改进的 6 轮 Square 算法中间相遇攻击

何 峰[1],董晓丽[2],韦永壮[1]
(1. 桂林电子科技大学 广西密码学与信息安全重点实验室,广西 桂林 541004;
2. 西安邮电大学 网络空间安全学院,陕西 西安 710121)

摘 要:Square 分组密码算法是由 Daemen 等人设计,并在 1997 年快速软件加密(FSE)国际会议上首次公布。该密码算法基于 SPN(Substitution-Permutation Network)结构,其分组长度和密钥长度均为 128 比特;具有设计新颖、实现高效等优点,也是高级加密标准(AES)算法的前身,在密码学界备受关注。中间相遇攻击作为分组密码算法的重要分析方法之一,如何研究 Square 算法抵抗中间相遇攻击一直是业内讨论的热点问题。该文基于 Square 算法的结构特点和截断差分特征,利用差分枚举技术,构造了一个 3.5 轮中间相遇区分器。通过密钥桥技术及 Square 算法的密钥编排特点,推演出了主密钥与子密钥之间的部分线性关系。由此,将 3.5 轮区分器向前扩展 1 轮,向后扩展 1.5 轮,实现了对 6 轮 Square 算法的中间相遇攻击。该攻击所需数据复杂度为 2^{105} 个选择明文,时间复杂度为 2^{105} 次 6 轮加密,存储复杂度为 2^{85} 个分组。与已有攻击结果相比,新的攻击有效地降低了所需的数据复杂度、时间复杂度和存储复杂度。

关键词:分组密码;Square 算法;中间相遇攻击;差分枚举技术;密钥桥技术
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2025)02-0063-07
doi:10.20165/j.cnki.ISSN1673-629X.2024.0313

An Improved Meet-in-the-middle Attack on 6-Round Square

HE Feng[1],DONG Xiao-li[2],WEI Yong-zhuang[1]
(1. Guangxi Key Laboratory of Cryptography & Information Security,Guilin University of Electronic Technology,Guilin 541004,China;
2. School of Cyberspace Security,Xi'an University of Posts & Telecommunications,Xi'an 710121,China)

Abstract:The Square block cipher was proposed by Daemen et al. at the Fast Software Encryption (FSE) conference in 1997. It uses the Substitution Permutation Network (SPN) structure,and its block length and key length are all 128-bit. Because of its novel design and efficient implementation,also as the predecessor of AES,Square are extensively received attention by cryptographic community. Moreover,the meet-in-the-middle attack is one of the important cryptanalytic methods. How to research the security of Square block cipher against the meet-in-the-middle attack appears to be an interesting topic. In this paper,a new 3.5-round meet-in-the-middle distinguisher is constructed by using the structural characteristics,truncated differentials of the Square and differential enumeration technique. In particular,the partial linear relationship between the subkey and master key in the key schedule is deduced via key bridge technique. Therefore,a new meet-in-the-middle attack on 6-round Square is proposed by adding 1 round (forward encryption operation) and 1.5 round (backward encryption operations) on 3.5-round meet-in-the-middle distinguisher. This attack requires the data complexity of 2^{105} chosen plaintexts,the time complexity of 2^{105} 6-rounds encryption operations,and the memory complexity of 2^{85} blocks. Compared with previous attacks,this attack can effectively reduce the data complexity,time complexity,and memory complexity.

Key words:block cipher;Square;meet-in-the-middle attack;differential enumeration technique;key bridge technique

0 引 言

随着互联网的普及和网络通信的广泛应用,数据的传输与交换已成为现代社会不可或缺的一部分。然而,随着网络技术的不断发展,网络安全问题也变得日益突出。在这个信息时代,保护数据的安全性和隐私已成为至关重要的任务。分组密码作为一种加密算

收稿日期:2024-07-06 **修回日期**:2024-11-07
基金项目:国家自然科学基金资助项目(62162016);广西壮族自治区研究生教育创新计划项目(YCSW2023304,YCBZ2023132);陕西省重点研发计划项目(2023-YBGY-015)
作者简介:何 峰(1999-),男(土家族),硕士研究生,研究方向为密码理论与技术;董晓丽(1982-),女,博士,副教授,研究方向为分组密码的设计与分析;通信作者:韦永壮(1976-),男(壮族),博士,博导,教授,研究方向为对称密码算法设计与分析。

法,在这个背景下备受关注。它能将固定大小的数据块(即分组)映射为相同大小的输出数据块,以提供数据的机密性和完整性。其设计目标通常是在确保高度安全性的同时兼顾高效实现,因此备受业界青睐。随着信息安全需求的持续增长,分组密码安全性的分析变得至关重要。

Square 分组密码[1]是由密码学领域的三位杰出学者 Daemen、Knudsen 以及 Rijmen 联合设计。Square 在密码学的演进历程中扮演着关键的角色,为后来 AES 标准的制定奠定了坚实的基础。

研究者对 Square 算法进行安全分析。Square 算法的设计者运用 Square 攻击[1],成功攻破了 6 轮 Square 密码。Koo 等人[2]对 Square 进行了 Boomerang 相关攻击,运用了 7 轮 Boomerang 区分器,成功实施了全轮攻击。但由于计算机计算能力的限制,没有办法对其进行验证。2011 年,王哲等人[3]构造了一个 4 轮中间相遇区分器,实现了对 5 轮 Square 算法的中间相遇攻击。2019 年,李蒙福等人[4]实现了对 6 轮 Square 算法的中间相遇攻击。表 1 列出了针对单密钥下 Square 算法攻击的结果。其中,MITM 全称为中间相遇攻击(Meet-in-the-Middle Attack)。

表 1 单密钥下 Square 算法攻击结果

攻击方法	攻击轮数	数据复杂度	时间复杂度	存储复杂度	文献
Square	6	2^{32}	2^{72}	2^{32}	文献[1]
Boomerang	8	2^{123}	2^{36}	-	文献[2]
MITM	5	-	2^{72}	2^{72}	文献[3]
MITM	6	2^{109}	2^{109}	2^{88}	文献[4]
MITM	6	2^{105}	2^{105}	2^{85}	该文

中间相遇攻击是一种针对分组密码的有效攻击方法,其原理是通过在加密和解密过程中寻找相同的中间状态,从而在知道明文和密文的情况下找到密钥。1977 年 Diffie 和 Hellman[5]最早提出中间相遇攻击思想,并应用于分组密码 DES[6]的安全性分析中。此后,这一思想迅速在分组密码领域得到广泛应用;在 2008 年 FSE 会议上,Demirci 和 Selçuk[7]构造了 AES 的 4 轮中间相遇区分器,首次提出了对 8 轮 AES-256 的中间相遇攻击。这一方法被普遍称为 Demirci-Selçuk 中间相遇(DS-MITM)攻击,成为针对 SPN 结构分组密码中最经典的中间相遇攻击方法之一;2010 年,Dunkelman 等人[8]利用多重集技术、差分枚举技术以及密钥桥技术对以前的方法进行了改进,有效地降低攻击的复杂度;在 2013 年欧密会上,Derbez 等人[9]通过搜索得到了大量高效路径,进一步降低 Demirci-Selçuk 中间相遇攻击的存储复杂度;2016 年,Derbez 和 Fouque[10]在美密会上提出了一项自动化搜索算法,专门用于中间相遇攻击的任意迭代分组密码算法;2018 年,Shi 等人[11]在亚密会上提出了一种利用 MILP 求解器的方法,用于对 Demirci-Selçuk 中间相遇攻击进行自动化搜索。通过这一自动化搜索模型,研究团队成功展示了对 22 轮 SKINNY-128-384[12]算法的中间相遇攻击;Boura 等人[13]在国际密码讨论会上提出了差分中间相遇攻击,这是一种创新的密码分析技术,融合了中间相遇和差分密码分析的方法。这一新型密码分析方法成功应用于对 SKINNY-128-384 的攻击,进一步提高攻击轮数;2023 年,Ma 等人[14]提出了一项基于自动搜索的 Demirci-Selçuk 中间相遇攻击方法,构建了 CRAFT 的 DS-MITM 自动搜索模型,并借助该模型成功构建了 9 轮 DS-MITM 区分器,提出 19 轮 DS-MITM 攻击;2024 年,Ahmadian 等人[15]在国际密码会议上提出了改进的差分中间相遇攻击方法,包括引入并行分区技术和结构化攻击方法,并将这些技术应用于 CRAFT[16]、SKINNY-128-384 和 SKINNY-64-192[12]密码算法的中间相遇攻击实验验证。2024 年,董晓丽等人[17]在国际期刊上提出一种通过引入价值约束和非线性方程的建立来减少 AES 中 MITM 攻击的预计算表大小,显著降低所需的存储空间,进一步优化 AES 的中间相遇攻击的效率和可行性;2024 年 Lu 等人[18]改进中间相遇攻击,并结合 MixColumns 的特性和多种字节密钥关系,以优化对 10 轮 AES-256 的攻击,降低内存和数据复杂度;Lee 等人[19]进一步优化了 DS-MITM 攻击,重点在多重集预计算中的假阳性概率计算。通过改进攻击分析框架和计算方法,提升了对 7 轮 AES 和 ARIA 的攻击效果。

文献[3-4]分别对 Square 算法进行了 5 轮和 6 轮的中间相遇攻击。文献[3]的问题在于攻击轮数较少,而文献[4]则面临数据复杂度、时间复杂度和存储复杂度较高的问题。受到文献[8,14]的启发,该文改进了 Square 算法的中间相遇攻击。主要创新点如下:

(1)利用 δ-集和差分枚举技术,结合算法的结构特点和截断差分特征,构造了一个新的 3.5 轮区分器。相比文献[4],该区分器在预算阶段输出了一个额外的活跃字节,并且结合截断差分特征,展现出更强的区分能力。

(2)将密钥桥技术其应用于 Square 算法的中间相遇攻击。利用密钥编排的弱点,找到了主密钥与子密钥之间的线性关系,从而减少了在线阶段的密钥猜测量,降低时间复杂度。

(3)与已有的最新攻击相比,时间、数据和存储复杂度降低。基于 3.5 轮的新区分器,前向扩展 1 轮,后向扩展 1.5 轮,最终实现了对 6 轮 Square 算法的中间

相遇攻击。攻击数据复杂度为 2^{105} 个选择明文,时间复杂度为 2^{105} 次 6 轮加密,存储复杂度为 2^{85} 个分组。与文献[4]相比,将攻击的数据复杂度降低了 2^4,时间复杂度降低了 2^4,存储复杂度降低了 2^3。

1　预备知识

1.1　符号说明

主要的符号说明如表 2 所示。

表 2　符号说明

符号	说明
P , C	明文,密文
ΔP , ΔC	明文差分,密文差分
RK_i	轮密钥
θ , γ , π , σ	行混淆,字节替换,矩阵转置,轮密钥加
$X \mid\mid Y$	X 和 Y 级联
$X_i^j[l]$	在第 i 轮中,通过行混淆 (θ) 后的第 j 个状态的第 l 个字节
$Y_i^j[l]$	在第 i 轮中,通过字节替换 (γ) 后的第 j 个状态的第 l 个字节
$Z_i^j[l]$	在第 i 轮中,通过矩阵转置 (π) 后的第 j 个状态的第 l 个字节
$W_i^j[l]$	在第 i 轮中,通过轮密钥加 (σ) 后的第 j 个状态的第 l 个字节
$\Delta X = X \oplus X^*$	状态 X 差分
ΔX^i	$\Delta X^i = X^i \oplus X^*$

1.2　Square 算法

Square 算法是一个 SPN 结构类型的分组密码,分组长度为 128 bit,对应的密钥长度为 128 bit。该算法共有 9 个轮密钥,其中一个为白化密钥,迭代轮数为 8 轮。

Square 算法的轮函数包含有行混淆、字节替换、矩阵转置和轮密钥加。将 128 bit 的明文、中间状态和密文统一称为密码状态。设 $M = \{m_0, m_1, \cdots, m_{15}\}$ 表示一个密码状态,由于密码算法分组长度为 128 bit,对于 $0 \leqslant i \leqslant 15$, m_i 的长度为一个字节。按着行的顺序,依次将密码状态输入到矩阵中,其矩阵的形式如下:

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

Square 算法的加密流程具体步骤如下:

(1)行混淆 (θ):对状态的四行中的每一行分别进行操作,每一行与混淆矩阵相乘。其中,矩阵 $\boldsymbol{U}$ 是一个线性操作,其乘法是在有限域 GF(2^8) 上。具体矩阵如下:

$$\boldsymbol{U} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

(2)字节替换 (γ):这里的 S 盒与 AES 密码算法 S 盒是相同的,将 16 个字节分别进入 S 盒进行替换。注意 S 盒是非线性操作。

(3)矩阵转置 (π):将矩阵进行转置, $b_{i,j} = a_{j,i}$ 即矩阵的行变成列。注意 $\pi = \pi^{-1}$ 。

(4)轮密钥加 (σ):将轮密钥 (RK_i) 与步骤 3 的输出进行异或操作,即 $b_{i,j} = a_{i,j} \oplus \mathrm{RK}_i$ 。

Square 密码算法的密钥编排过程如图 1 所示。

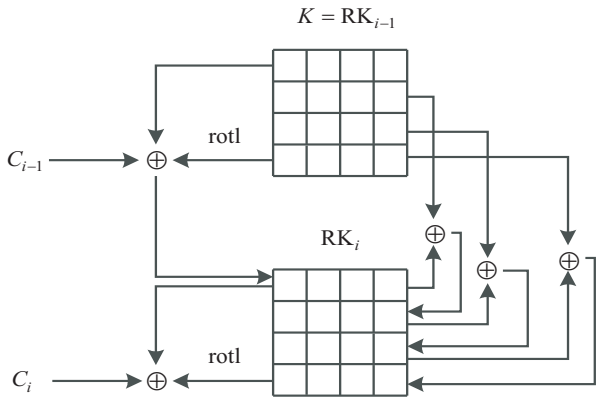

图 1　Square 分组密码算法密钥编排结构

假设 $\mathrm{RK}_0, \mathrm{RK}_1, \cdots, \mathrm{RK}_8$ 为 Square 密码算法 9 个 128 比特的轮密钥,以 RK_0 为主密钥,可以生成其他 8 个轮密钥。RK_i^j 表示第 i 轮密钥的第 j 行。轮密钥生成函数公式如下:

$$\mathrm{RK}_{i+1}^0 = \mathrm{RK}_i^0 \oplus \mathrm{rotr}(\mathrm{RK}_i^3) \oplus C_i \tag{1}$$

$$\mathrm{RK}_{i+1}^1 = \mathrm{RK}_i^1 \oplus \mathrm{RK}_{i+1}^0 \tag{2}$$

$$\mathrm{RK}_{i+1}^2 = \mathrm{RK}_i^2 \oplus \mathrm{RK}_{i+1}^1 \tag{3}$$

$$\mathrm{RK}_{i+1}^3 = \mathrm{RK}_i^3 \oplus \mathrm{RK}_{i+1}^2 \tag{4}$$

其中, C_i 是一个常数, rotr 运算为:

$$\mathrm{rotr}[a_{i,0} \quad a_{i,1} \quad a_{i,2} \quad a_{i,3}] = [a_{i,1} \quad a_{i,2} \quad a_{i,3} \quad a_{i,0}]$$

通过分析 Square 分组密码算法的密钥编排结构可以得知,如果已知其中任意一个轮密钥,就能够推算出上一轮和下一轮的轮密钥。从上面描述的密钥编排方案,可以得到以下性质:

性质 1:如果知道第 6 轮密钥,即 $\mathrm{RK}_6^0, \mathrm{RK}_6^1, \mathrm{RK}_6^2, \mathrm{RK}_6^3$,根据密钥编排就可以得到如下公式。

$$\mathrm{RK}_5^0 = \mathrm{RK}_6^0 \oplus \mathrm{rotr}(\mathrm{RK}_5^3) \oplus C_5 \tag{5}$$

$$\mathrm{RK}_5^1 = \mathrm{RK}_6^0 \oplus \mathrm{RK}_6^1 \tag{6}$$

$$\mathrm{RK}_5^2 = \mathrm{RK}_6^1 \oplus \mathrm{RK}_6^2 \tag{7}$$

$$\mathrm{RK}_5^3 = \mathrm{RK}_6^2 \oplus \mathrm{RK}_6^3 \tag{8}$$

如用 $\mathrm{RK}_6[0,1,2,3]$, $\mathrm{RK}_6[4,5,6,7]$, $\mathrm{RK}_6[8,9,$

10,11], $RK_6[12,13,14,15]$ 来表示 RK_6^0, RK_6^1, RK_6^2, RK_6^3,则上面的公式可以表示为:

$$RK_5[0,1,2,3] = RK_6[0,1,2,3] \oplus \text{rotr}(RK_5[12,13,14,15]) \oplus C_5 \quad (9)$$

$$RK_5[4,5,6,7] = RK_6[0,1,2,3] \oplus RK_6[4,5,6,7] \quad (10)$$

$$RK_5[8,9,10,11] = RK_6[4,5,6,7] \oplus RK_6[8,9,10,11] \quad (11)$$

$$RK_5[12,13,14,15] = RK_6[8,9,10,11] \oplus RK_6[12,13,14,15] \quad (12)$$

由以上公式可得知,若已知 $RK_6[4,5,6,7]$ 和 $RK_6[8,9,10,11]$,则可以计算出 $RK_5[8,9,10,11]$。根据以上公式,可以得知密钥编排是按行进行异或操作,每个字节的异或操作是逐一对应的。由此可知,若已知 $RK_6[6]$ 和 $RK_6[10]$,则可以计算出 $RK_5[10]$。同理,若已知 $RK_6[10]$ 和 $RK_6[14]$,则可以计算出 $RK_5[14]$。

1.3 Demirci-Selçuk 中间相遇攻击

在早期,分组密码的设计主要集中在代换-置换网络(Substitution-Permutation Network, SPN)等结构上。经典密码分析方法包括差分密码分析和线性密码分析等,于 1970 年代末和 1980 年代初广泛使用。中间相遇攻击最早应用于 DES 密码的破解[5]。攻击者可以通过预先计算一部分密钥空间的加密和解密结果,然后在中间的某个步骤相遇,找到正确的密钥。这是中间相遇攻击的开端。

定义 1[8]:对于 Square 算法,δ - 集 $\{Z_0^0, Z_0^1, \cdots, Z_0^{255}\}$ 是 Z_0 的 256 个中间状态,这 256 个中间状态在 Z_0 某一个字节活跃,在其余字节保持固定值。

在 2008 年,Demirci 和 Selçuk[7] 针对 AES 进行安全分析,提出了一种新的中间相遇攻击模型,称为 DS-MITM。主要的基本思想是将整个分组密码算法分成 3 个部分 E_0, E_1 和 E_2,即 $E = E_2 \circ E_1 \circ E_0$。$E_0$ 表示从明文到中间状态的前半部分加密,E_1 表示从中间状态到另一中间状态的中间部分加密,E_2 表示从密文到中间状态的后半部分解密,如图 2 所示。

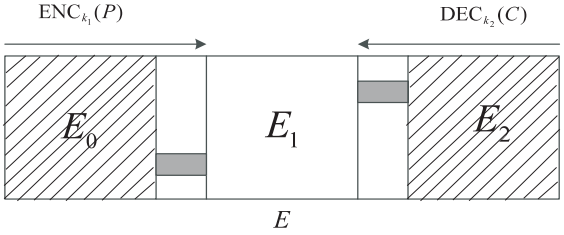

图 2 Demirci-Selçuk 中间相遇攻击模型

中间相遇攻击主要分为两个阶段,这两个阶段分为预计算阶段(离线阶段)和在线阶段。在预计算阶段 E_1 中,主要任务是构建区分器,并使用一个预算表 H 存储区分器的输出有序差分序列。在在线阶段 E_0 中,确定构成 δ - 集的明文集,并且猜测部分密钥 k_1,随后,将明文集加密得到相应的密文。在 E_2 中,猜测部分密钥 k_2,对密文进行解密,得到相应的值。然后,判断这个值与预计算表 H 存储的值是否匹配,若匹配,则猜测的密钥正确;若不匹配,则猜测密钥有误,由此可以排除这个错误密钥。最后,通过穷举搜索找出正确密钥。

2 Square 算法的 6 轮中间相遇攻击

本节,首先构造了 Square 的 3.5 轮中间相遇区分器,其次利用该区分器实现了 6 轮攻击,最后对攻击的数据、时间和存储复杂度进行分析。

2.1 构造 3.5 轮中间相遇区分器

性质 2[20]:对于 S 盒,分别给定非零输入差分 Δ_{in} 和非零输出差分 Δ_{out},则方程 $S(x) \oplus S(x \oplus \Delta_{in}) = \Delta_{out}$ 平均只有一个解。

定理 1:设定 Square 算法的一个 δ - 集,其中 $Z_0[15]$ 是活跃字节。将这个集合进行 3.5 轮 Square 加密。若任选择一个值 $Z_0^*[15]$,则输出有序差分序列 $(X_4^*[10,11] \oplus X_4^0[10,11], X_4^*[10,11] \oplus X_4^1[10,11], \cdots, X_4^*[10,11] \oplus X_4^{255}[10,11])$ 完全有由以下 24 个字节确定。

$X_1^*[12,13,14,15], X_2^*, X_3^*[2,6,10,14]$

证明:$\{Z_0^0, Z_0^1, \cdots, Z_0^{255}\}$ 是一个 δ - 集,其活跃字节为 $Z_0[15]$。记 $Z_0^i[15] \oplus Z_0^*[15] = \Delta Z_0^i[15]$,差分 $\Delta Z_0^i[15]$ 经过 σ 和 θ 操作,可以推算出差分 $\Delta X_1^i[12,13,14,15]$。若已知 $X_1^*[12,13,14,15]$,经过 γ 和 π 操作得到差分 $\Delta Z_1^i[3,7,11,15]$;再经 σ 和 θ 操作,可以得到差分 ΔX_2^i。若已知 X_2^*,则经过 γ 和 π 操作得到差分 ΔZ_2^i;再次经过 σ 和 θ 操作,可以得到差分 $\Delta X_3^i[2,6,10,14]$。若已知 $X_3^*[2,6,10,14]$,同理,依次经过 γ、π、σ 和 θ 操作,可以得到差分 $\Delta X_4^i[10,11]$,即形成有序差分序列 $(X_4^*[10,11] \oplus X_4^0[10,11], X_4^*[10,11] \oplus X_4^1[10,11], \cdots, X_4^*[10,11] \oplus X_4^{255}[10,11])$。

为了降低定理 1 的状态字节猜测,再进一步分析中结合差分枚举技术和 S 盒性质得到的结论如下。

定理 2:结合定理 1,将活跃字节是 $Z_0[15]$ 的 δ - 集 $\{Z_0^0, Z_0^1, \cdots, Z_0^{255}\}$ 进行 3.5 轮 Square 加密。如果该集合中存在一对 (Z_0, Z_0^*) 满足图 3 的截断差分特征,则输出有序差分序列 $(X_4^*[10,11] \oplus X_4^0[10,11], X_4^*[10,11] \oplus X_4^1[10,11], \cdots, X_4^*[10,11] \oplus X_4^{255}[10,11]$ 完全有由以下 11 个字节确定。

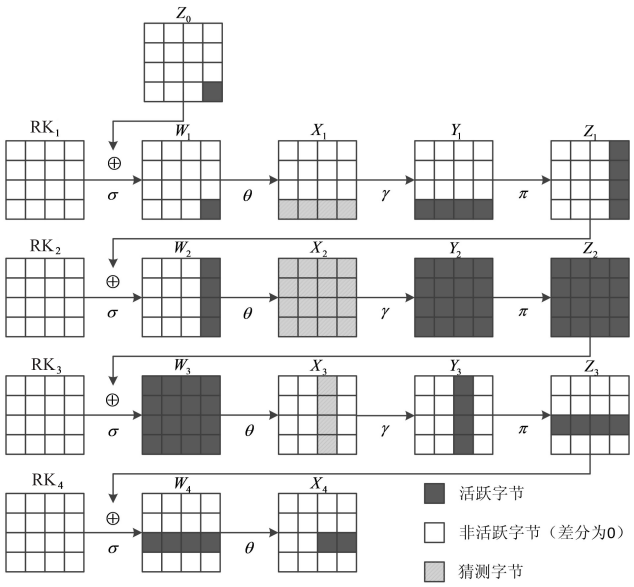

图 3　3.5 轮 Square 中间相遇区分器

$\Delta Z_0[15]$,$X_1^*[12,13,14,15]$,$Y_3^*[2,6,10,14]$,$\Delta X_4[10,11]$

证明:按照加密方向,已知 $\Delta Z_0[15]$,可以推导出 $\Delta X_1[12,13,14,15]$,则 $\Delta Y_1[12,13,14,15]$ 可以由 $\Delta X_1[12,13,14,15]$ 和 $X_1^*[12,13,14,15]$ 推导;根据 $\Delta Y_1[12,13,14,15]$,可以推导出 ΔX_2。按照解密方向,已知 $\Delta X_4[10,11]$,可以推导出 $\Delta Y_3[2,6,10,14]$,则 $\Delta X_3[2,6,10,14]$ 可以由 $\Delta Y_3[2,6,10,14]$ 和 $Y_3^*[2,6,10,14]$ 推导;根据 $\Delta X_3[2,6,10,14]$,可以推导出 ΔY_2。根据性质 2,X_2^* 可以由 ΔX_2 和 ΔY_2 得出。

综上所述,有序差分序列可由以上 11 个字节确定。

2.2　6 轮 Square 算法的中间相遇攻击

在 3.5 轮区分器前面扩展 1 轮,后面扩展 1.5 轮,实现对 6 轮 Square 算法进行中间相遇攻击。具体如图 4 所示。同时,在攻击过程中,利用密钥桥技术,并结合密钥编排的特点以及等价密钥 $\theta^{-1}(K)$ 的应用。

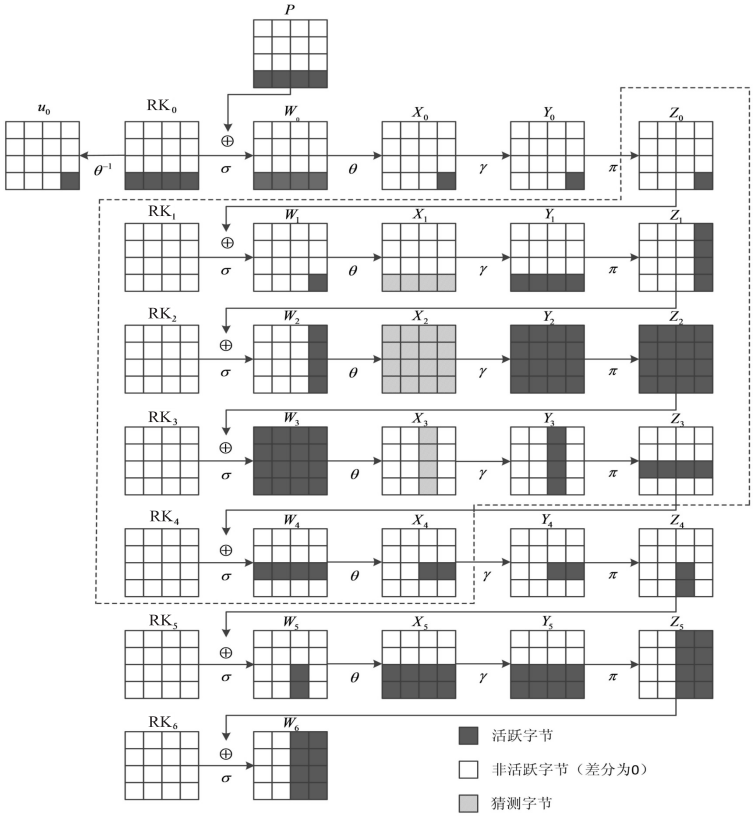

图 4　6 轮 Square 算法中间相遇攻击

攻击过程分为下面两个阶段:预计算阶段(离线阶段)和在线阶段。

预计算阶段:根据定理 2 分析,有序差分序列 $(X_4^*[10,11] \oplus X_4^0[10,11], X_4^*[10,11] \oplus X_4^1[10,11], \cdots, X_4^*[10,11] \oplus X_4^{255}[10,11])$ 由 11 个字节决定,则该序列有 $2^{8\times11}=2^{88}$ 种可能取值,将所有可能取值存储在哈希表 H 中。

在线阶段:首先,找到一对明文且满足图 3 所示的截断差分特征;然后,确定 δ - 集,计算出对应的有序差分序列 $(X_4^*[10,11] \oplus X_4^0[10,11], X_4^*[10,11] \oplus X_4^1[10,11], \cdots, X_4^*[10,11] \oplus X_4^{255}[10,11]$;最后,检测它是否存在于预计算阶段建立的哈希表 H 中。在线阶段的攻击过程具体描述如下:

(1)定义一个明文结构 P,在 (12,13,14,15) 处遍历所有可能的值,其他 12 个字节固定为常数。该结构由 2^{32} 个明文组成,从而可得 $2^{32}\times(2^{32}-1)/2\approx2^{63}$ 个明文对。

(2)确保每个对需要满足图 3 所示的截断差分特征。而 W_0 到 X_0,经过 θ 变换,活跃字节 4 个变成 1 个,其概率为 $2^{-3\times8}$;x_5 到 w_5 经过 θ^{-1} 变换,活跃字节 8 个变成 2 个,其概率为 $2^{-6\times8}$;要满足 w^6 中有 8 个非活跃字节,其概率为 $2^{-8\times8}$;其总概率为 $2^{-(3+6+8)\times8}=2^{-136}$。因此,需要 $2^{136-63}=2^{73}$ 个明文结构,即需要 2^{105} 个明文。对 2^{105} 个明文进行加密来获得密文。

(3)筛选满足 $\Delta W_6[0,1,4,5,8,9,12,13]=0$ 的消息对。符合以上条件的消息对共有 $2^{136-8\times8}=2^{72}$ 对。

(4)对剩余的消息对,对明文对差分 ΔP 进行 θ 变换。选择该变换后差分值在 (12,13,14) 字节为 0 的对。满足以上条件的消息对有 $2^{72-3\times8}=2^{48}$ 个。对 2^{48} 个消息对中每一对进行如下操作:

(a)猜测 $\Delta Z_0[15]$,推断出 $\Delta Y_0[15]$。由明文差分推断出 $\Delta X_0[15]$,根据性质 2 可得 $X_0[15]$,则可以计算出 $u_0[15]$。

(b)猜测 $\Delta Y_4[10,11]$,推断出 $\Delta X_5[8,9,10,11,12,13,14,15]$。由密文差分可以计算出 $\Delta Y_5[8,9,10,11,12,13,14,15]$,根据性质 2 可得 $Y_5[8,9,10,11,12,13,14,15]$,从而得到 $Z_5[2,3,6,7,10,11,14,15]$。最终,由 $Z_5[2,3,6,7,10,11,14,15]$ 和 $W_6[2,3,6,7,10,11,14,15]$ 可推出 $RK_6[2,3,6,7,10,11,14,15]$。由于 $RK_6[6,10,14]$ 已知,根据性质 1 中的密钥编排的弱点可以得到 $RK_5[10,14]$。

(5)对推测出的每一个密钥,选择一个消息对中 (P,P^*) 的消息 P^*,并获得对应的值 $Y_0^*[15]$。然后,令 $Y_0[15]$ 取遍值 (0,1,…,255),从而计算出 $P_1, P_2, \cdots, P_{255}$。然后对其明文进行 6 轮加密,得到对应的密文。利用步骤 4(b)所得到的子密钥对密文进行部分解密得到有序差分序列 $(X_4^*[10,11] \oplus X_4^0[10,11], X_4^*[10,11] \oplus X_4^1[10,11], \cdots, X_4^*[10,11] \oplus X_4^{255}[10,11])$。

(6)检测该序列是否存在于预计算阶段建立的哈希表 H,如果存在,则猜测的子密钥是正确,如果不存在,删除相应的子密钥。一个错误子密钥通过检测的概率为 $2^{88}\times2^{-4096}=2^{-4008}$,猜测的密钥为 $2^{8\times9}=2^{72}$ 个。最后大约剩下 $1+2^{72-4008}\approx1$ 个子密钥。

(7)通过穷举搜索的方式得到主密钥。

2.3 攻击复杂度分析

攻击的复杂度分析具体过程如下:

预计算阶段:基于定理 2 中的 11 字节的参数 $(\Delta Z_0[15], X_1^*[12,13,14,15], Y_3^*[2,6,10,14], \Delta X_4[10,11])$,该阶段需要存储 $2^{8\times11}=2^{88}$ 个有序序列,其中每个序列有 $256\times16=4\,096$ 位。因此,攻击的存储复杂度为 $2^{88}\times4\,096/128=2^{93}$ 个分组。构建预计算表所需要的时间为 $2^8\times2^{8\times11}\times2/6\approx2^{94.4}$ 次 6 轮加密。

在线阶段:在线阶段攻击的时间复杂度为 $2^8\times2^{8\times9}\times1/6\approx2^{77.4}$ 次 6 轮加密,因为涉及 9 个字节的密钥 $(RK_6[2,3,6,7,10,11,14,15], u_0[15])$。攻击的数据复杂度是由数据收集步骤决定。因此,攻击的数据复杂度为 $2^{73}\times2^{32}=2^{105}$ 个选择明文。然而,对于这些明文需要获得相应密文,因此需要的时间复杂度为 2^{105} 次 6 轮加密。综上可得,总攻击的数据复杂度为 2^{105} 个选择明文,时间复杂度为 2^{105} 次 6 轮加密,存储复杂度为 2^{93} 个分组。

2.4 改进预计算阶段复杂度

为了降低预计算阶段的复杂度,针对定理 2 中的 3.5 轮区分器,把 δ 的有序差分序列换成多重集。此时,由于多重集是无序的,且一个 δ - 集可以表示 2^8 个多重集,因此,通过选择 $Z_0^*[15]$ 可以令区分器中 $X_0^*[15]=0$ 来减少一个参数[8],这是可能的,因为状态中 X_1 的字节 15 是活动的。此时,3.5 轮的区分器的参数由 11 个减少到 10,因此,预计算阶段的时间复杂度和存储复杂度均可降低 2^8。

该总时间复杂度为预算阶段时间复杂度、在线阶段时间复杂度以及获得相应密文时间复杂度之和,总时间复杂度大约为 $2^{94.4-8}+2^{77.4}+2^{105}\approx2^{105}$ 次 6 轮加密。最终,数据复杂度为 2^{105} 个选择明文,时间复杂度为 2^{105} 次 6 轮加密,存储复杂度为 2^{85} 个分组。与已有结果相比,该攻击的时间、数据和存储复杂度均有效降低,比较结果见表 1。

3　结束语

如何评估 Square 算法抵抗中间相遇攻击的能力一直是业内研究重点。该文通过 DS-MITM 模型,结合算法的结构特点和截断差分特征,利用差分枚举技术,构造了 3.5 轮区分器。通过密钥桥技术,结合密钥编排的特点,推演出了主密钥与子密钥之间的部分线性关系。由此,将 3.5 轮区分器向前扩展 1 轮,向后扩展 1.5 轮,实现了对 6 轮 Square 算法的中间相遇攻击。与已有结果相比,该攻击的时间、数据和存储复杂度均降低。尽管 6 轮中间相遇攻击没有对全轮(满轮 8 轮)造成实际的威胁,但是文中的攻击手段将为分析其他后续算法如 RAIN[21] 等的多轮中间相遇区分器的设计提供新思路。

参考文献:

[1] DAEMEN J, KNUDSEN L, RIJMEN V. The block cipher square[C]//Proceedings of the 4th international workshop on fast software encryption. Berlin: Springer, 1997: 149-165.

[2] KOO B, YEOM Y, SONG J. Related-key boomerang attack on block cipher SQUARE[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2011, 94(1): 3-9.

[3] 王　哲,张文英. 对 5 轮 Square 的中间相遇攻击[J]. 计算机技术与发展,2011,21(6):132-135.

[4] 李蒙福,苏凡军. 6 轮 Square 密码算法的中间相遇攻击[J]. 计算机技术与发展,2019,29(3):106-110.

[5] DIFFIE W, HELLMAN M E. Special feature exhaustive cryptanalysis of the NBS data encryption standard[J]. Computer, 1977, 10(6): 74-84.

[6] DUNKELMAN O, SEKAR G, PRENEEL B. Improved meet-in-the-middle attacks on reduced-round DES[C]//International conference on cryptology in India. Chennai: Springer, 2007: 86-100.

[7] DEMIRCI H, SELÇUK A A. A meet-in-the-middle attack on 8-round AES[C]//Proceedings of the 15th international workshop on fast software encryption. Berlin: Springer, 2008: 116-126.

[8] DUNKELMAN O, KELLER N, SHAMIR A. Improved single-key attacks on 8-round AES-192 and AES-256[C]//Proceedings of 16th international conference on the theory and application of cryptology and information security. Singapore: Springer, 2010: 158-176.

[9] DERBEZ P, FOUQUE P A. Exhausting Demirci-Selçuk meet-in-the-middle attacks against reduced-round AES[C]//Proceedings of 20th international workshop on fast software encryption. Singapore: Springer, 2014: 541-560.

[10] DERBEZ P, FOUQUE P A. Automatic search of meet-in-the-middle and impossible differential attacks[C]//Annual international cryptology conference. Santa Barbara: Springer, 2016: 157-184.

[11] SHI D, SUN S, DERBEZ P, et al. Programming the Demirci-Selçuk meet-in-the-middle attack with constraints[C]//International conference on the theory and application of cryptology and information security. Brisbane: Springer, 2018: 3-34.

[12] BEIERLE C, JEAN J, KÖLBL S, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS[C]//Proceedings of 36th annual international cryptology conference. Santa Barbara: Springer, 2016: 123-153.

[13] BOURA C, DAVID N, DERBEZ P, et al. Differential meet-in-the-middle cryptanalysis[C]//Annual international cryptology conference. [s. l.]: Springer, 2023: 240-272.

[14] MA Z, LI M, CHEN S. Meet-in-the-middle attacks on round-reduced CRAFT based on automatic search[J]. IET Information Security, 2023, 17(3): 534-543.

[15] AHMADIAN Z, KHALESI A, M'FOUKH D, et al. Improved differential meet-in-the-middle cryptanalysis[C]//Annual international conference on the theory and applications of cryptographic techniques. [s. l.]: Springer, 2024: 280-309.

[16] 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.

[17] DONG X, LIU J, WEI Y, et al. Meet-in-the-middle attacks on AES with value constraints[J]. Designs, Codes and Cryptography, 2024, 92(9): 2423-2449.

[18] LU J, ZHOU W. Improved meet-in-the-middle attack on 10 rounds of the AES-256 block cipher[J]. Designs, Codes and Cryptography, 2024, 92(4): 957-973.

[19] LEE D, HONG D, SUNG J, et al. Accurate false-positive probability of multiset-based Demirci-Selçuk meet-in-the-middle attacks[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2024, 107(8): 1212-1228.

[20] KANDA M. Practical security evaluation against differential and linear cryptanalyses for Feistel ciphers with SPN round function[C]//International workshop on selected areas in cryptography. Waterloo: Springer, 2000: 324-338.

[21] 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.