RAIN-128算法的中间相遇攻击

杜小妮*①② 郑亚楠① 梁丽芳① 李锴彬③
①(西北师范大学数学与统计学院 兰州 730070)
②(西北师范大学密码技术与数据分析重点实验室 兰州 730070)
③(西北师范大学计算机科学与工程学院 兰州 730070)

摘 要：RAIN是一族SPN结构的轻量级分组密码算法，该算法具有软硬件实现效率高、安全性强等特点。中间相遇攻击被广泛应用于分组密码算法的安全性分析中。该文通过分析RAIN-128的结构特性和截断差分特征，利用差分枚举技术分别构造了4轮和6轮中间相遇区分器，给出了8轮及10轮的中间相遇攻击。当攻击轮数为8轮时，预计算阶段的时间复杂度为2^{68}次8轮RAIN-128加密，存储复杂度为2^{75} bit，在线攻击阶段的时间复杂度为2^{109}次8轮加密，数据复杂度是2^{72}个选择明文；当攻击轮数为10轮时，预计算阶段的时间复杂度为2^{214}次10轮加密，存储复杂度为2^{219} bit，在线攻击阶段的时间复杂度为2^{109}次10轮加密，数据复杂度是2^{72}个选择明文，分析结果显示，RAIN-128可以抵抗中间相遇攻击，并具有较高的安全冗余。

关键词：分组密码；RAIN-128；中间相遇攻击；差分枚举技术
中图分类号：TN918.2; TP309.7 **文献标识码**：A **文章编号**：1009-5896(2024)01-0327-08
DOI: 10.11999/JEIT221593

Meet-in-the-middle Attack on RAIN-128

DU Xiaoni①② ZHENG Yanan① LIANG Lifang① LI Kaibin③
①(*College of Mathematics and Statistic, Northwest Normal University, Lanzhou* 730070, *China*)
②(*Key Laboratory of Cryptography and Data Analytics, Northwest Normal University, Lanzhou* 730070, *China*)
③(*College of Computer Science and Engineering, Northwest Normal University, Lanzhou* 730070, *China*)

Abstract: RAIN is a lightweight block cipher with SPN structure, which not only has strong security, but also possesses high software and hardware implementation efficiency. Meet-in-the-middle attacks are widely used in the security analysis of block ciphers algorithms. In this paper, the meet-in-the-middle attack on RAIN is researched. By examining the structural characteristics and the properties of truncated differential of RAIN-128, both 4-round and 6-round meet-in-the-middle distinguishers are first constructed by using differential enumeration technique, and meet-in-the-middle attacks on 8-round and 10-round RAIN-128 are presented, respectively. For 8-round attack, in the preprocessing, the time complexity is 2^{68} 8-round encryptions, and the memory complexity is 2^{75} bit, in the online, the time complexity is 2^{109} 8-round encryptions, and the data complexity is 2^{72} chosen plaintexts. For 10-round attack, in the preprocessing, the time complexity is 2^{214} 10-round encryptions, and the memory complexity is 2^{219} bit, in the online, the time complexity is 2^{109} 10-round encryptions, and the data complexity is 2^{72} chosen plaintexts. The result shows that RAIN-128 can be against meet-in-the-middle attack and has high security redundancy.

Key words: Block ciphers; RAIN-128; Meet-in-the-middle attack; Differential enumeration technique

1 引言

近年来，随着射频识别技术和无线传感器的发展与应用，为了保障这些资源受限设备中的信息安全，功耗低和软硬件实现面积小的轻量级分组密码算法被广泛应用。然而轻量级分组密码算法追求低功耗与高效率并存，这一矛盾会使算法的安全性无法得到保证，因此对轻量级分组密码算法进行安全性分析是非常有必要的。

1977年Diffie和Hellman[1]提出中间相遇攻击的思想，并将其应用到DES[2]的安全性分析中，之后

收稿日期：2023-01-04；改回日期：2023-04-12；网络出版：2023-04-17
*通信作者： 杜小妮 ymldxn@126.com
基金项目：国家自然科学基金(62172337)
Foundation Item: The National Natural Science Foundation of China (62172337)

广泛应用于Feistel结构的密码算法分析中；随后在FSE 2008上Demirci和Selçuk[3]利用文献[1]的思想，构造了AES[4]的4轮中间相遇区分器，首次提出了8轮AES-256的中间相遇攻击，是目前针对SPN结构分组密码最经典的中间相遇攻击，一般称之为Demirci-Selçuk中间相遇攻击(DS-MITM)；2010年，Dunkelman等人[5]利用差分枚举技术、多重集技术和密钥桥技术对文献[3]方法进行了改进，降低了复杂度；随后Derbez等人[6]通过搜索得到了大量减轮AES-256的高效路径，进一步降低了复杂度。在ASIACRYPT 2018上，Shi等人[7]首次将自动化搜索模型与算法的相关约束条件结合，实现了SKINNY[8]的22轮中间相遇攻击；2020年，Chen等人[9]利用密钥桥技术，使得密钥恢复攻击的复杂度有所降低；为了进一步降低算法的复杂度，肖钰汾等人[10]在2021年对SKINNY的中间相遇区分器进行自动化搜索时，将搜索过程中一部分状态的猜测转换为密钥的猜测，同时结合密钥桥技术，降低了密钥的猜测量和区分器的存储复杂度。

截断差分分析[11]是差分分析[12]的一个变体。与经典差分分析考虑具体差分值不同，截断差分只考虑差分的一部分性质，比如差分落在某个集合里，差分的某一位为0等。

RAIN是由曹梅春等人[13]于2021年提出的一种面向软硬件和门限实现的轻量级分组密码算法，并额外的调柄输入形成可调分组密码，增加了算法的灵活性。具有类似结构的算法还包括QARMA[14]，CRAFT[15]等。设计者从差分分析[12]、不可能差分分析[16]、积分分析[17]和不变子空间分析[18]4个方面对算法进行了安全性评估，结果显示，算法具有较大的安全冗余，在PC, ARM平台和FPGA平台上都具有出色的实现性能。

本文首次对RAIN-128进行中间相遇攻击，表1给出了8轮和10轮攻击的复杂度对比，主要贡献如下：

(1) 通过分析算法的结构特性和截断差分特征，利用差分枚举技术找到了4轮区分器，实现了8轮中间相遇攻击，使得预计算阶段的时间复杂度为P次8轮加密，存储复杂度为2^{75} bit。

(2) 类似地，利用(1)中的方法找到了6轮区分器，并实现了10轮中间相遇攻击。其中预计算阶段的时间复杂度为2^{214}次10轮加密，存储复杂度为2^{219} bit。

表 1 RAIN算法的8/10轮中间相遇攻击复杂度

轮数(r)	时间(预计算)	时间(在线)	数据	存储(bit)
8	2^{68}	2^{109}	2^{72}	2^{75}
10	2^{214}	2^{109}	2^{72}	2^{219}

(3) 利用算法的结构特性和等价密钥$MC^{-1}(K)$，减少了在线攻击阶段状态字节的猜测量，使得在线攻击阶段的时间复杂度为2^{109}次8/10轮加密，数据复杂度是2^{72}个选择明文。

本文结构安排如下：第2节对本文用到的符号进行说明，并简要介绍RAIN算法和DS-MITM攻击；第3节对RAIN-128进行8轮中间相遇攻击；第4节给出RAIN-128的10轮中间相遇攻击；第5节总结全文。

2 预备知识

2.1 符号说明

P：明文；

C：密文；

K：白化密钥；

MC：列混合；

SC：字节替换；

SR：行移位；

ATK_i：轮可调密钥；

V：$P\oplus K$；

$X_i^j[l]$：在第i轮中，经过列混合(MC)后第j个状态的第l个单元格；

$Y_i^j[l]$：在第i轮中，经过字节替换(SC)后第j个状态的第l个单元格；

$Z_i^j[l]$：在第i轮中，经过行移位(SR)后第j个状态的第l个单元格；

$W_i^j[l]$：在第i轮中，经过轮可调密钥加(ATK_i)后第j个状态的第l个单元格；

$\Delta X_i^0[l]$：$\{X_i^0[l]\oplus X_i^0[l],\ X_i^0[l]\oplus X_i^1[l],\cdots,\ X_i^0[l]\oplus X_i^{255}[l]\}$；

$\Delta Y_i^0[l]$：$\{Y_i^0[l]\oplus Y_i^0[l],\ Y_i^0[l]\oplus Y_i^1[l],\cdots,\ Y_i^0[l]\oplus Y_i^{255}[l]\}$；

$\Delta Z_i^0[l]$：$\{Z_i^0[l]\oplus Z_i^0[l],\ Z_i^0[l]\oplus Z_i^1[l],\cdots,\ Z_i^0[l]\oplus Z_i^{255}[l]\}$；

$\Delta W_i^0[l]$：$\{W_i^0[l]\oplus W_i^0[l],\ W_i^0[l]\oplus W_i^1[l],\cdots,\ W_i^0[l]\oplus W_i^{255}[l]\}$。

2.2 RAIN算法介绍

RAIN[13]是一族轻量级分组密码算法，分组长度支持64 bit和128 bit，对应的密钥长度为64 bit和128 bit，迭代轮数为30轮和36轮，分别记为RAIN-64和RAIN-128。

算法的轮函数由列混合、 字节替换、行移位和轮可调密钥加4种运算构成。r轮($r=30/36$)算法的整体结构如图1所示，第i轮的轮函数结构如图2所示。将64/128 bit的明文、中间状态、密文统称为密码状态。令$M=m_0||m_1||\cdots||m_{15}$表示一个密码

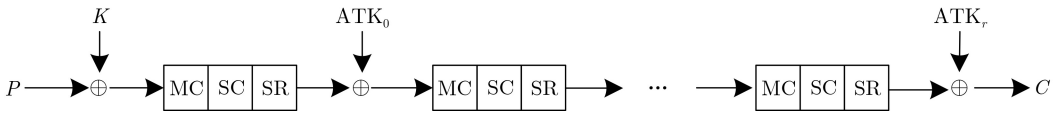

图 1 RAIN算法的整体结构

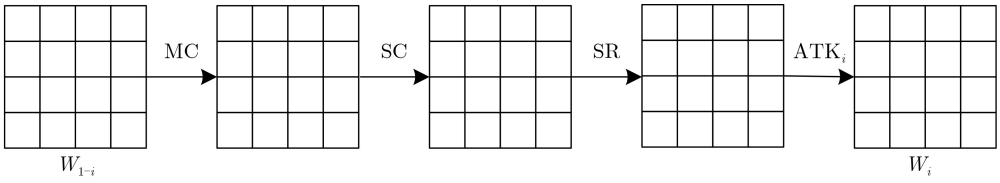

图 2 RAIN算法的轮函数结构

状态，对于$1 \le s \le 15$，当分组长度为64 bit时，m_s的长度是4 bit，即半字节；当分组长度为128 bit时，m_s的长度是8 bit，即一个字节。按照行的顺序，依次将密码状态输入到4×4的矩阵中，其矩阵形式为

$$\boldsymbol{M} = \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 & m_7 \\ m_8 & m_9 & m_{10} & m_{11} \\ m_{12} & m_{13} & m_{14} & m_{15} \end{bmatrix}$$

RAIN-128的加密流程，具体如下：

(1) 初始白化：将明文P与白化密钥K按字节进行异或操作。

(2) 列混合：密码状态左乘矩阵

$$\boldsymbol{U} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

(3) 字节替换：对密码状态中的16 Byte分别进行S盒替换，采用8 bit的S盒。

(4) 行移位：采用与AES相同的行移位变换，即密码状态的第2～4行分别向左循环移位1 Byte, 2 Byte, 3 Byte。

(5) 轮可调密钥加：将轮可调密钥与步骤(4)的输出进行异或操作。

2.3 DS-MITM攻击

在DS-MITM的分析过程中，将r轮加密算法E分解为如图3所示的E_0, E_1和E_2，即$E = E_2 \circ E_1 \circ E_0$。其中$E_1$为加密算法的第$r_0$轮至第$r_0 + r_1$轮，且为构建的区分器。$E_0$为加密算法的第0轮至第$r_0 - 1$轮，$E_2$为加密算法的第$r_0 + r_1 + 1$轮至第$r - 1$轮。

定义1 (δ-集)[3] 若RAIN-128的1 Byte遍历256个值，其它15 Byte取固定值，这样的结构被称为δ-集。其中，遍历256个值的字节为活跃字节，其余为非活跃字节。

在图3中，对于区分器E_1，首先需确定δ-集活跃字节的位置和区分器输出字节的位置；其次根据δ-集在第r_0轮至第$r_0 + r_1$轮的差分传播路径，以及输出字节在第$r_0 + r_1$轮至第r_0轮的差分传播路径，猜测两条路径在每一轮输入相交的状态值，穷尽所有相交状态值，在第$r_0 + r_1$轮可得到不同的输出状态/差分集合，将集合存储在哈希表中，构成中间相遇区分器。对于E_0，需确定在第$r_0 + r_1$轮形成δ-集的明文集，并猜测部分轮密钥；然后将明文集加密r轮生成密文，猜测E_2的部分轮密钥，对密文进行解密，求得第$r_0 + r_1$轮固定字节的输出状态/差分，判断状态/差分值是否在区分器的哈希表中；若存在，则判断E_0和E_2中猜测的密钥为正确密钥，否则为错误密钥，再通过穷举搜索筛选出正确密钥。

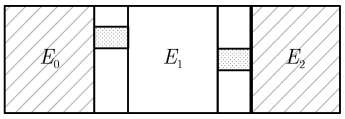

图 3 DS-MITM模型

3 RAIN-128的8轮中间相遇攻击

本节首先介绍RAIN-128的4轮中间相遇区分器，其次利用该区分器实现了8轮攻击，最后分析了复杂度。

3.1 4轮中间相遇区分器

性质1 (S盒性质)[6] 给定S盒的非零输入差分Δ_{in}，非零输出差分Δ_{out}，方程$S(x) \oplus S(x \oplus \Delta_{\text{in}}) = \Delta_{\text{out}}$平均有一个解。

定理1 设$u[i]$为活跃字节，$\{W_4^0[0], W_4^1[0], \cdots, W_4^{255}\}$是$\{Z_0^0[0], Z_0^1[0], \cdots, Z_0^{255}[0]\}$经过4轮RAIN-128加密的输出。若存在明文对$(Z_0^0, Z_0^j)$ $(0 \le j \le 255)$满足图4的截断差分特征，则输出差分$\{W_4^0[0] \oplus W_4^0[0], W_4^0[0] \oplus W_4^1[0], \cdots, W_4^0[0] \oplus W_4^{255}[0]\}$完全由以下11 Byte确定：

$$X_1^0[4, 8, 12],\ X_2^0[1, 2, 3, 9],\ X_3^0[5, 10, 15],\ X_4^0[0].$$

证明 (1) 由$Z_0[0]$为活跃字节得，δ-集$\{Z_0^0[0],$

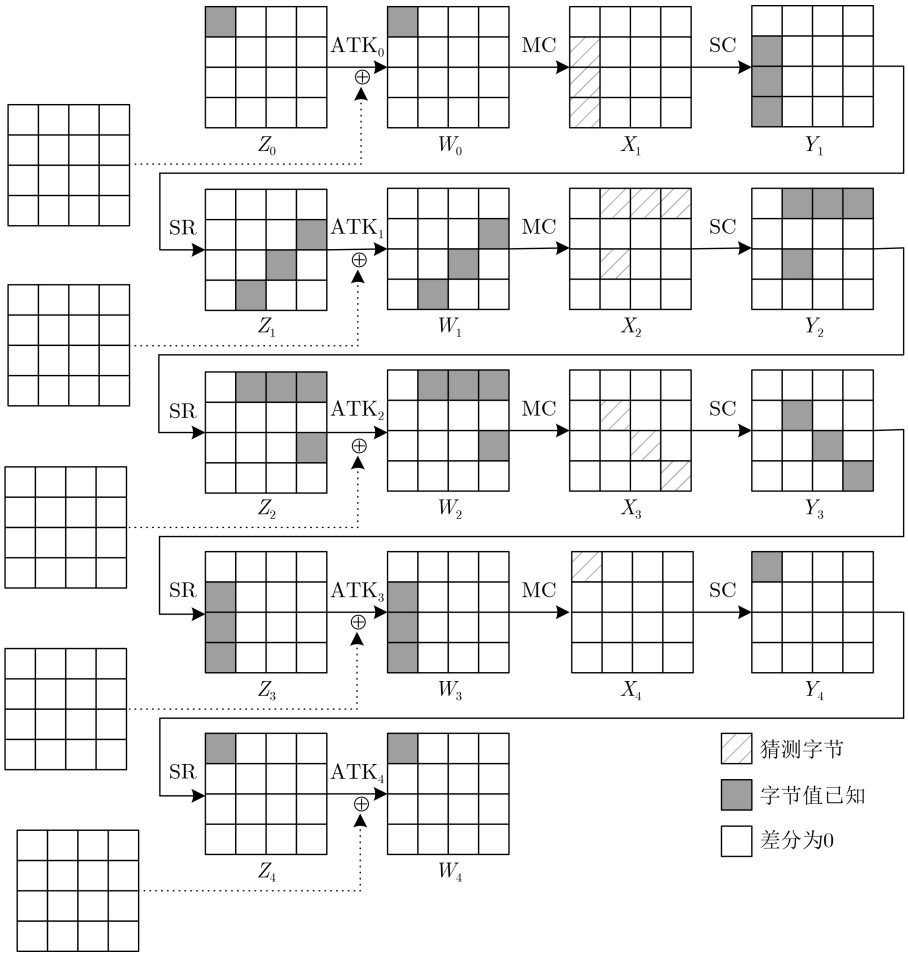

图 4 4轮RAIN-128算法的中间相遇区分器

$Z_0^1[0]$,…, $Z_0^{255}[0]$}在第0字节活跃，经过轮可调密钥加和列混合两个线性操作，可计算出差分集合{$X_1^0 \oplus X_1^0$, $X_1^0 \oplus X_1^1$,…, $X_1^0 \oplus X_1^{255}$} 的 第4, 8, 12 Byte的值。

(2) 假设$X_1^0[4,\ 8,\ 12]$已知，根据(1)得到的值，可计算出{X_1^0 , X_1^1 ,…, X_1^{255}}的第4, 8, 12 Byte的值。经过S盒和行移位操作后，计算出{Z_1^0 , Z_1^1 ,…, Z_1^{255}}的第7, 10, 13 Byte的值，从而得到差分集合$\Delta Z_1^0[l]$ ($l = 7,\ 10,\ 13$)的值。由于轮可调密钥加和列混合是线性操作，故可得差分集合$\Delta X_2^0[l]$ ($l = 1,\ 2,\ 3,\ 9$)的值。

(3) 假设$X_2^0[1,\ 2,\ 3,\ 9]$已知，根据(2)得到的值，可计算出{X_2^0 , X_2^1 ,…, X_2^{255}} 的 第1, 2, 3, 9 Byte的值。经过S盒和行移位操作后，可计算出{Z_2^0 , Z_2^1 ,…, Z_2^{255}}的第1, 2, 3, 11 Byte的值，从而得到差分集合$\Delta Z_2^0[l]$ ($l = 1,\ 2,\ 3,\ 11$)的值。由于轮可调密钥加和列混合是线性操作，故可得差分集合$\Delta X_3^0[l]$ ($l = 5,\ 10,\ 15$)的值。

(4) 假设$X_3^0[5,\ 10,\ 15]$已知，同理，利用(3)得到的差分集合可计算输出差分$\Delta X_4^0[0]$。

(5) 假设$X_4^0[0]$已知，同理，利用(4)得到的差分集合可得输出差分$\Delta W_4^0[0]$。 证毕

为了进一步减少定理1中状态字节的猜测量，我们使用差分枚举技术给出如下结论。

定理2 若参数满足定理1的条件，且$X_1^0[4] = 0$，则差分集合$\Delta W_4^0[0]$可分别由以下字节确定：

(1) $X_1^0[8,\ 12]$, $X_2^0[1,\ 2,\ 3,\ 9]$, $X_3^0[5,\ 10,\ 15]$, $X_4^0[0]$;

(2) $\Delta Z_0^0[0]$, $X_1^0[8,\ 12]$, $Y_3^0[5,\ 10,\ 15]$, $Y_4^0[0]$, $\Delta Z_4^0[0]$.

证明 (1)由于δ-集{$Z_0^0[0]$, $Z_0^1[0]$,…, $Z_0^{255}[0]$}根据$Z_0^0[0]$的选择可以表示为2^8个差分集合，故可通过确定$Z_0^0[0]$使得$X_1^0[4] = 0$来减少猜测的字节数，再结合定理1的证明可得差分集合$\Delta W_4^0[0]$由$X_1^0[8,\ 12]$, $X_2^0[1,\ 2,\ 3,\ 9]$, $X_3^0[5,\ 10,\ 15]$, $X_4^0[0]$这10 Byte决定。

(2) 已知$\Delta Z_0^0[0]$，可推得$\Delta X_1^0[8,\ 12]$，而$\Delta Y_1^0[8,\ 12]$可由$X_1^0[8,\ 12]$和$\Delta X_1^0[8,\ 12]$推得；根据得到的$\Delta Y_1^0[8,\ 12]$，可计算出$\Delta X_2^0[1,\ 2,\ 3,\ 9]$；已知$\Delta Z_4^0[0]$，可推得$\Delta Y_4^0[0]$，根据$Y_4^0[0]$和$\Delta Y_4^0[0]$可推得$\Delta X_4^0[0]$，同理可得$\Delta Y_2^0[1,\ 2,\ 3,\ 9]$。

根据性质1，$X_2^0[1,\ 2,\ 3,\ 9]$可由$\Delta X_2^0[l]$和$\Delta Y_2^0[l]$ $(l=1,\ 2,\ 3,\ 9)$推得。

综上所述，差分集合$\Delta W_4^0[0]$由以上8 Byte决定。证毕

3.2 8轮RAIN-128的中间相遇攻击

8轮RAIN-128的中间相遇攻击由两个阶段组成：预计算阶段和在线攻击阶段。攻击过程如图5所示。

预计算阶段：由定理2，差分集合$\Delta W_4^0[0]$由8 Byte决定，因此最多有$2^{8\times 8}=2^{64}$种取值，将2^{64}个差分集合存储在哈希表T中。

在线攻击阶段：

(1) 选择$2^{8\times 9}=2^{72}$个明文，这些明文满足在第0, 4, 6, 8, 9, 12, 15 Byte处差分为0，其余9 Byte穷尽即可；

(2) 由于列混合与密钥加是线性操作，故可交换这两个操作，具体操作为：

(a) 用等价密钥$\mathrm{MC}^{-1}(K)$异或密码状态；

(b) 对(a)中的状态进行列混合(MC)操作。

记$u[i]=\mathrm{MC}^{-1}(K[i])$，则只需猜测$u[5]$, $u[10]$, $u[15]$和$\mathrm{ATK}_0[4,\ 8,\ 12]$，筛选一对明文，使得这对明文在第2轮输入状态差分仅在第0 Byte非0，取其中之一记为P_0；

(3) 根据δ-集的性质，以及上述猜测的轮密钥和P_0，可反解P_1，P_2，…，P_{255}。对P_0，P_1，…，P_{255}进行8轮加密，得到相应密文；

(4) 猜测$\Delta W_5^0[0]$和$X_6^0[4,\ 8,\ 12]$。$\Delta W_5^0[0]$可推出$\Delta X_6^0[4,\ 8,\ 12]$，结合$X_6^0[4,\ 8,\ 12]$的值，可计算出$\{X_6^0[l],\ X_6^1[l],\cdots,\ X_6^{255}[l]\}$ $(l=4,\ 8,\ 12)$，进而得到$\{Y_6^0[l]$，$Y_6^1[l]$，…，$Y_6^{255}[l]\}$ $(l=4,\ 8,\ 12)$，即可得到$\Delta Y_6^0[4,\ 8,\ 12]$，经过行移位操作后，得到$\Delta Z_6^0[7,\ 10,\ 13]$。由于列混合和密钥加是线性操作，故可得$\Delta X_7^0[1,\ 2,\ 3,\ 5,\ 7,\ 9,\ 10,\ 14,\ 15]$。根据密文差分推得$\Delta Y_7^0[1,\ 2,\ 3,\ 5,\ 7,\ 9,\ 10,\ 14,\ 15]$，运用性质1可计算出$Y_7^0[1,\ 2,\ 3,\ 5,\ 7,\ 9,\ 10,\ 14,\ 15]$，最终推出$\mathrm{ATK}_7[1,\ 2,\ 3,\ 5,\ 7,\ 9,\ 10,\ 14,\ 15]$。

(5) 猜测$\mathrm{ATK}_6[7,\ 10,\ 13]$。

(6) 利用上述密钥值部分解密第 (3) 步得到的密文，可得$W_5[0]$，计算差分集$\Delta W_5^0[0]$。判断差分集是否在预计算阶段建立的哈希表T中。若在，则猜测的密钥为正确密钥，否则为错误密钥。

(7) 通过穷尽搜索的方式筛选出正确密钥。

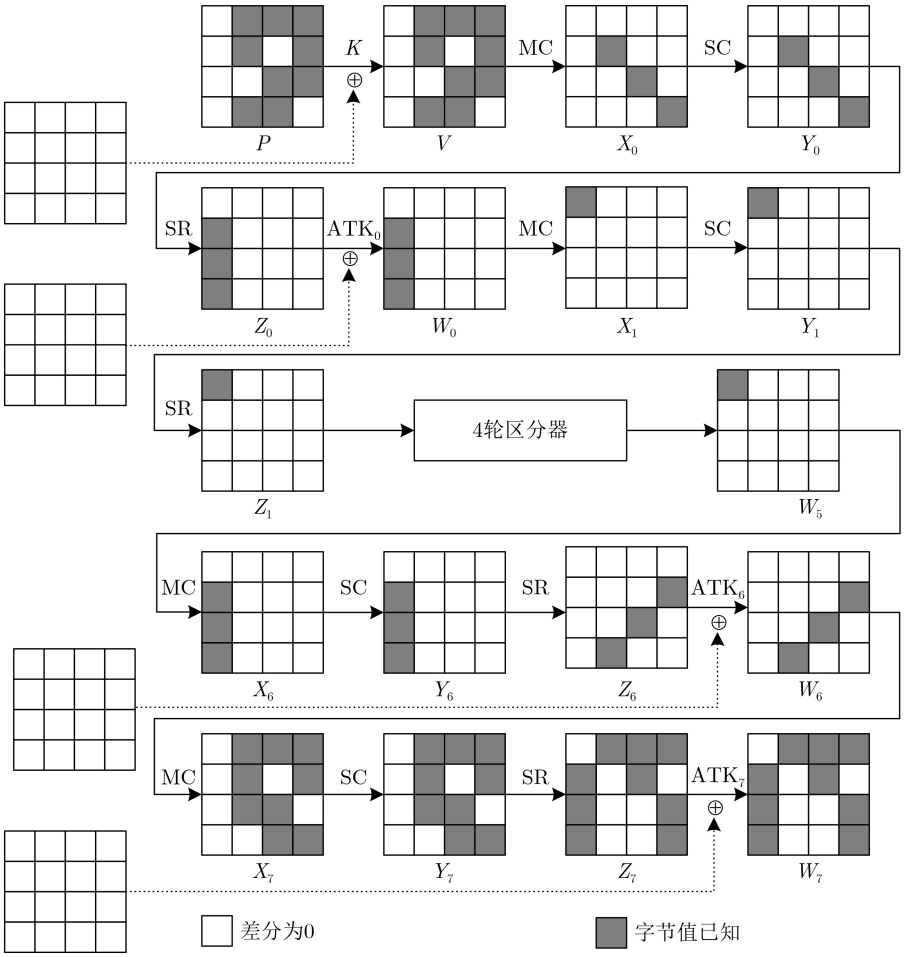

图 5 8轮RAIN-128算法的中间相遇攻击

上述攻击步骤中，需要猜测13 Byte，得到18 Byte的密钥：$u[5]$, $u[10]$, $u[15]$, $\mathrm{ATK}_0[4, 8, 12]$, $\mathrm{ATK}_6[7, 10, 13]$, $\mathrm{ATK}_7[1, 2, 3, 5, 7, 9, 10, 14, 15]$。

3.3 攻击复杂度分析

预计算阶段需猜测8 Byte，计算$2^{64}\times 2^8=2^{72}$个差分值，因此预计算的时间复杂度为$2^{64}\times 2^8\times 8/(8\times 16)=2^{68}$次加密8轮RAIN-128，存储复杂度为$(2^8-1)\times 8\times 2^{8\times 8}\approx 2^{75}$ bit。

在线攻击阶段选择明文量为2^{72}个明文，即数据复杂度为2^{72}个选择明文。在线攻击阶段需猜测13 Byte，因此在线攻击的时间复杂度为$2^{104}\times 2^8\times 16/(8\times 16)=2^{109}$次加密8轮算法。

4 10轮RAIN-128的中间相遇攻击

本节实现了RAIN-128 的10轮中间相遇攻击，因攻击过程与第3节类似，故仅给出结论，不再给出证明过程。

首先给出6轮中间相遇区分器的相关结论，区分器如图6所示。

定理3 设$Z_0[0]$为活跃字节，$\{W_6^0[0]$，$W_6^1[0]$，

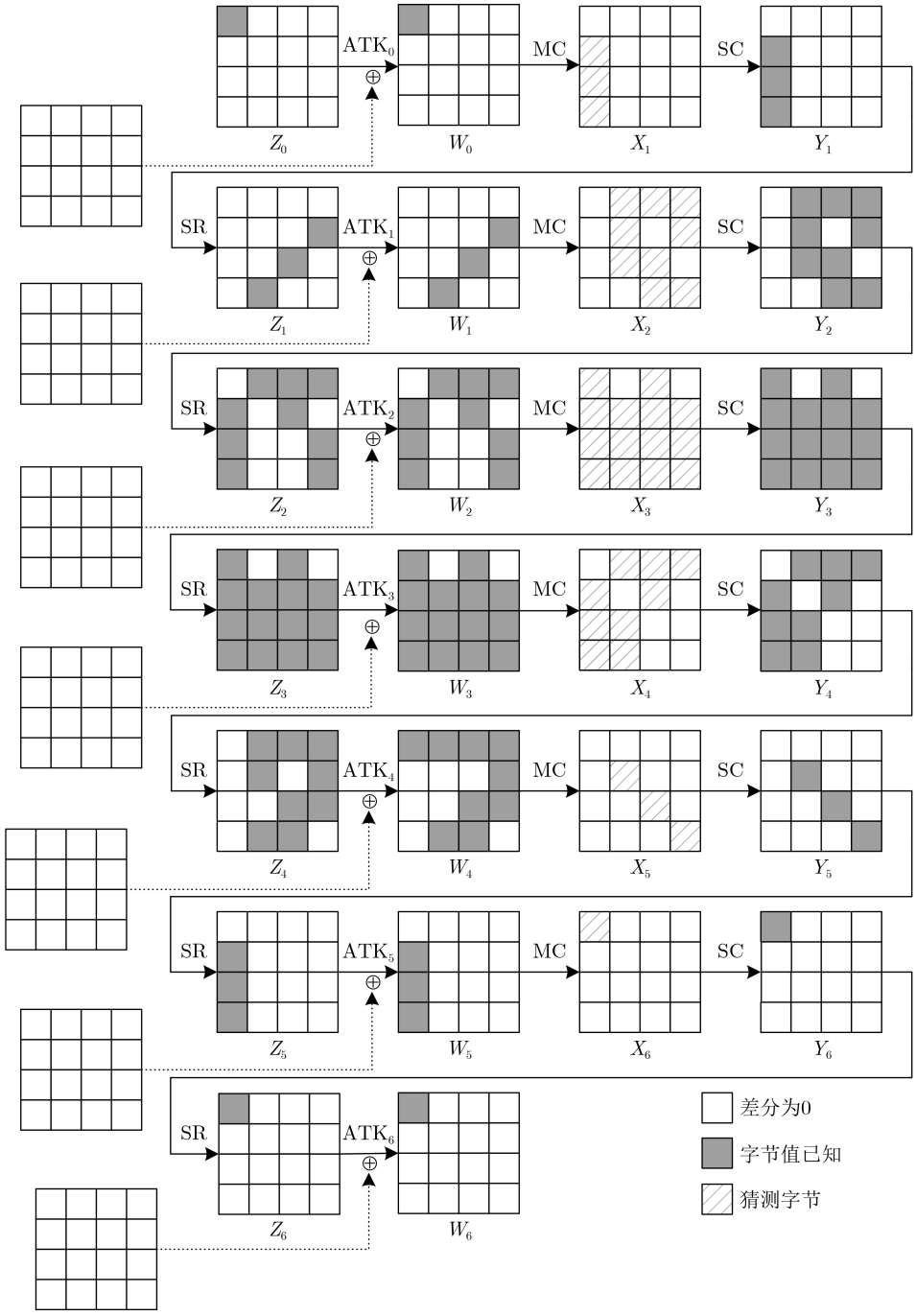

图 6 6轮RAIN-128算法的中间相遇区分器

$\cdots$, $W_6^{255}\}$是$\{Z_0^0[0], Z_0^1[0], \cdots, Z_0^{255}[0]\}$经过6轮RAIN-128加密的输出，若存在明文对$(Z_0^0, Z_0^j)$ $(0 \le j \le 255)$满足图6的截断差分特征，则输出差分$\Delta W_6^0[0]$由以下39 Byte确定：

$X_1[4, 8, 12]$, $X_2[1, 2, 3, 5, 7, 9, 10, 14, 15]$, $X_3[l]$ $(0 \le l \le 15$ 且$l \ne 1, 3)$,

$X_4[1, 2, 3, 4, 6, 8, 9, 12, 13]$, $X_5[5, 10, 15]$, $X_6[0]$

定理4　若参数满足定理3的条件，且$X_1^0[4] = 0$，则差分集合$\Delta W_6^0[0]$可分别由以下字节确定：

(1) $X_1[8, 12]$, $X_2[1, 2, 3, 5, 7, 9, 10, 14, 15]$, $X_3[l]$ $(0 \le l \le 15$ 且$l \ne 1, 3)$, $X_4[1, 2, 3, 4, 6, 8, 9, 12, 13]$, $X_5[5, 10, 15]$, $X_6[0]$;

(2) $\Delta Z_0^0[0]$, $X_1[8, 12]$, $X_2[1, 2, 3, 5, 7, 9, 10, 14, 15]$,$X_4[1, 2, 3, 4, 6, 8, 9, 12, 13]$,$X_5[5, 10, 15]$, $X_6[0]$, $\Delta Z_6^0[0]$

10轮中间相遇攻击过程及复杂度分析如下。

预计算阶段：由定理4，预计算阶段差分集合$\Delta W_6^0[0]$由26 Byte决定，因此最多有$2^{8\times 26} = 2^{208}$种取值，将2^{208}个差分集合存储在哈希表T中。此时时间复杂度为$2^{208} \times 2^8 \times 26/(10 \times 16) \approx 2^{214}$次加密10轮RAIN-128，存储复杂度为$(2^8 - 1) \times 8 \times 2^{8\times 26} \approx 2^{219}$ bit。

在线攻击阶段：需要猜测13 Byte。利用算法的结构特性和等价密钥$\mathrm{MC}^{-1}(K)$可推导出18 Byte的密钥：$u[5]$, $u[10]$, $u[15]$, $\mathrm{ATK}_0[4, 8, 12]$, $\mathrm{ATK}_8[7, 10, 13]$, $\mathrm{ATK}_9[1, 2, 3, 5, 7, 9, 10, 14, 15]$。利用所得密钥值对密文进行部分解密，可得$W_7[0]$，计算差分集$\Delta W_7^0[0]$。若差分集在预计算阶段建立的哈希表$T$中，则猜测的密钥为正确密钥，否则为错误密钥，并通过穷尽搜索的方式筛选出正确密钥。因此10轮RAIN-128所需数据复杂度为2^{72}个选择明文，时间复杂度为$2^{104} \times 2^8 \times 16/(10 \times 16) \approx 2^{109}$次加密10轮算法。

5　结束语

本文将DS-MITM模型与 RAIN-128的一类截断差分特征相结合，构造了RAIN-128的4轮和6轮区分器，并利用该算法的结构特性和差分枚举技术，实现了8轮和10轮的中间相遇攻击。用等价密钥$\mathrm{MC}^{-1}(K)$和算法的结构特性减少了在线攻击需猜测的密钥量，降低了在线攻击阶段的时间复杂度。在攻击的过程中如何减少状态字节的猜测量仍需进一步研究。

参 考 文 献

[1] DIFFIE W and HELLMAN M E. Special feature exhaustive cryptanalysis of the NBS data encryption standard[J]. *Computer*, 1977, 10(6): 74–84. doi: 10.1109/C-M.1977.217750.

[2] National Institute of Standards and Technology. FIPS 46–3 Data encryption standard (DES)[S]. National Institute of Standards and Technology, 1999.

[3] DEMIRCI H and SELÇUK A A. A meet-in-the-middle attack on 8-round AES[C]. Proceedings of the 15th International Workshop on Fast Software Encryption, Lausanne, Switzerland, 2008: 116–126. doi: 10.1007/978-3-540-71039-4_7.

[4] DAEMEN J and RIJMEN V. The Design of Rijndael: AES -The Advanced Encryption Standard[M]. Berlin: Springer, 2002: 137–139. doi: 10.1007/978-3-662-04722-4.

[5] DUNKELMAN O, KELLER N, and SHAMIR A. Improved single-key attacks on 8-round AES-192 and AES-256[J]. *Journal of Cryptology*, 2015, 28(3): 397–422. doi: 10.1007/s00145-013-9159-4.

[6] DERBEZ P and FOUQUE P A. Exhausting Demirci-Selçuk meet-in-the-middle attacks against reduced-round AES[C]. Proceedings of the 20th International Workshop on Fast Software Encryption, Singapore, 2013: 541–560. doi: 10.1007/978-3-662-43933-3_28.

[7] SHI Danping, SUN Siwei, DERBEZ P, *et al*. Programming the Demirci-Selçuk meet-in-the-middle attack with constraints[C]. Proceedings of the 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, Australia, 2018: 3–34. doi: 10.1007/978-3-030-03329-3_1.

[8] 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, Santa Barbara, USA, 2016: 123–153. doi: 10.1007/978-3-662-53008-5_5.

[9] CHEN Qiu, SHI Danping, SUN Siwei, *et al*. Automatic Demirci-Selçuk meet-in-the-middle attack on SKINNY with key-bridging[C]. Proceedings of the 21st International Conference on Information and Communications Security, Beijing, China, 2020: 233–247. doi: 10.1007/978-3-030-41579-2_14.

[10] 肖钰汾, 田甜. 减轮SKINNY-128-384算法的中间相遇攻击[J]. 密码学报, 2021, 8(2): 338–351. doi: 10.13868/j.cnki.jcr.000442.

XIAO Yufen and TIAN Tian. Meet-in-the-Middle attack on round-reduced SKINNY-128-384[J]. *Journal of Cryptologic Research*, 2021, 8(2): 338–351. doi: 10.13868/j.cnki.jcr.000442.

[11] SUGITA M, KOBARA K, and IMAI H. Security of reduced version of the block cipher Camellia against truncated and

impossible differential cryptanalysis[C]. Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, 2001: 193–207. doi: 10.1007/3-540-45682-1_12.

[12] BIHAM E. Cryptanalysis of Patarin's 2-round public key system with S boxes (2R)[C]. Proceedings of 2000 International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 2000: 408–416. doi: 10.1007/3-540-45539-6_28.

[13] 曹梅春, 张文英, 陈彦琴, 等. RAIN: 一种面向软硬件和门限实现的轻量分组密码算法[J]. 计算机研究与发展, 2021, 58(5): 1045–1055. doi: 10.7544/issn1000-1239.2021.20200933.
CAO Meichun, ZHANG Wenying, CHEN Yanqin, *et al*. RAIN: A lightweight block cipher towards software, hardware and threshold implementations[J]. *Journal of Computer Research and Development*, 2021, 58(5): 1045–1055. doi: 10.7544/issn1000-1239.2021.20200933.

[14] AVANZI R. The QARMA block cipher family. Almost MDS matrices over rings with Zero Divisors, Nearly Symmetric Even-mansour constructions with Non-involutory central rounds, and search heuristics for low-latency S-Boxes[J]. *IACR Transactions on Symmetric Cryptology*, 2017, 2017(1): 4–44. doi: 10.13154/tosc.v2017.i1.4-44.

[15] 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. doi: 10.13154/tosc.v2019.i1.5-45.

[16] 蒋梓龙, 金晨辉. Saturnin算法的不可能差分分析[J]. 通信学报, 2022, 43(3): 53–62. doi: 10.11959/j.issn.1000-436x.2022045.
JIANG Zilong and JIN Chenhui. Impossible differential cryptanalysis of Saturnin algorithm[J]. *Journal on Communications*, 2022, 43(3): 53–62. doi: 10.11959/j.issn.1000-436x.2022045.

[17] 叶涛, 韦永壮, 李灵琛. ACE密码算法的积分分析[J]. 电子与信息学报, 2021, 43(4): 908–914. doi: 10.11999/JEIT200234.
YE Tao, WEI Yongzhuang, and LI Lingchen. Integral cryptanalysis of ACE encryption algorithm[J]. *Journal of Electronics & Information Technology*, 2021, 43(4): 908–914. doi: 10.11999/JEIT200234.

[18] LEANDER G, ABDELRAHEEM M A, ALKHZAIMI H, *et al*. A cryptanalysis of PRINTCIPHER: The invariant subspace attack[C]. Proceedings of the 31st Annual Cryptology Conference, Santa Barbara, USA, 2011: 206–221. doi: 10.1007/978-3-642-22792-9_12.

杜小妮：女，博士后，教授，研究方向为应用密码学.
郑亚楠：女，硕士生，研究方向为应用密码学.
梁丽芳：女，硕士生，研究方向为应用密码学.
李锴彬：男，硕士生，研究方向为分组密码.

责任编辑：陈 倩