

改进的Joltik-BC-128算法的中间相遇攻击

李征, 任炯炯, 李曼曼, 陈少真
(信息工程大学, 河南 郑州 450001)

摘要: Joltik-BC是在2014年亚密会上发布的一种轻量级可调分组密码算法, 该算法采用替换-置换网络(SPN, substitution-permutation network)结构和Tweakey框架。通过研究Joltik-BC算法的内部特征, 利用控制调柄差分方法, 并结合差分枚举技术和S盒的差分特性, 构造了6轮Joltik-BC-128算法的中间相遇区分器。利用此区分器改进了9轮Joltik-BC-128算法的中间相遇攻击结果。改进后的9轮Joltik-BC-128算法的中间相遇攻击的存储复杂度为 $2^{44.91}$ 个长度为64的比特块, 时间复杂度为 2^{48} 次9轮Joltik-BC-128加密。相比已有中间相遇攻击结果, 该算法的时间复杂度和存储复杂度均显著降低。

关键词: 分组密码; Joltik-BC; 中间相遇攻击; 差分枚举

中图分类号: TN918.1

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2024073

Improved meet-in-the-middle attack on Joltik-BC-128

LI Zheng, REN Jiongjiong, LI Manman, CHEN Shaozhen
Information Engineering University, Zhengzhou 450001, China

Abstract: The Joltik-BC, adopting the substitution-permutation network structure and Tweakey framework, was a lightweight tweakable block cipher published at ASIACRPYPT 2014. By researching the internal characteristic of the Joltik-BC, a 6-round meet-in-the-middle distinguisher against the Joltik-BC-128 was constructed by controlling the tweakey differentials and combining differential enumeration and differential characteristics of S-boxes. An improved meet-in-the-middle attack against the 9-round Joltik-BC-128 was developed using this distinguisher. The memory and time complexities of the improved 9-round Joltik-BC-128 were $2^{44.91}$ 64-bits blocks and 2^{48} 9-round Joltik-BC-128 encryptions. Compared with existing meet-in-the-middle attack results, the time complexity and memory complexity of this method were significantly reduced.

Keywords: block cipher, Joltik-BC, meet-in-the-middle attack, differential enumeration

0 引言

近年来, 物联网应用得到了广泛普及, 在此

背景下轻量级分组密码被提出, 用于保障物联网应用的安全。

收稿日期: 2024-01-16; 修回日期: 2024-07-30

通信作者: 任炯炯, jiongjiong_fun@163.com

基金项目: 国家自然科学基金(62206312)

Foundation Item: The Natural Science Foundation of China (62206312)

引用格式: 李征, 任炯炯, 李曼曼, 等. 改进的Joltik-BC-128算法的中间相遇攻击[J]. 网络与信息安全学报, 2024, 10(5): 145-151.

Citation Format: LI Z, REN J J, LI M M, et al. Improved meet-in-the-middle attack on Joltik-BC-128[J]. Chinese Journal of Network and Information Security, 2024, 10(5): 145-151.

Jean 等^[1]在 2014 AsiaCrypt 会议上提出了轻量级可调分组密码 Joltik-BC 算法。作为 CAESAR^[2] 的胜选算法, Joltik-BC 的安全性一直是人们关注的重点。该算法有 2 个版本, 分别为 Joltik-BC-128 和 Joltik-BC-192。目前, Joltik-BC-128 算法的安全性已经过多种攻击方法的评估, 其中较有效的有不可能差分^[3]和中间相遇^[4]等攻击方法。2018 年, Zong 等^[5]对 9 轮 Joltik-BC-128 算法进行了不可能差分攻击, 攻击的时间复杂度为 $2^{61.7}$ 、存储复杂度为 2^{50} 、数据复杂度为 2^{60} 。2019 年, Li 等^[6]对 8 轮 Joltik-BC-128 算法进行了中间相遇攻击, 攻击的时间复杂度为 $2^{53.6}$ 、存储复杂度为 2^{53} 、数据复杂度为 $2^{53.5}$ 。2019 年, Liu 等^[7]针对 9 轮 Joltik-BC-128 算法给出了改进的中间相遇攻击, 攻击的时间复杂度为 $2^{56.6}$ 、存储复杂度为 $2^{52.91}$ 、数据复杂度为 2^{53} 。2021 年, Li 等^[8]利用差分枚举和控制调柄等技术改进了 9 轮 Joltik-BC-128 算法的中间相遇攻击结果, 攻击的时间复杂度为 $2^{54.1}$ 、存储复杂度为 $2^{52.91}$ 、数据复杂度为 2^{53} 。

中间相遇攻击是一种较有效的分析手段, 主要包括两个阶段, 即预计算阶段和在线处理阶段。预计算阶段的主要工作是根据区分器来构造哈希表, 在线处理阶段的主要工作是通过猜测相关密钥来检测所求的有序序列是否存在于预计算阶段构造的哈希表。如何提升 Joltik-BC 算法的中间相遇攻击效果, 是本文的工作重点。

本文从 Joltik-BC-128 算法的结构特点出发, 通过控制特定位置的调柄差分, 使内部状态差分与调柄差分充分重合, 构造了一个新的 6 轮 Joltik-BC-128 算法的中间相遇区分器, 并利用差分枚举和 S 盒 (substitution-box) 的差分特性等减少预计算阶段和在线处理阶段的字节猜测量, 有效降低攻击的时间复杂度, 从而提升 9 轮 Joltik-BC-128 算法的中间相遇攻击效果。改进的 9 轮 Joltik-BC-128 算法的中间相遇攻击的时间复杂度、存储复杂度、数据复杂度分别为 2^{48} 、 $2^{44.91}$ 和 2^{53} 。相比已有 Joltik-BC-128 算法的中间相遇攻击结果, 该方法的时间复杂度和存储复杂度均显著降低。

1 预备知识

1.1 符号表示

本文使用的符号及说明如表 1 所示。

表 1 符号及说明

符号	说明
P, C	明文, 密文
K, T	主密钥, 主调柄
f	轮函数
h	一个字节替换操作
RC	轮常数
$\oplus, \parallel$	异或, 级联
$x_i[j]$	x 第 i 轮的 j 个单元的值
$x_i^k[j]$	一系列 $x_i[j]$ 的第 k 个取值
x_i, y_i, z_i, w_i	在第 i 轮的单元代换、行移位、列混合和可调轮密钥加操作之前的中间状态值

1.2 Joltik-BC-128 算法介绍

Joltik-BC 算法采用了经典的替换-置换网络 (SPN, substitution-permutation network) 结构, 并引入了调柄, 主体加密流程与 AES 算法^[9]十分相似。本文的攻击只针对 Joltik-BC-128 算法。Joltik-BC-128 算法的总轮数为 24 轮, 分组长度、密钥长度和调柄长度均为 64 bit。其明文、密文及中间状态可以表示为 4×4 的矩阵, 共 16 个单元格, 其中每个单元格代表半个字节 (4 bit), 具体表示如下。

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

轮函数 f 包含下列 4 种运算。

1) 字节代换 (SB): 将每单元格 4 bit 的内部状态使用 Piccolo 的 4 bit 非线性 S 盒进行代换。

2) 行移位 (SR): 将第 i 行 4 个单元格分别进行循环左移 $i - 1$ 位的操作, 移动形式如下所示。

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

SR →

0	4	8	12
5	9	13	1
10	14	2	6
15	3	7	11

3) 列混合 (MC): 将内部状态矩阵左乘 1 个 4×4 MDS 矩阵, 矩阵 M 如下所示。

$$M = \begin{pmatrix} 1 & 4 & 9 & 13 \\ 4 & 1 & 13 & 9 \\ 8 & 13 & 1 & 4 \\ 13 & 9 & 4 & 1 \end{pmatrix} = M^{-1}$$

4) 可调轮密钥加：将内部状态、更新后的轮子密钥和调柄三者进行异或运算。

Joltik-BC-128算法的整体框架和加密流程如图1所示。

1.3 密钥扩展算法

在Joltik-BC-128算法中，主密钥 K 与主调柄 T 都是64 bit， $KT = K||T$ ，令 $TK_0^1 = K$ 、 $TK_0^2 = T$ ， $STK_i = TK_i^1 \oplus TK_i^2 \oplus RC_i$ ，为方便计算，将 TK_i^1 记为 k_i^0 ，将 TK_i^2 记为 k_i^1 或 t_i ，调柄与子密钥的更新函数如下。

$$TK_{i+1}^1 = g_1(h(TK_i^1))$$

$$TK_{i+1}^2 = g_2(h(TK_i^2))$$

在密钥扩展算法中， g_a 函数表示将有限域中的元素 a 乘以状态矩阵的每个半字节， h 为一个字节替换操作，如下所示。

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 6 & 11 & 12 & 5 & 10 & 15 & 0 & 9 & 14 & 3 & 4 & 13 & 2 & 7 & 8 \end{pmatrix}$$

1.4 相关定义与定理

定义1 ($4-\delta$ 集)^[7] 将Joltik-BC-128算法16个半字节的状态矩阵中的某个半字节进行遍历，剩下的全部取固定值，由此得到的集合称为 $4-\delta$ 集，该 $4-\delta$ 集共有 2^4 个元素。

定理1 (S盒的差分特性)^[10] 对于一个已知的非线性S盒，若随机给定一个非零的输入差分 Δ_i 和随机给定一个非零的输出差分 Δ_o ，则平均有一个输入值 x 满足 $S(x) \oplus S(x \oplus \Delta_i) = \Delta_o$ 。

定理2 (轮密钥调柄差分叠加消除特性)^[1,5] 在Joltik-BC-128算法中，轮密钥调柄差分叠加

消除现象每15轮出现一次。假定在 TK^1 和 TK^2 中有相同的活动半字节位，分别以 a_1 和 a_2 代表其差分，那么首轮中该字节的差分为 $a_1 \oplus a_2$ 。由于 TK^1 和 TK^2 每轮都经历相同的移位函数 h ，因此在第 l 轮中该字节的差分为 $g_2(a_2) \oplus a_1$ 。因为 $a_1 \neq 0$ 、 $a_2 \neq 0$ ，所以每15轮中至多存在一个 l 使 $g_2(a_2) \oplus a_1 = 0$ 。

1.5 中间相遇攻击简介

中间相遇攻击是一种用于分析密码算法安全性的技术，1977年由Diffie和Hellman两位学者提出^[11]，最初是用来评估Double-DES算法^[12]安全性的，其主要思想是将算法分解成两部分：首先，选择特殊形式的明文集合，并对该集合的前半部分进行加密，得到对应的筛选集合；然后，猜测相关的密钥，正向加密和逆向解密明文对的若干字节和比特，判断是否构成中间相遇的碰撞，进而形成一个有效的攻击。

2008年，Demirci和Selçuk两位学者^[13]提出一种融合了碰撞思想^[14]与中间相遇攻击思想的新方法DS中间相遇攻击，用于评估AES算法的安全性。这一方法为分析AES类型算法提供了崭新的思路，受到了密码学界的广泛关注，此后AES型算法的DS中间相遇攻击研究工作进展迅速。

2014年，Li等^[15]为进一步降低哈希表 H 的存储空间提出了一种名为相关密钥筛选(key-dependent sieve)的解决方案，其通过各子密钥间的制约关系来降低存储空间。这一方案可以显著降低攻击的存储复杂度，并提高中间相遇攻击效果。

2019年，Liu等^[7]等利用调柄差分控制技术增加了中间相遇攻击的轮数。考虑到调柄可自由

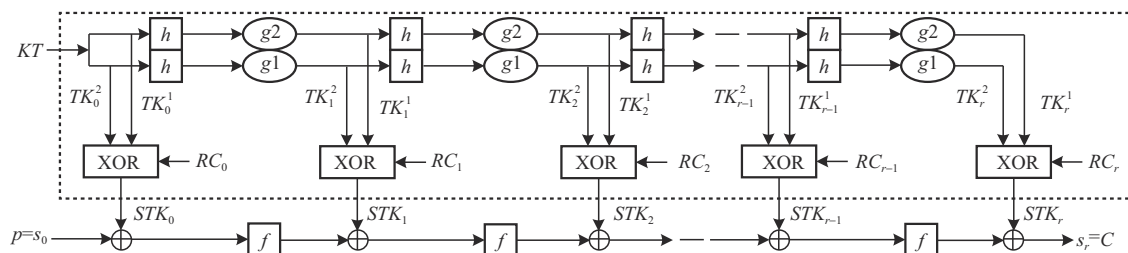


图1 Joltik-BC-128算法的整体框架和加密流程

Figure 1 Overall framework and encryption flow of Joltik-BC-128 algorithm

控制,通过引入特定的调柄,使构造的区分器在第 1 轮不引入差分,从而在不增加复杂度的情况下使区分器增加 1 轮、整个攻击增加 1 轮。

2021 年, Li 等^[8]将轮密钥调柄差分消除技术与中间相遇攻击相结合,使区分器和整体攻击又增加了 1 轮。

近年来,人们提出了多个基于自动化搜索技术的中间相遇攻击方案^[16-18],进一步提升了中间相遇攻击的效果。与此同时,中间相遇攻击也被大量用于分组面膜算法的安全性分析,效果显著。

2 Joltik-BC-128 算法的中间相遇攻击

本节构造了一个全新的 6 轮 Joltik-BC-128 算法中间相遇区分器,在此区分器的基础上向前延伸 1 轮和向后延伸 2 轮,实现了对 9 轮 Joltik-BC-128 算法的中间相遇攻击,减少了攻击的时间复杂度和存储复杂度。

2.1 6 轮 Joltik-BC-128 算法的中间相遇区分器

根据 Joltik-BC-128 算法的差分传播扩散特征,通过引入调柄差分,利用差分枚举和轮密钥调柄差分叠加消除特性构造出一个全新的 6 轮 Joltik-BC-128 算法的中间相遇区分器。考虑到 $w_0[6]$ 为一个活动的半字节、调柄差分可以控制,为使攻击能够达到更多的轮数,首先给定一个特殊的调柄差分使 $\Delta k_1^1[6] = \Delta w_0[6]$,那么 $\Delta k_1^1[6] + \Delta w_0[6] = 0$,因此区分器在第 1 轮中便没有产生差分。在下一轮中再利用轮密钥调柄差分叠加消除特性,使区分器在第 2 轮中又没有产生差分。由于在前两轮中并未产生有效差分,因此可以得到一个较好的区分器。经过 6 轮的 Joltik-BC-128 加密可以得到相应的有序序列 $(x_7^{-1}[10] \oplus x_7^0[10], x_7^{-2}[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus x_7^0[10])$ 。

定理 3 Joltik-BC-128 算法的两个 $4 - \delta$ 集 $\{w_0^0, w_0^1, \dots, w_0^{15}\}$ 和 $\{t_0^0, t_0^1, \dots, t_0^{15}\}$, 满足 $w_0^a[6] = t_1^a[6] \oplus t_1^0[6] = a (0 \leq a \leq 15)$, 进行 6 轮 Joltik-BC-128 算法加密,两个 $4 - \delta$ 集中存在一对状态-调柄元素 $\{(w_0^i, t_1^i)(w_0^j, t_1^j)\}$, 满足如图 2 所示的截断差分特征,则 60 bit 的有序序列 $(x_7^{-1}[10] \oplus x_7^0[10], x_7^{-2}[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus$

$x_7^0[10])$, 可以由 11 个半字节决定,共有 2^{44} 种可能性。

证明: 首先证明,当一对状态-调柄元素 $\{(w_0^i, t_1^i)(w_0^j, t_1^j)\}$ 满足截断差分时,经过 6 轮 Joltik-BC-128 算法加密,得到的输出差分序列 $(x_7^{-1}[10] \oplus x_7^0[10], x_7^{-2}[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus x_7^0[10])$ 的值是由以下 26 个半字节决定的。

$$w_0^i[6], x_3^i[8], x_4^i[8,9,10,11], x_5^i, x_6^i[2,7,8,13]$$

这里, Δx^a 表示 $x^a \oplus x^0 (0 \leq a \leq 15)$ 。

已知 $w_0^i[6]$, 可得 $t_1^i[6] \oplus t_1^0[6]$, 从而可以计算出各轮的调柄差分即 $\Delta t_1^i, \Delta t_2^i, \Delta t_3^i, \Delta t_4^i, \Delta t_5^i, \Delta t_6^i, \Delta t_7^i$ 。因为 $\Delta w_0^i[6] = \Delta t_1^i[6]$, 所以 $\Delta x_1^i = 0$ 。由定理 2 可知 $\Delta \text{STK}_2 = 0$, 故 $\Delta w_2^i = 0$ 。已知 $\Delta t_3^i[8]$, 可得 $\Delta x_3^i[8]$, 又已知 $x_3^i[8]$, 因此可以求出 $\Delta y_3^i[8]$ 。由于轮函数中行移位、列混合和可调轮密钥加均为线性运算,同时已知 $\Delta t_4^i[9]$, 因此可得 $\Delta x_4^i[8,9,10,11]$ 。同理,由于已知 $x_4^i[8,9,10,11]$, 行移位、列混合和可调轮密钥加均为线性运算,因此可得 Δx_5^i 。同理,由于已知 x_5^i , 可得 $\Delta x_6^i[2,7,8,13]$, 已知 $x_6^i[2,7,8,13]$, 可得序列 $(x_7^{-1}[10] \oplus x_7^0[10], x_7^{-2}[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus x_7^0[10])$ 。

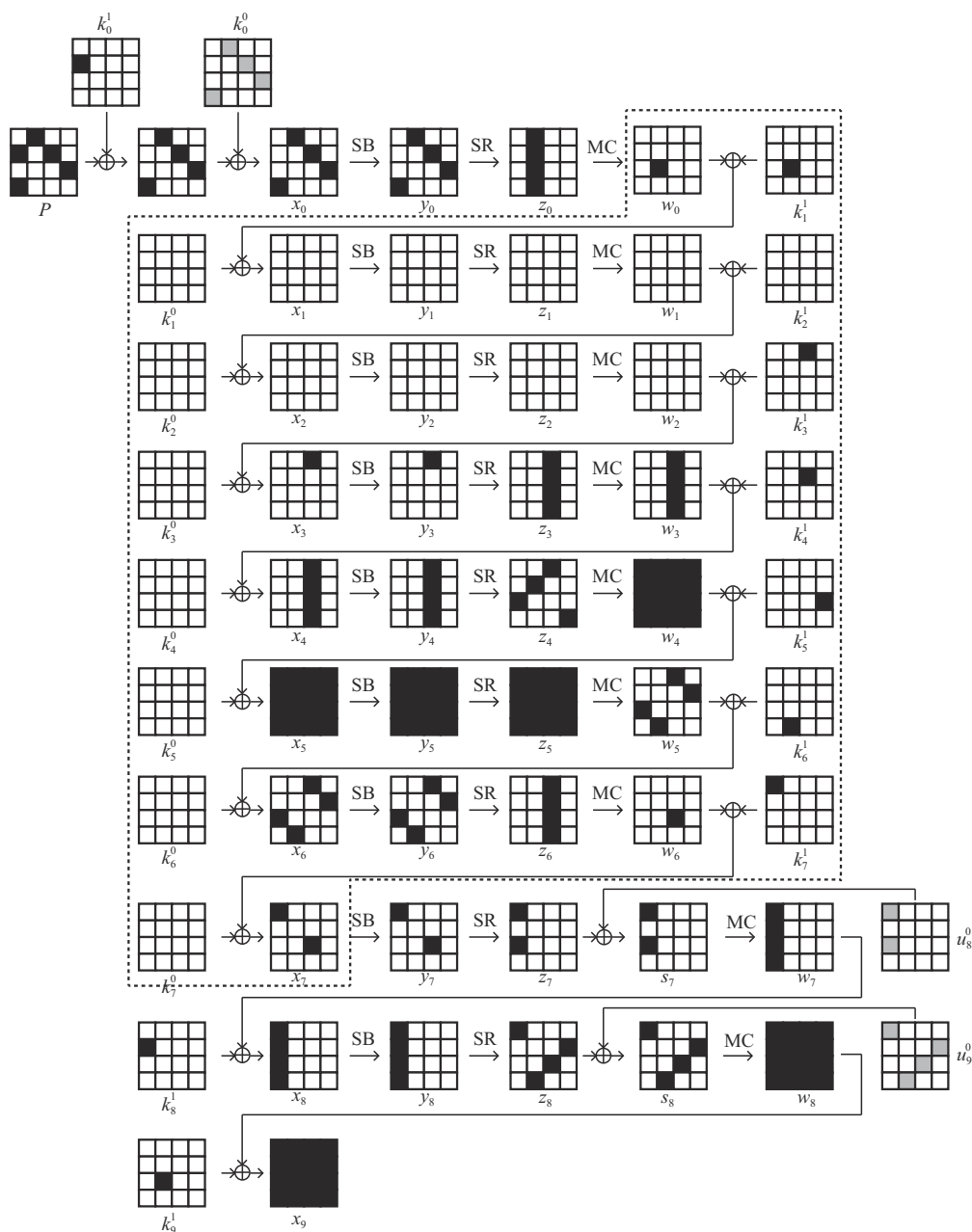
然后,证明上述 26 个半字节可以由以下 11 个半字节代替。

$$w_0^i[6], x_3^i[8], x_4^i[8,9,10,11], y_6^i[2,7,8,13], \Delta w_6^i[10]$$

已知 $w_0^i[6]$, 可得 $t_1^i[6] \oplus t_1^0[6]$, 然后计算出各轮的调柄差分 $\Delta t_1^i, \Delta t_2^i, \Delta t_3^i, \Delta t_4^i, \Delta t_5^i, \Delta t_6^i, \Delta t_7^i$ 。因为 $\Delta w_0^i[6] = \Delta t_1^i[6]$, 所以 $\Delta x_1^i = 0$ 。已知 $\Delta \text{STK}_2 = 0$, 故 $\Delta w_2^i = 0$ 。已知 $\Delta t_3^i[8]$, 那么 $\Delta x_3^i[8]$ 可知。又已知 $x_3^i[8]$, 因此可得 $\Delta y_3^i[8]$ 。由于各轮轮函数中行移位、列混合和可调轮密钥加均为线性运算,并且已知 $\Delta t_4^i[9]$, 因此可得 $\Delta x_4^i[8,9,10,11]$ 。

同理,已知 $x_4^i[8,9,10,11]$, 可得 Δx_5^i 。因为已知 $\Delta w_6^i[10]$, 所以可以求出 $\Delta y_6^i[2,7,8,13]$ 。已知 $y_6^i[2,7,8,13]$, 可以计算出 Δy_5^i 。接下来,根据已知的 Δx_5^i 和 Δy_5^i 、定理 1, x_5^i 和 y_5^i 可解,最后得到所求的差分序列。

因此,所求的有序序列 $(x_7^{-1}[10] \oplus x_7^0[10], x_7^{-2}[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus x_7^0[10])$, 仅有



(注:白块字节处的差分为0,黑块字节处的差分为非0,绿色字节处表示所需的密钥)

图2 9轮 Joltik-BC-128 算法的中间相遇攻击

Figure 2 Meet-in-the-middle attack of the 9-round Joltik-BC-128 algorithm

$2^{11 \times 4} = 2^{44}$ 个取值。证毕。

根据定理3, 用11个半字节就可以表示原来需要26个半字节表示的有序序列, 将有序序列 $(x_7^1[10] \oplus x_7^0[10], x_7^2[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus x_7^0[10])$ 的可能性降低为 2^{44} , 极大地减少了哈希表 H 的存储空间。可见, 这是一个较好的区分器。

2.2 9轮 Joltik-BC-128 算法的中间相遇攻击

基于 Joltik-BC-256 算法的6轮中间相遇区分器, 通过向前延伸1轮, 向后延伸2轮, 实现了一个全新的9轮 Joltik-BC-128 算法的中间相遇攻击, 如图2所示。其中, 在第8轮和第9轮将列混合和可调轮密钥加进行换序, 把密钥的位置提前, 而后利用等价密钥来减少解密过程猜测的密钥字节。

攻击过程分为预计算阶段和在线阶段。在预计算阶段, 根据区分器计算序列 $(x_7^1[10] \oplus x_7^0[10], x_7^2[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus x_7^0[10])$ 的取值有 2^{44} 种可能, 可知哈希表 H 的大小为 $2^{44} \times 60$ bit。接下来的在线处理阶段, 首先需要找到满足截断差分的一对明文 (P_i, P_j) ; 然后猜测密钥并计算对应的序列 $(x_7^1[10] \oplus x_7^0[10], x_7^2[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus x_7^0[10])$; 最后检测求得的序列是否存储在所造的表中, 判断所猜测的密钥是否正确。

在线处理阶段的操作步骤如下。

1) 首先, 定义一个明文结构, 使其在 $P[1,3,4,9,14]$ 处取遍, 剩余的 11 个字节取固定值, 那么该明文集合共有 2^{20} 个取值; 然后, 定义一个调柄结构, 使其在 $k_1^1[6]$ 的半字节取遍, 剩余的 15 个半字节取固定值, 那么该调柄集合共有 2^4 个取值。通过这两个集合可得 $2^{20+4} = 2^{24}$ 个明文-调柄组合, 因此可以得到 $2^{24} \times (2^{24} - 1) \div 2 \approx 2^{47}$ 个满足截断差分特征的明文差分对。

2) 计算 9 轮中间相遇攻击的截断差分特征路径成功的概率, 即 $2^{-4 \times (3+1+1+2+12)} = 2^{-76}$ 。所需的调柄结构为 $2^{76-47} = 2^{29}$ 个, 所以明文-调柄组合共有 $2^{24+29} = 2^{53}$ 个。

3) 由 2^{53} 个明文-调柄组合可以得到其密文, 对密文进行部分解密得到 s_8 , 每个消息对需满足在 $s_8[1,2,3,4,5,6,8,9,11,12,14,15]$ 处没有差分, 满足此条件的消息对仅有 $2^{76-4 \times 12} = 2^{28}$ 个。接下来, 对每个消息对执行以下操作。

① 由 $\Delta w_0[6] = \Delta k_1^1[6]$ 可以得到 $\Delta y_0[3,4,9,14]$, 已知明文差分, 那么可得 $\Delta x_0[3,4,9,14]$, 依据定理 1 可得 $x_0[3,4,9,14]$, 通过异或可得 $k_0^0[3,4,9,14]$ 。

② 猜测 $\Delta y_7[0,10]$, 计算 $\Delta x_8[0,1,2,3]$, 通过密文差分可得 $\Delta y_8[0,1,2,3]$, 依据定理 1 可知 $x_8[0,1,2,3]$ 和 $y_8[0,1,2,3]$, 由 $y_8[0,1,2,3]$ 可得 $z_8[0,7,10,13]$, 由密文可推算出 $s_8[0,7,10,13]$, 通过异或可得 $u_9^0[0,7,10,13]$ 。

③ 在该消息对中选择一个消息进行部分加密, 构造上述的两个 4- δ 集, 然后进行部分解密, 得到 2^4 个明文-调柄组合, 记为

$\{(P^0, T^0), (P^1, T^1), \dots, (P^{15}, T^{15})\}$ 。

④ 猜测 $u_8^0[0,2]$, 由上述 2^4 个明文-调柄组合得到密文, 并利用 $u_9^0[0,7,10,13]$ 和 $u_8^0[0,2]$ 进行解密, 得到有序序列 $(x_7^1[10] \oplus x_7^0[10], x_7^2[10] \oplus x_7^0[10], \dots, x_7^{15}[10] \oplus x_7^0[10])$ 。

⑤ 检测求得的序列是否存储在预计算阶段所造的哈希表 H 中, 若不在, 则舍去; 若在, 则保留。一个错误密钥通过的概率为 $2^{44-60} = 2^{-16}$ 。

4) 穷举 $u_9^0[1,2,3,4,5,6,8,9,11,12,14,15]$ 剩余的 12 个字节, 使主密钥得以恢复。

2.3 复杂度分析

预计算阶段, 时间复杂度为 $2^{44} \times 16 \times 25 \div (9 \times 16) \approx 2^{45.5}$ 次 9 轮 Joltik-BC-128 加密, 存储复杂度为 $2^{44} \times 120 \div 64 \approx 2^{44.91}$ 个长度是 64 的比特块, 数据复杂度为 2^{53} 个选择明文-调柄。

在线处理阶段, 时间复杂度为 $2^{28} \times 2^8 \times 2^8 \times 16 \times 10 \div (9 \times 16) \approx 2^{44.2}$ 次 9 轮 Joltik-BC-128 加密, 穷举 $u_9^0[1,2,3,4,5,6,8,9,11,12,14,15]$ 剩余的 12 个字节的时间复杂度为 2^{48} 次 9 轮 Joltik-BC-128 加密。

因此, 整个攻击总的存储复杂度为 $2^{44.91}$ 个长度是 64 的比特块, 数据复杂度为 2^{53} 个选择明文-调柄, 时间复杂度为 2^{48} 次 9 轮 Joltik-BC-128 加密。

3 结束语

本文主要研究了 Joltik-BC-128 算法的中间相遇攻击, 通过引入调柄差分, 利用差分枚举、S 盒的差分特性和轮密钥调柄差分叠加消除特性, 构造了一个全新的 6 轮 Joltik-BC-128 算法的中间相遇区分器, 将此区分器向前和向后延伸, 实现了对 9 轮 Joltik-BC-128 算法的中间相遇攻击。由于攻击选取的内部差分扩散字节位与调柄差分字节位充分重合, 因此减少了猜测字节量。与已有 Joltik-BC-128 算法的中间相遇攻击结果相比, 该攻击方法显著降低了时间复杂度和存储复杂度。

近年来, 自动化搜索技术成为密码领域关注的焦点, 利用自动化搜索技术改进 Joltik-BC 算法的攻击复杂度将是后续工作的重点。

参考文献：

- [1] JEAN J, NIKOLIC I, PEYRIN T. Tweaks and keys for block ciphers: the tweakable framework [C]//Proceedings of AsiaCrypt 2014. Cham: Springer, 2014: 274-288.
- [2] DOBRAUNIG C, EICHLSEDER M, MENDEL F. Ascon v1. 2. submission to the CAESAR competition[J]. Institute for Applied Information Processing and Communications, 2016, 5(6):7
- [3] BIHAM E, BIRYUKOV A, SHAMIR A. Miss in the middle attacks on IDEA and Khufu[C]//Proceedings of Fast Software Encryption (FSE 1999). Heidelberg: Springer, 1999: 124-138.
- [4] DERBEZ P, FOUQUE P A. Exhausting demirci-selçuk meet-in-the-middle attacks against reduced-round AES[C]//Proceedings of FSE 2013. Heidelberg: Springer, 2014: 541-560.
- [5] ZONG R, DONG X Y. MILP-aided related-tweak/key impossible differential attack and its applications to QARMA, Joltik-BC[J]. IEEE Access, 2019, 7: 153683-153693.
- [6] LI R J, JIN C H, PAN H C. Key recovery attacks on reduced-round Joltik-BC in the single-key setting[J]. Information Processing Letters, 2019, 151: 105834.
- [7] LIU Y, SHI Y F, GU D W, et al. Improved meet-in-the-middle attacks on reduced-round Kiasu-BC and Joltik-BC[J]. The Computer Journal, 2019, 62(12): 1761-1776.
- [8] LI M, CHEN S. Improved meet-in-the-middle attacks on reduced round Joltik-BC[J]. IET Information Security, 2021, 15: 247-255.
- [9] DAEMEN J, RIJMEN V. The design of rijndael: AES--the advanced encryption standard [M]. Berlin, Germany: Springer, 2002.
- [10] KANDA M. Practical security evaluation against differential and linear cryptanalyses for feistel ciphers with SPN round function[C]//Proceedings of the 7th Annual International Workshop on Selected Areas in Cryptography. New York: ACM, 2000: 324-338.
- [11] DIFFIE W, HELLMAN M E. Special feature exhaustive cryptanalysis of the NBS data encryption standard[J]. Computer, 1977, 10(6): 74-84.
- [12] U.S. Department of Commerce, National bureau of standards. data encryption standard[J]. Computers & Security, 1977, 6(3): 195-196.
- [13] DEMIRCI H, SELÇUK A A. A meet-in-the-middle attack on 8-round AES[C]//Proceedings of Fast Software Encryption(FSE 2008). Heidelberg: Springer, 2008: 116-126.
- [14] GILBERT H, MINIER M. A collision attack on 7 rounds of rijndael [C]// Proceedings of the Third Advanced Encryption Standard Candidate Conference. Heidelberg: Springer, 2000: 230-241.
- [15] LI L, JIA K, WANG X. Improved single-key attacks on 9-round AES-192/256[C]//Proceedings of 21st International Workshop on Fast Software Encryption(FSE 2014). Heidelberg: Springer, 2014: 1-20.
- [16] LV Y, SHI D P, GUO Y, et al. Automatic demirci - selçuk meet-In-the-middle attack on SIMON[J]. The Computer Journal, 2023, 66 (12): 3052-3068.
- [17] SHI D P, SUN S W, SONG L, et al. Exploiting non-full key additions: full-fledged automatic demirci-selçuk meet-in-the-middle cryptanalysis of SKINNY[C]//Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Heidelberg: Springer, 2023: 67-97.
- [18] BOURA C, DAVID N, DERBEZ P, et al. Differential meet-in-the-middle cryptanalysis[C]// Proceedings of 43rd Annual International Cryptology Conference(CRYPTO 2023). Heidelberg: Springer, 2023: 240-272.

[作者简介]



李征 (1994-), 男, 辽宁沈阳人, 信息工程大学硕士生, 主要研究方向为分组密码的设计与分析。



任炯炯 (1995-), 男, 甘肃天水人, 信息工程大学讲师, 主要研究方向为分组密码的安全性分析。



李曼曼 (1986-), 女, 河南开封人, 信息工程大学讲师, 主要研究方向为网络空间安全、对称密码的设计与分析。



陈少真 (1967-), 女, 江苏无锡人, 信息工程大学教授、博士生导师, 主要研究方向为密码学与信息安全。