数字视频广播通用加扰算法的不可能差分分析

沈 璇[①②] 孙 兵[①②] 刘国强[①] 李 超*[①]

[①](国防科技大学文理学院 长沙 410073)

[②](湖南警察学院网络侦查技术湖南省重点实验室 长沙 410073)

摘 要：数字视频广播通用加扰算法(DVB-CSA)是一种混合对称加密算法，由分组密码加密和流密码加密两部分组成。该算法通常用于保护视讯压缩标准(MPEG-2)中的信号流。主要研究DVB-CSA分组加密算法(DVB-CSA-Block Cipher, CSA-BC)的不可能差分性质。通过利用S盒的具体信息，该文构造了CSA-BC的22轮不可能差分区分器，该区分器的长度比已有最好结果长2轮。进一步，利用构造的22轮不可能差分区分器，攻击了缩减的25轮CSA-BC，该攻击可以恢复24 bit种子密钥。攻击的数据复杂度、时间复杂度和存储复杂度分别为$2^{53.3}$个选择明文、$2^{32.5}$次加密和2^{24}个存储单元。对于CSA-BC的不可能差分分析，目前已知最好结果能够攻击21轮的CSA-BC并恢复16 bit的种子密钥量。就攻击的长度和恢复的密钥量而言，该文的攻击结果大大改进了已有最好结果。

关键词：混合对称密码；分组密码；数字视频广播通用加扰算法；不可能差分分析

中图分类号：TN918.1 **文献标识码**：A **文章编号**：1009-5896(2019)01-0046-07

DOI: 10.11999/JEIT180245

Impossible Differential Cryptanalysis of the Digital Video Broadcasting-common Scrambling Algorithm

SHEN Xuan[①②] SUN Bing[①②] LIU Guoqiang[①] LI Chao[①]

[①](*College of Liberal Arts and Sciences, National University of Defense Technology, Changsha* 410073, *China*)

[②](*Hunan Provincial Key Laboratory of Network Investigational Technology, Changsha* 410073, *China*)

Abstract: The Digital Video Broadcasting-Common Scrambling Algorithm (DVB-CSA) is a hybrid symmetric cipher. It is made up of the block cipher encryption and the stream cipher encryption. DVB-CSA is often used to protect MPEG-2 signal streams. This paper focuses on impossible differential cryptanalysis of the block cipher in DVB-CSA called CSA-BC. By exploiting the details of the S-box, a 22-round impossible differential is constructed, which is two rounds more than the previous best result. Furthermore, a 25-round impossible differential attack on CSA-BC is presented, which can recover 24 bit key. For the attack, the data complexity, the computational complexity and the memory complexity are $2^{53.3}$ chosen plaintexts, $2^{32.5}$ encryptions and 2^{24} units, respectively. For impossible differential cryptanalysis of CSA-BC, the previous best result can attack 21-round CSA-BC and recover 16 bit key. In terms of the round number and the recovered key, the result significantly improves the previous best result.

Key words: Hybrid symmetric cipher; Block cipher; Digital Video Broadcasting-Common Scrambling Algorithm (DVB-CSA); Impossible differential cryptanalysis

收稿日期：2018-03-16；改回日期：2018-07-25；网络出版：2018-08-06

*通信作者： 李超 academic_lc@163.com

基金项目：国家重点研发计划(2017YFB0802000)，国家自然科学基金(61672530, 61702537, 61772545)，湖南省教育厅优秀青年项目(16B086)，网络侦查技术湖南省重点实验室开放基金(2016WLZC018)

Foundation Items: The National Key R&D Program of China (2017YFB0802000), The National Natural Science Foundation of China (61672530, 61702537, 61772545), The Project of Hunan Province Department of Education (16B086), The Open Research Fund of Hunan Provincial Key Laboratory of Network Investigational Technology (2016WLZC018)

1　引言

数字视频广播通用加扰算法(DVB-CSA)主要用于保护MPEG-2中的信号，例如付费电视中的数字转换信号。1994年，该算法被DVB联合企业采用。2002年以前，由于一些安全方面的原因，CSA的软件运行是被禁止的。2002年，在软件团队FreeDec的帮助下，利用逆向工程技术，CSA的加密流程被公之于众。

在CSA的加密细节被公开之后，陆续出现了许多关于CSA安全性方面的分析结果。2004年，Weinmann等人[1]利用猜测决定攻击方法对CSA的流密码算法(DVB-CSA-Stream Cipher, CSA-SC)进行了分析。之后，在ICCSA 2005会议上，Wirt[2]针对CSA的分组密码算法(CSA-BC)进行了故障攻击。2009年，Simpson等人[3]指出文献[1]中存在一个错误，并且针对CSA-SC提出了时空折中的攻击方法。后来，在2011年，Tews等人[4]利用彩虹表技术针对缩减版本的CSA提出了时空折中的攻击方法。2015年，Zhang等人[5]针对CSA-SC提出了一个区分攻击。最近，Zhang等人[6]又针对缩减轮的CSA-BC进行了不可能差分分析。

不可能差分分析是一种针对分组密码算法非常有效的密码分析方法[7−13]。该方法由Knudsen[14]和Biham等人[15]独立提出。其主要思想是利用概率为零的差分来筛除错误密钥进而得到正确密钥。不可能差分分析分为区分器的构造和密钥恢复两步。第1步区分器的构造主要是构造尽可能长的不可能差分区分器，第2步密钥恢复是利用构造的不可能差分区分器恢复正确密钥。其中，第1步不可能差分区分器的构造非常关键，它是整个攻击的基础。不可能差分区分器通常是利用中间相错技术来构造。

2016年，Zhang等人[6]利用一条20轮不可能差分区分器攻击了缩减的21轮CSA-BC。进一步，他们利用CSA整体结构上的缺陷将该结果推广到整个CSA中。注意到他们在构造CSA-BC的不可能差分区分器时仅仅利用了S盒是双射的性质，并没有利用到S盒的具体信息，故只能利用中间差分第1个字节的信息来构造矛盾。因此，如果能充分挖掘S盒的具体细节，利用中间差分更多字节信息来构造矛盾，则有望获得更长轮数的不可能差分区分器。

本文主要研究CSA-BC的不可能差分性质，并利用文献[6]中提出的CSA整体结构上的缺陷将CSA-BC的结果推广到整个CSA中。首先，本文通过求解带S盒具体信息的差分方程组，构造了CSA-BC的22轮不可能差分区分器，该区分器长度比已有最好结果长2轮。其次，本文利用构造的22轮不可能差分区分器，攻击了缩减的25轮CSA-BC，该攻击能恢复24 bit种子密钥。攻击的数据复杂度为$2^{53.3}$个选择明文，时间复杂度为$2^{32.5}$次加密，存储复杂度为2^{24}个存储单元。最后，本文将25轮CSA-BC的分析结果推广到整个CSA中。

论文后续安排如下：第2节介绍了CSA整体结构和CSA-BC；第3节构造了CSA-BC的22轮不可能差分区分器；第4节给出了CSA-BC的25轮不可能差分攻击；第5节给出了本文结果和已有最好结果的比较；最后总结全文。

2　CSA简介

本文所用符号定义如下：K为CSA中的种子密钥，K^E为CSA-BC的扩展密钥，k_i为K的第i比特。SB_i为第i块调制数据流，CB_i为第i块流密码输出，IB_i为第i个中间块，DB_i为第i块解调数据流。R为小于8 Byte的剩余量，SR为调制剩余量。$\Delta S(a)$表示输入差分a经过S盒后的所有可能的输出差分集合。“|”表示两个比特串之间的连结。

2.1　CSA整体结构

CSA加密流程由两部分组成，分别为分组加密部分和流密码加密部分。这两部分共用相同的种子密钥K，如图1所示。m Byte的明文首先分为若干个8 Byte的数据块$(\mathrm{DB}_0, \mathrm{DB}_1, \cdots, \mathrm{DB}_{n-1})$。最后不足8 Byte的数据块记为剩余量$R$。

CSA按照图1中的方式进行加密。注意到CSA-SC的随机数由中间值IB_0提供，调制数据流SB_i由CSA-SC生成的密钥流CB_i和CSA-BC加密后的中间值IB_i异或得到。最后的调制剩余量SR由最后一块密钥流CB_n与剩余量R直接异或得到。

本文主要研究CSA-BC的不可能差分性质，CSA-SC的具体细节描述可参考文献[1]。

2.2　CSA-BC描述

CSA-BC可以视为一个迭代56轮的分组加密算法。它的分组长度为64 bit，每轮中的64 bit状态可以看成一个8 Byte的向量。CSA-BC的密钥为64 bit的种子密钥K，它通过密钥扩展方案可以扩展成为448 bit的扩展密钥K^E。表1的算法1给出了CSA-BC的加密流程。

CSA-BC加密流程中轮函数f如图2所示。

令轮函数的输入为$\boldsymbol{x}=(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$，输出为$\boldsymbol{y}=(y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$，这里$x_i$, y_i均为一个字节，则轮函数f可以表示为

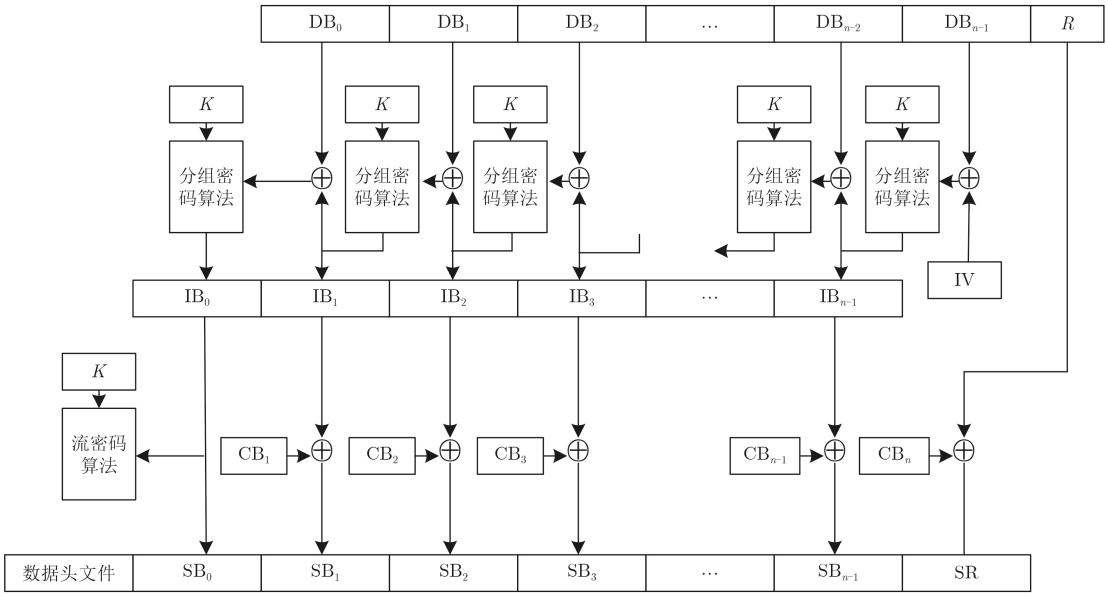

图 1 CSA的整体结构

表 1 算法1：CSA-BC的加密流程

输入：明文$\boldsymbol{M}=(M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7)$

输出：密文$\boldsymbol{C}=(C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7)$

(1) $\boldsymbol{S}^0=\boldsymbol{M}$;

(2) for r=0 to 55

(3) $\boldsymbol{S}^{r+1}=f(\boldsymbol{S}^r,(k_{8r}^E,k_{8r+1}^E,\cdots,k_{8r+7}^E))$;

(4) end for

(5) $\boldsymbol{C}=\boldsymbol{S}^{56}$.

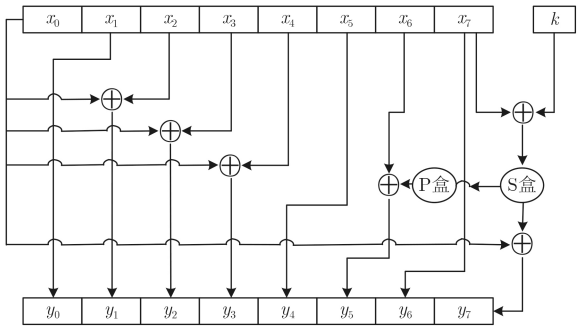

图 2 CSA-BC加密的轮函数

$$\left.\begin{aligned} y_0&=x_1\\ y_1&=x_2\oplus x_0\\ y_2&=x_3\oplus x_0\\ y_3&=x_4\oplus x_0\\ y_4&=x_5\\ y_5&=x_6\oplus P\circ S(x_7\oplus k)\\ y_6&=x_7\\ y_7&=x_0\oplus S(x_7\oplus k) \end{aligned}\right\} \tag{1}$$

其中，P盒是一个比特置换，记为P，它将输入的第0, 1, 2, 3, 4, 5, 6, 7 bit依次变为输出的第1, 7, 5, 4, 2, 6, 0, 3 bit的数。令P盒的输入为$a=(a_7a_6a_5a_4a_3a_2a_1a_0)\in F_{2^8}$，这里$a_i\in F_2$表示字节$a$的第$i$比特；输出为$b=(b_7b_6b_5b_4b_3b_2b_1b_0)\in F_{2^8}$，这里$b_i\in F_2$表示字节$b$的第$i$比特。P盒变换的矩阵表示为

$$b\triangleq \boldsymbol{P}a\Leftrightarrow \begin{pmatrix} b_7\\ b_6\\ b_5\\ b_4\\ b_3\\ b_2\\ b_1\\ b_0 \end{pmatrix}= \begin{pmatrix} 0&0&0&0&0&0&1&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0&0\\ 1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&0 \end{pmatrix} \begin{pmatrix} a_7\\ a_6\\ a_5\\ a_4\\ a_3\\ a_2\\ a_1\\ a_0 \end{pmatrix} \tag{2}$$

注意在本文中，矩阵与字节相乘定义为矩阵与该字节相应比特分量构造的列向量相乘，如上所示。轮函数中S盒是一个字节置换，记为S，其具体表示参见文献[6]。

密钥扩展方案 $k_{0,\cdots,63}$表示64 bit的种子密钥，则扩展密钥$\boldsymbol{K}^E=(k_0^E,k_1^E,\cdots,k_{447}^E)$可以通过式(3)得到：

$$\left.\begin{aligned} k_{0,\cdots,63}^E&=k_{0,\cdots,63}\\ k_{64i,\cdots,64i+63}^E&=\rho(k_{64(i-1),\cdots,64i-1}^E)\oplus C[i],\\ &\quad 1\le i\le 6 \end{aligned}\right\} \tag{3}$$

这里，$C[i]$表示一个64 bit的常数，ρ为一个比特置换，其具体表示参见文献[6]。

3 CSA-BC不可能差分区分器构造

文献[6]构造了CSA-BC的20轮不可能差分区分器。作者仅仅利用了S盒是双射的性质，故只能利用中间差分第1个字节的信息来构造矛盾。本文利用了S盒的具体信息，考虑中间差分全部8个字节，

建立含S盒具体信息的差分方程组，对于给定的输入输出差分，利用计算机编程可以遍历搜索所有可能的22轮差分特征。因此，当遍历具有特定形式的所有22轮差分后，除可能的差分外，剩下的均为22轮不可能差分。与文献[6]相比，本文构造的不可能差分区分器长2轮。

首先，回顾文献[6]构造20轮不可能差分区分器的方法：从加密方向，给定输入差分为$(0|0|0|0|0|0|u|0)$，经过12轮加密，以概率为1演化为中间差分$(0|?|?|?|?|?|?|?)$，这里$u\neq 0$且?表示一个未知的字节。与此同时，从解密方向，给定输出差分为$(0|v|v|v|0|0|0|v)$，经过8轮解密，以概率为1演化为中间差分$(\Delta S(v)|0|\Delta S(v)|\Delta S(v)|\Delta S(v)|0|\boldsymbol{P}\Delta S(v)|v)$，这里$v\neq 0$。对于从加解密方向分别得到的中间差分的第1个字节，考虑到S盒为双射函数，则有$0\notin\Delta S(v)$。因此，$(0|0|0|0|0|0|u|0)\rightarrow(0|v|v|v|0|0|0|v)$是一条20轮不可能差分。

上述区分器的构造仅仅利用了中间差分第1个字节来构造矛盾。因此，如果中间差分的所有字节都能够用来构造矛盾，则可能构造出更长的不可能差分区分器。基于上述思想，对于相同的输入差分$(0|0|0|0|0|0|u|0)$和输出差分$(0|v|v|v|0|0|0|v)$，以概率1进行传播，从加密方向加密14轮同时从解密方向解密8轮，得到两个方向的中间差分。进一步，匹配中间差分的所有字节建立差分方程组。当该差分方程组存在满足约束条件的解时，则该条差分为22轮可能的差分，反之，该条差分为22轮不可能差分。注意到，解差分方程组需要利用到S盒的具体信息而不能仅仅将其看成双射函数。根据上述方法，本文构造了CSA-BC如下的22轮不可能差分区分器。

命题1　$(0|0|0|0|0|0|u|0)\xrightarrow{22r}(0|v|v|v|0|0|0|v)$是CSA-BC的一条22轮不可能差分，这里$u,v\neq 0$并且$v\in\Omega$，其中集合$\Omega$如式(4)：

$$\begin{aligned}\Omega=F_{2^8}^{*}\backslash\{&10,13,25,26,27,32,35,41,53,59,\\&64,71,73,86,87,102,110,111,114,117,\\&135,140,143,158,194,196,221,226\}\end{aligned}\tag{4}$$

证明　当22轮差分的输入和输出分别为$(0|0|0|0|0|0|u|0)$和$(0|v|v|v|0|0|0|v)$时，输入输出差分分别从加解密方向以概率1进行演化，它们的传播规律分别如表2和表3所示。

与文献[6]相比，解密方向是相同的，但是对于加密方向本文多传播了两轮。如果该条22轮差分是可能的，则从加解密方向得到的两个中间差分在所有8个字节处均能匹配成功。即有式(5)所示差分方程组成立。

$$\left.\begin{aligned}v_1&=u\oplus\boldsymbol{P}u_2\\0&=u_1\oplus\boldsymbol{P}u_3\oplus u\oplus\boldsymbol{P}u_1\\v_1&=u_2\oplus\boldsymbol{P}u_4\oplus\boldsymbol{P}u_1\\v_1&=u_3\oplus\boldsymbol{P}u_5\oplus\boldsymbol{P}u_1\\v_1&=u_4\oplus\boldsymbol{P}u_6\\0&=u_5\oplus\boldsymbol{P}u_7\\\boldsymbol{P}v_1&=u_6\\v&=u\oplus\boldsymbol{P}u_1\oplus u_7\end{aligned}\right\}\tag{5}$$

表 2　加密方向的差分传播规律

轮数	差分传播	约束条件
0	$(0\|0\|0\|0\|0\|0\|u\|0)$	
1	$(0\|0\|0\|0\|0\|u\|0\|0)$	
2	$(0\|0\|0\|0\|u\|0\|0\|0)$	
3	$(0\|0\|0\|u\|0\|0\|0\|0)$	
4	$(0\|0\|u\|0\|0\|0\|0\|0)$	
5	$(0\|u\|0\|0\|0\|0\|0\|0)$	
6	$(u\|0\|0\|0\|0\|0\|0\|0)$	
7	$(0\|u\|u\|u\|0\|0\|0\|u)$	
8	$(u\|u\|u\|0\|0\|\boldsymbol{P}u_1\|u\|u_1)$	$u_1\in\Delta S(u)$
9	$(u\|0\|u\|u\|\boldsymbol{P}u_1\|u\oplus\boldsymbol{P}u_2\|u_1\|u\oplus u_2)$	$u_2\in\Delta S(u_1)$
10	$(0\|0\|0\|u\oplus\boldsymbol{P}u_1\|u\oplus\boldsymbol{P}u_2\|u_1\oplus\boldsymbol{P}u_3\|u\oplus u_2\|u\oplus u_3)$	$u_3\in\Delta S(u\oplus u_2)$
11	$(0\|0\|u\oplus\boldsymbol{P}u_1\|u\oplus\boldsymbol{P}u_2\|u_1\oplus\boldsymbol{P}u_3\|u\oplus u_2\oplus\boldsymbol{P}u_4\|u\oplus u_3\|u_4)$	$u_4\in\Delta S(u\oplus u_3)$
12	$(0\|u\oplus\boldsymbol{P}u_1\|u\oplus\boldsymbol{P}u_2\|u_1\oplus\boldsymbol{P}u_3\|u\oplus u_2\oplus\boldsymbol{P}u_4\|u\oplus u_3\oplus\boldsymbol{P}u_5\|u_4\|u_5)$	$u_5\in\Delta S(u_4)$
13	$(u\oplus\boldsymbol{P}u_1\|u\oplus\boldsymbol{P}u_2\|u_1\oplus\boldsymbol{P}u_3\|u\oplus u_2\oplus\boldsymbol{P}u_4\|u\oplus u_3\oplus\boldsymbol{P}u_5\|u_4\oplus\boldsymbol{P}u_6\|u_5\|u_6)$	$u_6\in\Delta S(u_5)$
14	$(u\oplus\boldsymbol{P}u_2\|u_1\oplus\boldsymbol{P}u_3\oplus u\oplus\boldsymbol{P}u_1\|u_2\oplus\boldsymbol{P}u_4\oplus\boldsymbol{P}u_1\|$ $u_3\oplus\boldsymbol{P}u_5\oplus\boldsymbol{P}u_1\|u_4\oplus\boldsymbol{P}u_6\|u_5\oplus\boldsymbol{P}u_7\|u_6\|u\oplus\boldsymbol{P}u_1\oplus u_7)$	$u_7\in\Delta S(u_6)$

表 3 解密方向的差分传播规律

轮数	差分传播	约束条件
22	$(0\|v\|v\|v\|0\|0\|0\|v)$	
21	$(v\|0\|0\|0\|0\|0\|0\|0)$	
20	$(0\|v\|0\|0\|0\|0\|0\|0)$	
19	$(0\|0\|v\|0\|0\|0\|0\|0)$	
18	$(0\|0\|0\|v\|0\|0\|0\|0)$	
17	$(0\|0\|0\|0\|v\|0\|0\|0)$	
16	$(0\|0\|0\|0\|0\|v\|0\|0)$	
15	$(0\|0\|0\|0\|0\|0\|v\|0)$	
14	$(v_1\|0\|v_1\|v_1\|v_1\|0\|\boldsymbol{P}v_1\|v)$	$v_1 \in \Delta S(v)$

化简上述方程组后，得

$$\left.\begin{aligned}
&(\boldsymbol{I}\oplus\boldsymbol{P}^{-1}\oplus\boldsymbol{P}\oplus\boldsymbol{P}^{3})u_2\\
&\quad=u_1\oplus(\boldsymbol{I}\oplus\boldsymbol{P}^{-1}\oplus\boldsymbol{P}^{2})u\\
&u_3=(\boldsymbol{I}\oplus\boldsymbol{P}^{-1})u_1\oplus\boldsymbol{P}^{-1}u\\
&u_4=(\boldsymbol{I}\oplus\boldsymbol{P}^{-1})u_2\oplus u_1\oplus\boldsymbol{P}^{-1}u\\
&u_5=\boldsymbol{P}^{-1}u_3\oplus u_2\oplus u_1\oplus\boldsymbol{P}^{-1}u\\
&u_6=\boldsymbol{P}^{-1}u_4\oplus u_2\oplus\boldsymbol{P}^{-1}u\\
&u_7=\boldsymbol{P}^{-1}u_5\\
&v=u\oplus\boldsymbol{P}u_1\oplus u_7\\
&v_1=u\oplus\boldsymbol{P}u_2
\end{aligned}\right\}\tag{6}$$

在差分方程组式(6)中，$\boldsymbol{I}$和$\boldsymbol{P}^{-1}$分别表示单位矩阵和置换$\boldsymbol{P}$的逆矩阵。对于式(6)的第1个等式，通过计算可以得到$\boldsymbol{I}\oplus\boldsymbol{P}^{-1}\oplus\boldsymbol{P}\oplus\boldsymbol{P}^{3}$的秩为7。因此，如果式(6)的第1个等式有解，则解的个数有2个。进一步，当遍历u和u_1所有可能的值时，通过式(6)从上至下的8个等式，可以依次得到u_2, u_3, u_4, u_5, u_6, u_7, v, v_1的值。注意到，u, u_1, u_2, u_3, u_4, u_5, u_6, u_7, v, v_1还需要满足表2和表3中的约束条件，即

$$\left.\begin{aligned}
&u_1\in\Delta S(u)\\
&u_2\in\Delta S(u_1)\\
&u_3\in\Delta S(u\oplus u_2)\\
&u_4\in\Delta S(u\oplus u_3)\\
&u_5\in\Delta S(u_4)\\
&u_6\in\Delta S(u_5)\\
&u_7\in\Delta S(u_6)\\
&v_1\in\Delta S(v)
\end{aligned}\right\}\tag{7}$$

结合差分方程组式(6)和约束条件式(7)，遍历所有可能的(u, v)，这里u, v均为非零字节，其取值均从1到255。当(u, v)同时满足式(6)和式(7)时，将其放入集合Φ中。通过计算机编程搜索，可得集合Φ：

$$\begin{aligned}
\Phi=\{&(1,114),(3,41),(7,64),(15,87),(25,221),\\
&(45,53),(48,135),(57,26),(72,13),(73,32),\\
&(73,111),(80,194),(105,86),(106,196),\\
&(107,71),(110,27),(130,73),(131,140),\\
&(134,226),(165,102),(181,35),(186,143),\\
&(198,226),(199,135),(212,59),(226,117),\\
&(229,25),(233,114),(239,110)(241,10),\\
&(253,158)\}
\end{aligned}\tag{8}$$

在集合Φ中的任意元素(u, v)均使得22轮差分$(0|0|0|0|0|0|u|0)\rightarrow(0|v|v|v|0|0|0|v)$为可能差分。反之，若$(u, v)$取值不在集合$\Phi$中，则该$(u, v)$使得给定形式的22轮差分为22轮不可能差分。考虑不可能差分的截断形式，对于任意非零变量u，若v均满足(u, v)使给定形式的22轮差分为不可能差分，则将v的取值放入集合Ω中，则集合Ω为

$$\begin{aligned}
\Omega&=\{v\in F_{2^8}^{*}\,|\forall u\in F_{2^8}^{*},(u,v)\notin\Phi\}\\
&=F_{2^8}^{*}\backslash\{10,13,25,26,27,32,35,41,53,59,64,71,\\
&\quad 73,86,87,102,110,111,114,117,135,140,143,\\
&\quad 158,194,196,221,226\}
\end{aligned}\tag{9}$$

这里$F_{2^8}^{*}$表示有限域F_{2^8}上所有非零元素的集合。根据集合Ω的表示，知Ω的元素个数为227。综上所述，命题1成立。 证毕

4 CSA-BC的25轮不可能差分攻击

利用上述构造的22轮不可能差分区分器，本文对CSA-BC进行25轮不可能差分攻击，攻击过程如图3所示。

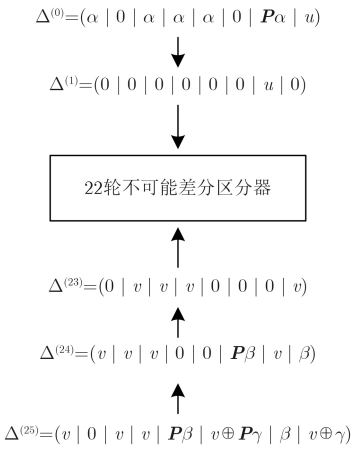

图 3 CSA-BC的25轮不可能差分攻击

令$\Delta^{(i)}$表示第i轮的差分，整个25轮不可能差分攻击步骤如下：

步骤1 选择一个结构。对于该结构，任意选择两个明文M_1和M_2，它们均满足如下差分：

$$M_1 \oplus M_2 = (\alpha|0|\alpha|\alpha|\alpha|0|\boldsymbol{P}\alpha|u) \tag{10}$$

这里$\alpha, u \neq 0$。该结构能够提供$2^8 \times 2^8 = 2^{16}$个明文和$2^{16} \times 2^{16}/2 = 2^{31}$组明文对。

步骤2　选择2^N个上述结构，则这些结构总共能够提供2^{N+16}个明文，2^{N+31}组明文对。选择使得密文差分满足式(1)的明密文对：

$$\Delta^{(25)} = (v|0|v|v|\boldsymbol{P}\beta|v \oplus \boldsymbol{P}\gamma|\beta|v \oplus \gamma) \tag{11}$$

这里$v \in \Omega$, $\beta, \gamma \neq 0$。考虑到集合Ω的元素个数为227，满足上述密文差分形式的概率约为$(227 \times (2^8-1) \times (2^8-1))/2^{8\times 8} \approx 2^{-40.2}$。因此，经过步骤2后，剩下的明密文对数目约为$2^{N+31} \times 2^{-40.2} = 2^{N-9.2}$。

步骤3　对于经过步骤2后的剩余明密文对，猜测第24轮的8 bit轮密钥$k^E_{192,193,\cdots,199}$，选择使得差分方程$\gamma = \Delta S(\beta)$成立的明密文对。上述差分方程成立的概率约为2^{-8}。则经过步骤3后的剩余明密文对的数目约为$2^{N-9.2} \times 2^{-8} = 2^{N-17.2}$。

步骤4　对于经过步骤3后的剩余明密文对，猜测第23轮的8 bit轮密钥$k^E_{184,185,\cdots,191}$，选择使得差分方程$\beta = \Delta S(v)$成立的明密文对。上述差分方程成立的概率约为2^{-8}。则经过步骤4后的剩余明密文对的数目约为$2^{N-17.2} \times 2^{-8} = 2^{N-25.2}$。

步骤5　对于经过步骤4后的剩余明密文对，猜测初始的8 bit密钥$k^E_{0,1,\cdots,7}$，并且检测差分方程$\alpha = \Delta S(u)$是否成立。如果该差分方程成立，则去掉猜测的24 bit候选密钥值$(k^E_{0,1,\cdots,7}|k^E_{184,185,\cdots,191}|k^E_{192,193,\cdots,199})$。

因为中间22轮差分是不可能差分，所以当猜测的24 bit候选密钥满足不可能差分的输入输出时，该24 bit候选密钥为错误密钥。在分析了$2^{N-25.2}$组明密文对后，错误候选密钥仍然能被保留的数目约为

$$(2^{24}-1)(1-2^{-8})^{2^{N-25.2}} \approx 2^{24} \times e^{-2^{N-33.2}} \tag{12}$$

当$N = 37.3$时，$2^{24} \times e^{-2^{N-33.2}} = 2^{24} \times e^{-2^{4.1}} \approx 0.62 < 1$。则经过上述步骤后，只有唯一的24 bit候选密钥被保留下来，因此它可以被认为是正确的候选密钥值。进一步，根据密钥扩展方案，24 bit的种子密钥能够相应得到。

复杂度分析　因为N=37.3，则攻击的数据复杂度为$2^{N+16}=2^{53.3}$个选择明文。攻击的时间复杂度分析如下：步骤3需要$2\times 2^8 \times 2^{N-9.2}=2^{37.1}$次1轮解密运算；步骤4需要$2\times 2^8 \times 2^{N-17.2}=2^{29.1}$次1轮解密运算；步骤5需要$2\times 2^8 \times 2^{N-25.2}=2^{21.1}$次1轮加密运算。因此，总共需要$(2^{37.1}+2^{29.1}+2^{21.1})/25 \approx 2^{32.5}$次25轮CSA-BC加密。对于攻击的存储复杂度，需要存储$2^{N-25.2}=2^{12.1}$明密文对和2^{24}个候选密钥，则攻击的存储复杂度约为2^{24}个存储单元。

5　本文结果和对比

本文利用构造的22轮不可能差分区分器，对CSA-BC进行了25轮的攻击。该攻击能够恢复24 bit种子密钥，其数据复杂度为$2^{53.3}$个选择明文，时间复杂度为$2^{32.5}$次25轮CSA-BC加密，存储复杂度为2^{24}个存储单元。表4给出了本文结果与已有最好结果的比较。从表4中可以看出，与已有最好不可能差分结果相比，本文结果将区分器轮数提高了2轮，不可能差分攻击轮数提高了4轮，攻击能够恢复的密钥量增加了8 bit。

表 4　本文结果与已有最好结果比较

区分器长度	攻击长度	恢复密钥量	数据复杂度	时间复杂度	存储复杂度	来源
20轮	21轮	16 bit	$2^{44.5}$	$2^{22.7}$	$2^{10.5}$	文献[6]
22轮	25轮	24 bit	$2^{53.3}$	$2^{32.5}$	2^{24}	本文

在文献[6]中，作者指出了CSA整体结构上存在的一个缺陷，并且利用该缺陷将21轮CSA-BC的不可能差分攻击扩展到21轮CSA整个算法的不可能差分攻击中。同样地，本文利用该缺陷也能够将25轮的CSA-BC的不可能差分攻击扩展到25轮CSA整个算法中。

6　结束语

本文主要研究了CSA-BC的不可能差分性质。首先，利用S盒的具体信息构造了一条22轮的不可能差分区分器，该区分器的长度比已有最好结果长2轮。然后，利用构造的22轮不可能差分区分器，攻击了25轮的CSA-BC。该攻击的数据复杂度为$2^{53.3}$个选择明文，时间复杂度为$2^{32.5}$次加密，存储复杂度为2^{24}个存储单元。到目前为止，本文的攻击结果是CSA关于不可能差分攻击的最好结果。

参 考 文 献

[1] WEINMANN R P and WIRT K. Analysis of the DVB common scrambling algorithm[C]. International Federation for Information Processing, Boston, USA, 2005: 195–207. doi: 10.1007/0-387-24486-7_15.

[2] WIRT K. Fault attack on the DVB common scrambling algorithm[C]. Computational Science and Its Applications, Singapore, 2005: 511–517. doi: 10.1007/11424826_61.

[3] SIMPSON L, HENRICKSEN M, and YAP W S. Improved cryptanalysis of the common scrambling algorithm stream cipher[C]. The 14th Australasian Conference on Information Security and Privacy, Brisbane, Australia, 2009: 108–121. doi: 10.1007/978-3-642-02620-1_8.

[4] TEWS E, WALDE J, and WEINER M. Breaking DVB-CSA[C]. West European Workshop on Research in Cryptography, Weimar, Germany, 2011: 41–45. doi: 10.1007/978-3-642-34159-5_4.

[5] ZHANG Kai and GUAN Jie. Distinguishing attack on common scrambling algorithm[J]. *The International Arab Journal of Information Technology*, 2015, 12(4): 410–414.

[6] ZHANG Kai, GUAN Jie, and HU Bin. Impossible differential cryptanalysis on DVB-CSA[J]. *KSII Transactions on Internet and Information Systems*, 2016, 10(3): 1944–1956. doi: 10.3837/tiis.2016.04.027.

[7] 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, China, 2014: 158–178. doi: 10.1007/978-3-662-45611-8_9.

[8] 李俊志, 关杰. 一种基于完全性的不可能差分区分器构造方法[J]. 电子与信息学报, 2018, 40(2): 430–437. doi: 10.11999/JEIT170422.

LI Junzhi and GUAN Jie. A method of constructing impossible differential distinguishers based on completeness[J]. *Journal of Electronics & Information Technology*, 2018, 40(2): 430–437. doi: 10.11999/JEIT170422.

[9] 徐洪, 苏鹏晖, 戚文峰. 减轮SPECK算法的不可能差分分析[J]. 电子与信息学报, 2017, 39(10): 2479–2486. doi: 10.11999/JEIT170049.

XU Hong, SU Penghui, and QI Wenfeng. Impossible differential cryptanalysis of reduced-round SPECK[J]. *Journal of Electronics & Information Technology*, 2017, 39(10): 2479–2486. doi: 10.11999/JEIT170049.

[10] 付立仕, 崔霆, 金晨辉. 嵌套SP网络的New-Structure系列结构的零相关线性逼近与不可能差分性质研究[J]. 电子学报, 2017, 45(6): 1367–1374. doi: 10.3969/j.issn.0372-2112.2017.06.013.

FU Lishi, CUI Ting, and JIN Chenhui. Zero correlation linear approximations and impossible differentials of New-Structure series with SP networks[J]. *Acta Electronica Sinica*, 2017, 45(6): 1367–1374. doi: 10.3969/j.issn.0372-2112.2017.06.013.

[11] SUN Bing, LIU Meicheng, GUO Jian, *et al*. Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis[C]. Advances in Cryptology – EUROCRYPT 2016, Vienna, Austrian, 2016: 196–213. doi: 10.1007/978-3-662-49890-3_8.

[12] SHEN Xuan, LI Ruilin, SUN Bing, *et al*. Dual relationship between impossible differentials and zero correlation linear hulls of SIMON-like ciphers[C]. Information Security Practice and Experience, Melbourne, Australia, 2017: 237–255. doi: 10.1007/978-3-319-72359-4_14.

[13] BOURA C, LALLEMAND V, PLASENCIA M N, *et al*. Making the impossible possible[J]. *Journal of Cryptology*, 2018, 31(1): 101–133. doi: 10.1007/s00145-016-9251-7.

[14] KNUDSEN L. DEAL-A 128-bit block cipher[R]. Department of Informatics, University of Bergen, Norway, 1998.

[15] BIHAM E, BIRYUKOV A, and SHAMIR A. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials[C]. Advances in Cryptology – EUROCRYPT 1999, Prague, Czech, 1999: 12–23. doi: 10.1007/3-540-48910-X2.

沈 璇：男，1990年生，博士生，研究方向为分组密码的安全性分析.
孙 兵：男，1981年生，讲师，研究方向为对称密码的设计与分析.
刘国强：男，1986年生，讲师，研究方向为对称密码的设计与分析.
李 超：男，1966年生，博士生导师，教授，研究方向为编码密码理论及其应用.