_j,x_j'\in\{0,1,*\}$，*是指取值任意，所有的 x_j 均满足以下 3 种条件之一时：

条件 1：若 $x_j'=0$，则 $x_j=0$；

条件 2：若 $x_j'=1$，则 $x_j=1$；

条件 3：若 x_j' 为*，则 x_j=0 或 1 或*；

则称 α_1 包含于 α_2，记为 $\alpha_1\prec\alpha_2$。

则不可能差分区分器的链接过程如图 2 所示。

下面给出利用完全性扩展 MOC 算法不可能差分区分器轮数的方法如表 7 所示。

算法 7 的最大复杂度为 $O\left(R^2|A|^3\right)$，其中 R 为算法轮数，$|A|$ 为集合 A 的元素个数。此处给出的为最大复杂度，实际搜索时由于很多路径不存在，因此搜索复杂度往往远小于最大值。

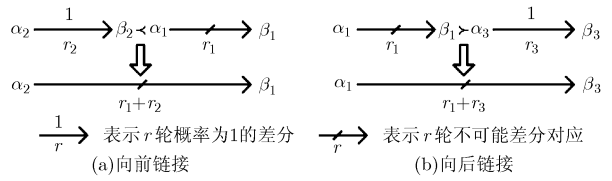

图2　MOC模型不可能差分区分器的扩展

表 7 利用完全性构造扩展的不可能差分区分器

算法 7　利用完全性构造扩展的不可能差分区分器

输入：分组密码算法，r 轮不可能差分区分器的集合 A

输出：扩展后 r' 轮不可能差分传递链的集合 A'

初始化：$A'=\varnothing$

步骤 1　从加密方向运行算法 5*，直到每个输入比特均不出现情况 3，设此时算法运行 r_3 轮，运行时存储每轮输出的完全性 $s_j^i=\left(L\left(s_j^i\right),N\left(s_j^i\right)\right),\ (0\le i<r_3,0\le j<l)$；

步骤 2　从解密方向运行算法 5*，直到每个输入比特均不出现情况 3，设此时算法运行 r_4 轮，运行时存储每轮输出的完全性 $s_j^{'i}=\left(L\left(s_j^{'i}\right),N\left(s_j^{'i}\right)\right),\ (0\le i<r_4,0\le j<l)$；

步骤 3　For $n=r_3+r_4-2$ to 1 do

For $a=0$ to r_3-1 do

$b=n-a$；

If $b\ge r_4$, continue；

If $a=0$, 令 $\widetilde{E}^a=\{(\alpha\to\alpha)\,|\,\exists\beta,$ s.t. $(\alpha\to\beta)\in A\}$；

Else 根据 a 轮输出的完全性构造 $\widetilde{E}^a$；

If $b=0$，令 $E^b=\{(\beta\to\beta)\,|\,\exists\alpha,$ s.t. $(\alpha\to\beta)\in A\}$；

Else 根据 b 轮输出的完全性 E^b；

For 所有 $(\alpha_1\to\beta_1)\in A$，所有 $(\alpha_2\to\beta_2)\in\widetilde{E}^a$，所有 $(\alpha_3\to\beta_3)\in E^b$ do

If $\beta_2\prec\alpha_1$ 且 $\alpha_3\prec\beta_1$，令 $A'=A'\bigcup(\alpha_2\to\beta_3)$；

If $A'\ne\varnothing$，令 $r'=n$，输出不可能差分区分器的集合 A' 及轮数 r' 并终止。

If $A'=\varnothing$，输出未找到扩展的不可能差分并终止算法。

*同算法 6 中执行的算法 5

3.3 构造方法比较

与穷尽搜索和转化为混合整数规划问题(MILP)的自动搜索技术相比，本文提出的基于完全性构造不可能差分的方法搜索速度快，例如文献[12]利用 MILP 寻找 SPECK64 算法的不可能差分区分器，找到 157 条 6 轮单比特不可能差分区分器需要约 10 min，而本文在同等环境下寻找到 SPECK64 算法的 6 轮高重量不可能差分区分器只需要约 1 s，当分组规模增大时这种优势将更加明显。

与模式运算搜索方法相比，本文的方法可直接搜索大重量不可能差分传递链，不需要对单比特不可能差分区分器进行后续的组合来构造大重量的不可能差分区分器。与弱扩散构造方法相比，本文的方法在构造区分器的数量和长度方面均有优势。此外在针对 MOC 模型不可能差分单比特矛盾构造情况下，本文的方法从理论上(在完全性表示是精确的前提下)可以给出所有的不可能差分区分器，从而可以比其它两种方法构造更多的不可能差分区分器。

上述几种不可能差分构造方法的对比如表 8 所示。

4　基于完全性的不可能差分区分器构造法的应用

本节应用完全性构造方法对 SIMON 系列、Speck 系列算法的不可能差分区分器进行了寻找，取得了较好的效果。对 SIMON 系列算法，找到了目前公开文献中的所有最长的不可能区分器，对 SPECK 系列算法，还找到了一些新的等长的不可能差分区分器。

SIMON 系列算法[3]和 SPECK 系列算法[3]均是由美国国家安全局于 2013 年提出的轻量级分组密码算法，它们的分组规模分别为 32 bit, 48 bit, 64 bit, 96 bit 和 128 bit，它们的结构框图如图 3 和图 4 所示，具体算法见文献[3]。其中 SIMON 算法的整体结构为 Feistel 结构，轮函数由逻辑与、异或及循环移位运算构成；SPECK 算法的整体结构为基于 ARX 运算的 Feistel 变形结构。

4.1 SIMON 系列算法的不可能差分区分器

对于 SIMON32 算法，将算法结构代入算法 6，得到了 32 条 11 轮的不可能差分区分器，经比较它们与文献[7]中穷尽搜索得到的 32 条不可能差分区分器相同。对于其它分组长度的 SIMON 算法，均得到了全部公开文献中最长的不可能区分器(具体结果见表 9)，说明了完全性构造方法的有效性。

表 8 几种MOC不可能差分区分器构造方法的比较

搜索方法	速度	重量	数量	长度	理论依据	搜索策略	文献
穷举搜索	慢	高重量	所有	长	–	穷举	–
MILP 自动搜索	较慢	低重量	少	较长	比特级差分扩散	MILP 求解	[12,15]
模式运算构造	快	低重量†	较多	较长	模式运算	中间相错(单比特矛盾)	[7]
弱扩散构造	快	高重量	少	短	简单差分扩散	中间相错	[10]
完全性构造	快	高重量	多	较长	完全性	中间相错(单比特矛盾)	本文

†组合后达到高重量

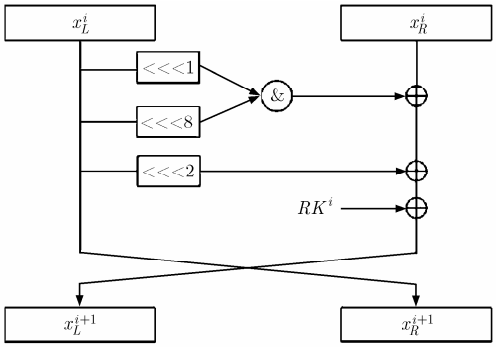

图 3 SIMON算法轮函数结构图

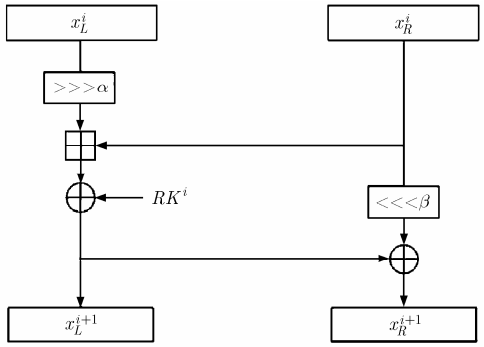

图 4 SPECK算法轮函数结构图

表 9 SIMON系列算法不可能差分区分器搜索结果

算法	区分器长度	区分器总个数
SIMON32	11	32
SIMON48	12	48
SIMON64	13	64
SIMON96	16	96
SIMON128	19	128

我们求得 SIMON 算法各版本的全部最长轮数不可能差分区分器的时间均小于 1 s(实验环境为 3.2 GHzCPU, 4 GB 内存)。并且我们的方法可以直接得到完整的不可能差分区分器，而不用文献[7]中 EBT 构造方法那样通过将单比特差分进行桥接和测试的方法来构造。

4.2 SPECK 系列算法的不可能差分区分器

对于 SPECK32 算法，将算法结构代入算法 6，得到了 5 轮的不可能差分区分器的集合 A，再将 A 代入算法 7，得到 5 条 6 轮的不可能差分区分器。

对于其它规模的 SPECK 算法，本文利用算法 6 和算法 7 找到现有的最长的区分器，还找到了一些新的区分器，表 10 给出了本文首次给出的不可能差分区分器。

表 10 SPECK系列算法新的不可能差分区分器搜索结果

规模	区分器长度	矛盾	差分集合
48	6 轮	S_{41}^2	in $*1(0)_{14}(*)_8\ (0)_3(*)_6 1(0)_{14}$ out $1(0)_{23}\ 1(0)_{20}100$
64	6 轮	S_{57}^2	in $(*)_9 1(0)_{14}(*)_8\ (0)_3(*)_{14}1(0)_{14}$ out $1(0)_{31}\ 1(0)_{28}100$
96	6 轮	S_{89}^2	in $(*)_{25}1(0)_{14}(*)_8\ (0)_3(*)_{30}1(0)_{14}$ out $1(0)_{47}\ 1(0)_{44}100$
128	6 轮	S_{121}^2	in $(*)_{41}1(0)_{14}(*)_8\ (0)_3(*)_{46}1(0)_{14}$ out $1(0)_{63}\ 1(0)_{60}100$

表示此比特任意，$(*)_i$ 表示连续 i 个，$(0)_i$ 表示连续 i 个 0

5 结束语

本文提出了针对 MOC 的完全性通用算法，并在此基础上提出了利用完全性寻找 MOC 的不可能差分区分器的方法，与穷尽搜索和转化为 MILP 的自动搜索技术相比，本文方法具有更快的搜索速度，与模式运算构造和弱扩散构造方法相比，本文方法在构造不可能差分区分器的重量和数量上有优势。将此方法应用于搜索 SIMON 算法和 SPECK 算法的不可能差分区分器时提高了搜索效率，但是不可能差分攻击的轮数没有增加，对这些算法的安全性并不构成威胁。本文的结果说明了 MOC 算法完全性和不可能差分(单比特矛盾)区分器之间存在的内在联系，为不可能差分区分器的实际构造提供了切实的理论指导和技术支持，在 MOC 算法安全分析方面具有较高的实际应用价值。下一步拟将此不可能差分搜索算法应用于其它 MOC 型算法中，寻找更长轮或更多的不可能差分区分器，并尝试寻找完全性和零相关线性区分器的关系。

参 考 文 献

[1] WU Wenling, ZHANG Wentao, and FENG Dengguo. Impossible differential cryptanalysis of reduced-round ARIA and Camellia[J]. *Journal of Computer Science and Technology*, 2007, 22(3): 449–456. doi: 10.1007/s11390-007-9056-0.

[2] 付立仕, 金晨辉. MIBS-80 的 13 轮不可能差分分析[J]. 电子与信息学报, 2016, 38(4): 848–855. doi: 10.11999/JEIT150673.
FU Lishi and JIN Chenhui. Impossible differential cryptanalysis on 13-round MIBS-80[J]. *Journal of Electronics & Information Technology*, 2016, 38(4): 848–855. doi: 10.11999/JEIT150673.

[3] 唐学海, 李超, 王美一, 等. 3D 密码的不可能差分攻击[J]. 电子与信息学报, 2010, 32(10): 2516–2520. doi: 10.3724/SP.J.1146.2009.01375.
TANG Xuehai, LI Chao, WANG Meiyi, *et al*. Impossible differential attack on 3D cipher[J]. *Journal of Electronics &*

Information Technology, 2010, 32(10): 2516–2520. doi: 10.3724/SP.J.1146.2009.01375.

[4] 张凯. 基于混合运算密码模型的安全性研究[D]. [博士论文], 信息工程大学, 2016.
ZHANG Kai. Research on the security evaluation against mixed operation based cipher model[D]. [Ph.D. dissertation], Information Engineering University, 2016.

[5] HONG D, SUNG J, HONG S, *et al*. HIGHT: A new block cipher suitable for low-resource device[C]. International Workshop on Cryptographic Hardware and Embedded Systems, Yokohama, 2006: 46–59. doi: 10.1007/11894063_4.

[6] BEAULIEU R, TREATMAN-CLARK S, SHORS D, *et al*. The SIMON and SPECK lightweight block ciphers[C]. 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), Texas, 2015: 1–6. doi: 10.1145/2744769.2747946.

[7] BIHAM E, BIRYUKOV A, and SHAMIR A. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials [C]. International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech, 1999: 12–23. doi: 10.1007/3-540-48910-X_2.

[8] National Security Agency. Skipjack and KEA algorithm specifications, Version 2.0.[OL]. http: scrc.nist.gov/ Crypto Toolkit/skipjack/skipjack-kea.htm. 1998.

[9] BIHAM E, DUNKELMAN O, and KELLER N. Related-key impossible differential attacks on 8-round AES-192[C]. Topics in Cryptology-CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, 2006: 21–33. doi: 10.1007/11605805_2.

[10] CHEN J, WANG M, and PRENEEL B. Impossible differential cryptanalysis of the lightweight block ciphers TEA, XTEA and HIGHT[C]. International Conference on Cryptology in Africa. Ifrance, Morocco, 2012: 117–137. doi: 10.1007/978-3-642-31410-0_8.

[11] SUN Siwei, HU Lei, WANG Peng, *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]. International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, 2014: 158–178. doi: 10.1007/978-3-662-45611-8_9.

[12] CUI Ting, JIA Keting, FU Kai, *et al*. New automatic search tool for impossible differentials and zero-correlation linear approximations[OL]. http://eprint.iacr.org/2016/689.pdf. 2017.04.

[13] 李俊志. 三类非线性反馈移存器模型的代数性质研究及应用[D]. [硕士论文], 信息工程大学, 2015.
LI Junzhi. Algebraic properties and applications on three non-linear feedback models[D]. [Master. dissertation], Information Engineering University, 2015.

[14] 金晨辉, 郑浩然, 张少武, 等. 密码学[M]. 北京: 高等教育出版社, 2009: 166–167.
JIN Chenhui, ZHENG Haoran, ZHANG Shaowu, *et al*. Cryptography[M]. Beijing: Higher Education Press, 2009: 166–167.

[15] LEE H C, KANG H C, HONG D, *et al*. New impossible differential characteristic of SPECK64 using MILP[OL]. http://eprint.iacr.org/2016/1137.pdf.2017.04.

李俊志：　男，1990 年生，博士，研究方向为密码学.
关　杰：　女，1974 年生，教授，主要研究方向为密码设计与分析.