6 轮 Square 密码算法的中间相遇攻击

李蒙福,苏凡军
(上海理工大学 光电信息与计算机工程学院,上海 200093)

摘 要: 分组密码具有速度快、易于标准化和便于软硬件实现等特点,通常是信息和网络安全中实现数据加密、数字签名、认证及密钥管理的核心体制。密码算法的安全性分析与设计两者难以分离,一方面,在对密码进行安全性分析的过程中,可以为设计出更加安全的密码积累经验,另一方面,在密码算法的设计中也会涉及很多具有现实意义的技术和应用价值的知识。作为分组密码的一个重要组成部分—SPN 型分组密码,对其进行研究和分析具有很大的现实意义。Square 是 SPN 型分组密码其中之一,其密钥长和分组长都为 128 bit。通过研究 Square 算法的结构特征和一类截断差分的性质,利用差分枚举技术和多重集构造了 Square 算法的 4 轮中间相遇区分器,给出了对 6 轮 Square 密码算法的中间相遇攻击。新的区分器由 10 个参数决定。基于新的区分器,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^{109},时间复杂度为 2^{109},存储复杂度为 2^{84}。

关键词: Square 密码; 差分枚举; 多重集; 中间相遇攻击
中图分类号: TN918.1 **文献标识码:** A **文章编号:** 1673-629X(2019)03-0106-05
doi: 10.3969/j.issn.1673-629X.2019.03.023

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

LI Meng-fu, SU Fan-jun
(School of Optoelectronic Information and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Block ciphers are characterized by their high speed, easy standardization and hardware and software implementation, usually as the core system of data encryption, digital signature, authentication and key management in information and network security. It is difficult to separate the security analysis and design of cryptographic algorithms. On the one hand, in the process of ciphers security analysis, experience can be accumulated for the design of more secure ciphers. On the other hand, in the design of cryptographic algorithms, there will be a lot of practical significance of technology and application value of knowledge. SPN block ciphers are an important part of block ciphers, which is of great significance to be studied and analyzed. Square is a block cipher with substitution-permutation network, which operates on 128-bit blocks and 128-bit keys. By studying the structural characteristics and the properties of truncated differential of Square, we construct a 4-round meet-in-the middle distinguisher by using differential enumeration technique and multiple sets, and give a meet-in-the-middle attack on 6-round Square. The new distinguisher is determined by 10 parameters. Based on the new distinguisher, we extend the meet-in-the-middle attack on 6-round Square for the first time with 2^{109} chosen plantexts, 2^{109} computations and 2^{84} memories.

Key words: Square; differential enumeration; multiple sets; meet-in-the-middle

1 概 述

随着信息技术的高速发展,数据安全的问题愈加凸显。无论是在理论上还是技术上,密码学在信息安全领域都是不可或缺的。分组密码具有速度快、易于标准化和便于软硬件实现等特点,通常是信息和网络安全中实现数据加密、数字签名、认证及密钥管理的核心体制,也是对称密码学的一个重要分支。分组密码已经在信息安全领域得到了非常广泛的应用,如数字通信安全、工业网络控制安全、无线传感器网络感知安全、无线射频识别安全以及电子商务支付安全等领域。分组密码的研究内容主要包括分组密码的设计和分析,两者相互作用,共同推动着分组密码理论的发展。

收稿日期: 2018-04-12 **修回日期:** 2018-08-16 **网络出版时间:** 2018-12-19
基金项目: 国家自然科学基金(61703278)
作者简介: 李蒙福(1989-),女,硕士,研究方向为信息安全、分组密码; 苏凡军,博士,讲师,研究方向为计算机网络和网络安全。
网络出版地址: http://kns.cnki.net/kcms/detail/61.1450.TP.20181219.1532.046.html

一方面,在对密码进行安全性分析的同时,可以为设计出更加安全的密码积累更多的经验,另一方面,在密码算法的设计中也会涉及到很多具有现实意义的信息安全技术和一些具有实际应用价值的理论知识。

Square 算法是由 Joan Daemen 和 Vincent Rijmen 提出的分组密码[1],发表于 1997 年,是 Rijndael 的先驱。Square 也是一个具有 8 轮代换-置换网络[2](SPN)结构的分组密码,可以在各种处理器上实现非常高效的操作。该算法采用了 2 维 4 × 4 的字节矩阵,分组长度和密钥长度都是 128 bit。通常,实施对整轮密码算法攻击的难度是非常大的,但是可以利用一些攻击方法减少迭代次数并对密码算法进行分析以衡量其安全性,这样不仅能够促进密码的发展,而且对信息安全也具有重要意义。

当前,分组密码的安全性研究也是一个热点问题。很多密码学研究者在多篇文献中对多种密码算法的安全性进行分析和研究,另外,许多高效的分析方法也被陆续提出,如差分密码分析、截断差分分析、中间相遇攻击等[3-6]。

截断差分密码分析是由 Knudsen 等提出的差分密码分析的一个变形,与经典的差分分析考虑的具体差分值不同,截断差分只考虑差分的一部分性质。利用截断差分的思想,可以对某些抵抗经典差分密码分析的算法进行攻击。截断差分分析的一般流程是: 首先,寻找一条高概率且有效的 $r-1$ 轮截断差分路径; 然后,加密符合要求的明文进而获得密文,从得到的密文对中筛选除差分对符合要求的密文对; 最后,对上一步中的密文对进行部分解密,通过计数的方法确定正确的密钥。文献[7]给出了 Crypton 算法的不可能差分分析。

中间相遇攻击是在文献[8]中提出,最早是用来对分组密码的一种尝试性扩展进行攻击,之后广泛应用于分组密码和 Hash 函数的安全性分析中。它通过存储加密或解密的中间值,并使用这些中间值来改善强制解密密钥所需的时间,从而削弱了使用多重加密的安全性好处。这使得中间相遇攻击(MITM)成为通用的时空权衡密码分析方法。该种攻击方法的攻击原理是: 首先构造出一个合适的区分器,然后将区分器前面的明文从前向后进行加密,后面的密文从后向前进行解密,接着和前面构造出的区分器连接,最后形成一条连贯的攻击路线。其中可能会出现两种结果,假设加密和解密的部分能够顺利连接上区分器,那么猜出的密钥值是无误的,不然为错误的,最后获取符合的密钥值。通过以上步骤,多数的不对密钥会被排除掉。尽管在构造区分器的过程中预计算的复杂度和存储复杂度可能有点大,不过由于仅仅只进行一次的预计算过程,从而在很大程度上也能降低攻击的时间复杂度,所以中间相遇攻击还是一种比较有效的攻击方法。

近年来,中间相遇攻击已成为比较成熟和富有成效的密码分析方法之一,广泛应用于多种分组密码的安全性分析中。文献[9]给出了 8 轮 AES 的中间相遇攻击,文献[10]给出了利用多重集和差分枚举技术进行 8 轮 AES 中间相遇攻击,文献[11]给出了利用差分枚举技术和有序差分集合进行 11 轮 3D 中间相遇攻击,文献[12]给出了缩减轮 Crypton 的中间相遇攻击分析,文献[13]使用中间相遇攻击对 Kalyna 算法进行安全性分析。

Square 密码算法的安全性分析最早是文献[1]中提出的 Square 攻击,其成功给出了 6 轮 Square 密码的攻击结果,时间复杂度为 2^{72}。文献[14]对 Square 进行了 Boomerang 攻击,运用了 7 轮 Boomerang 区分器,实现了全轮的攻击,这也是目前最好的攻击结果。但是,限于计算机的计算能力,其可行性还有待验证。2011 年,文献[15]给出了 5 轮 Square 的中间相遇攻击分析,预计算时间复杂度为 2^{34},空间复杂度为 2^{72}。

Dunkelman 等在分析 AES 时提出了中间相遇攻击"多重集"的概念。主要思想为: 在构造区分器的过程中,完整的输出序列不需要进行存储,只将符合要求的无序的差分筛选集合进行存储,同时再结合有关截断差分的性质,进一步减少决定区分器的参数个数,达到降低存储复杂度的目的。

在分析 Square 算法的结构特征和一类截断差分的性质的基础上,文中利用差分枚举技术和反弹式思想构造出 Square 算法的 4 轮中间相遇区分器,该区分器由 11 个参数决定。在 4 轮区分器的基础上前后各扩展一轮,首次实现了对 6 轮 Square 算法的中间相遇攻击。在 6 轮 Square 密码的攻击过程中,文中通过减少明文数量以降低数据复杂度,又通过使用了多重集进而将决定区分器的参数个数减少为 10 个,从而进一步降低了预计算的存储复杂度。

2　预备知识

2.1　Square 算法描述

Square 是一个分组长为 128 bit,密钥长为 128 bit 的 SPN 型迭代分组密码。Square 的轮变换由四个不同的变换组成。128 bit 的分组可以用 4 × 4 的 2 维字节矩阵表示。16 个字节分组 $A=\{a_0,a_1,\cdots,a_{15}\}$ 可表示为如下形式:

$$\boldsymbol{A}=\begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{pmatrix}$$

(1) M:列混合,对一个状态的每一列进行操作。

$M:\ b = M(a) \Leftrightarrow b_{i,j} = c_j a_{i,0} \oplus c_{j-1} a_{i,1} \oplus c_{j-2} a_{i,2} \oplus c_{j-3} a_{i,3}$

(2) S:非线性变换,S 是一个可逆的 8 位替换表或 S 盒。

$S:\ b = S(a) \Leftrightarrow b_{i,j} = S(a_{i,j})$

(3) T:转置,状态的行列交换。

$T:\ b = T(a) \Leftrightarrow b_{i,j} = a_{i,j}$, $T^{-1} = T$

(4) σ:轮密钥加。

$\sigma[k_t]:\ b = \sigma[k_t](a) \Leftrightarrow b = a \oplus k_t$

每一轮进行以上 4 个操作后,迭代 8 轮即可得到相关密文,解密操作基本上和加密操作一样,只是调换了密钥的顺序,Square 的密钥编排详见文献[15]。

在某些情况下,在两个连续的轮次中交换 M 和 k_i 的顺序。由于这些操作是线性的,所以可以互换[2]。具体操作是:首先用相等的密钥 $M^{-1}(k_i)$ 排除数据,然后再应用操作 M。例如,$k_0 \oplus M(x) = M(M^{-1}(k_0) \oplus x)$。

2.2 符号说明

x_i:第 i 轮经过密钥加 k_i 的状态;

y_i:第 i 轮经过 M 变换的状态;

z_i:第 i 轮经过 S 的状态;

w_i:第 i 轮经过 T 的状态;

Δx:状态 x 的差分;

$x[i]$:状态 x 的第 i 个字节;

$x[i,\cdots,j]$:状态 x 从第 i 个字节到第 j 个字节;

$u_i = M^{-1}(k_i)$;

$x \| y$: x 与 y 连接;

$\oplus$:异或;

Δx^m: $= x^m \oplus x^i$。

3 6 轮 Square 算法的中间相遇攻击

这一节中,将描述在 6 轮 Square 上的 MITM 攻击。首先,给出需要的定义、引理和推论。然后,提出 4 轮 MITM 区分器,并在 6 轮 Square 上进行 MITM 攻击。最后,通过改进 4 轮区分器,以实现相应攻击复杂度的降低。

3.1 4 轮中间相遇区分器

定义 1:Square 算法的 1 个字节遍历 256 个所有可能值,其他 15 个字节取固定值,这样的结构称为 δ 集,表示为 $\{X_m^0, X_m^1, \cdots, X_m^{255}\}$,其中 X_m^i 是 256 bit,$0 \leqslant i \leqslant 255$,$m(m \in \{1,2,\cdots,8\})$ 为 Square 算法的迭代轮数。一个 δ 集的例子如下所示:

$$\begin{pmatrix} c & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{pmatrix}$$

定义 2[10]:允许元素出现多次,不计元素顺序的集合称为多重集。δ 集中的元素 X_m^i 构造的多重集:$(X_m^0[0] \oplus X_m^0[0], X_m^1[0] \oplus X_m^0[0], \cdots, X_m^{255}[0] \oplus X_m^0[0])$,记为 $(\Delta X_m^0, \Delta X_m^1, \cdots, \Delta X_m^{255})$,一个无序的多重集有 $2^{506.17}$ 个可能的取值。

引理 1:给定 Δ_i 和 Δ_o 两个非零差分元素,方程 $S(x) \oplus S(x \oplus \Delta_i) = \Delta_o$,平均有一个解。这个属性也适用于 s^{-1}。

推论 1:选择 δ 集 $\{w_0^0, w_0^1, \cdots, w_0^{255}\}$ 前 128 个值进行 4 轮 Square 加密,若存在明文对 (w_0^i, w_0^j) $(0 \leqslant i, j \leqslant 255)$ 符合图 1 截断差分特征,则对应差分序列 $(X_5^0[0] \oplus X_5^i[0], X_5^1[0] \oplus X_5^i[0], \cdots, X_5^{127}[0] \oplus X_5^i[0])$ 可由以下 25 个字节决定:

$y_1^i[0,4,8,12] \| y_2^i[\{m \mid 0 \leqslant m \leqslant 15\}] \| y_3^i[0,1,2,3] \| y_4^i[0]$

证明:通过 $y_1^i[0,4,8,12]$ 可计算出 $\Delta y_1^i[0,4,8,12]$,利用 S 盒的非线性替换性质可推算 $\Delta z_1^i[0,4,8,12]$,$\Delta z_1^m[0,4,8,12] = S(y_1^i[0,4,8,12]) \oplus S(\Delta y_1^i[0,4,8,12] \oplus y_1^i[0,4,8,12])$,$\Delta x_2^m[0,1,2,3]$ 可通过等式 $\Delta x_2^m[0,1,2,3] = T(z_1^m[0,4,8,12])$ 得到。同样,可以通过 y_2^i 推导出 Δz_2^i,然后推算 Δx_3^i,接着利用 $y_3^i[0,1,2,3]$ 推导差分 $\Delta z_3^i[0,1,2,3]$,计算 $\Delta x_4^i[0]$,最后利用 $y_4^i[0]$,从而推导出序列 $(X_5^0[0] \oplus X_5^i[0], X_5^1[0] \oplus X_5^i[0], \cdots, X_5^{127}[0] \oplus X_5^i[0])$。

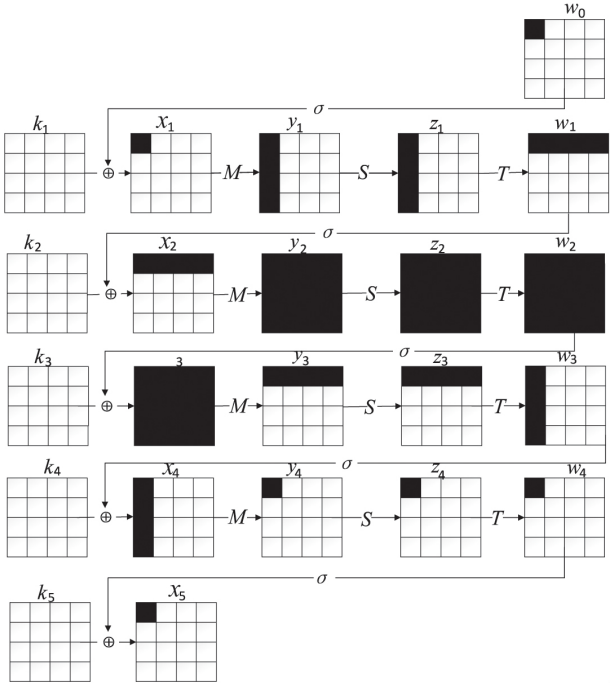

图 1 6 轮 Square 攻击的截断差分特性图

然而,以上给出的 25 个字节其实可以由以下 11 个字节决定:

$\Delta w_0^i[0] \| y_1^i[0,4,8,12] \| z_3^i[0,1,2,3] \| z_4^i[0] \|$

Δw_4^i [0]

证明: 已知 Δw_0^i [0]，可推算 Δy_1^i [0,4,8,12]，而 Δz_1^i [0,4,8,12] 可以由 Δy_1^i [0,1,2,3] 和 y_1^i [0,4,8,12] 计算得到，Δx_2^i [0,1,2,3] 可由 Δz_1^i [0,4,8,12] 推导计算出，然后推导出 Δy_2^i。利用 Δw_4^i [0] 经过 T^{-1} 操作可以得 Δz_4 [0]，又已知 z_4^i [0] 可以计算出 Δy_4^i [0] 和 Δx_4^i [0,4,8,12]，从而求出 Δw_3^i [0,4,8,12]，进而推导出 Δz_3^i [0,1,2,3]，同样利用 z_3^i [0,1,2,3] 推导出 Δz_2^i。根据引理 1，y_2^i 可由 Δy_2^i 和 Δz_2^i 算出。所以差分序列 (X_5^0 [0] $\oplus X_5^i$ [0], X_5^1 [0] $\oplus X_5^i$ [0], …, X_5^{127} [0] $\oplus X_5^i$ [0]) 可以完全由 11 个字节决定。

3.2　6 轮中间相遇攻击

6 轮 Square 算法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 2 所示。

1. 预计算阶段。

遵循推论 1 的证明，预计算阶段需建包含 $2^{11\times8} = 2^{88}$ 个差分序列 (X_5^0 [0] $\oplus X_5^i$ [0], X_5^1 [0] $\oplus X_5^i$ [0], …, X_5^{127} [0] $\oplus X_5^i$ [0]) 的预计算表。存储这些预计算表需要 $2^{11\times8} \times 2^7 \times 8/2^7 = 2^{91}$ 个 128 bit 块。

2. 在线攻击阶段。

首先找到满足图 1 中截断差分特征的明文对 (p_i, p_j)，并确定 δ 集，然后计算差分序列并检测它是否在预计算阶段建立的预计算表中，在线阶段的攻击过程描述如下:

(1) 定义一个 2^{32} 个明文结构 P [0,4,8,12]，其余 12 个字节固定为常数。使用这一结构可构成 $2^{32} \times (2^{32}-1)/2 \approx 2^{63}$ 个明文对，并且每个明文对都满足明文差分特性。攻击过程中还需对 2^{81} 个明文结构进行加密以找到 $2^{81} \times 2^{63} \times 2^{-12\times8} = 2^{48}$ 个对应的密文对以验证密文差分。因此，对于正确的密钥，2^{48} 对中有 $2^{48} \times 2^{-2\times3\times8} = 1$ 对遵循整个截断差分特征。因为在第 1 轮和第 5 轮的 M 操作中有两个 4→1 的转换。注意，不必检查每一对去找到 2^{48} 个密文对，可以将结构存储在由密文中的 12 个非活动字节索引的散列表中，以获得正确的对。

(2) 对 2^{48} 对中的每一对，假设它遵循图 1 中的整个截断差分特征，进行如下操作:

(a) 猜测 Δw_0 [0]，推导出 Δz_0 [0]，根据明文差分推导出 Δy_0 [0]，根据引理 1 得到 y_0 [0]，然后推出 u_0 [0]。

(b) 猜测值 Δx_5 [0]，推导出 Δy_5 [0,4,8,12]，根据密文差分，推导出 Δz_5 [0,4,8,12]，根据引理 2 可得出 z_5 [0,4,8,12]，z_5 [0,4,8,12] 经过 T 操作后得到 w_5 [0,1,2,3]，最后推出 k_6 [0,1,2,3]。

(c) 根据步骤 a 中的 2^8 个值推出的子密钥，改变 w_0^0 [0] 的值，并计算 (w_0^0 [4,8,12], w_0^1 [4,8,12], …, w_0^{127} [4,8,12])，然后计算出对应的明文 (P^0, P^1, …, P^{127}) 并查询它们对应的密文。

(d) 利用步骤 b 中的 2^8 个推导的子密钥，对在步骤 c 中推导出的每个密文进行部分解密以得到差分序列 (X_5^0 [0] $\oplus X_5^i$ [0], X_5^1 [0] $\oplus X_5^i$ [0], …, X_5^{127} [0] $\oplus X_5^i$ [0])。检查该序列是否在预计算表中，如果在，则认为猜测的子密钥是正确的，否则删除子密钥。而错误子密钥通过的概率是 $2^{88} \times 2^{-1024} = 2^{-936}$，非常小，所以可认为猜测出来的子密钥都是正确的。

(3) 最后只剩下 $1 + 2^{48} \times 2^{16} \times 2^{-936} \approx 1$ 个子密钥。根据得到的部分密钥，通过穷举搜索剩余未知字节以恢复全部密钥。

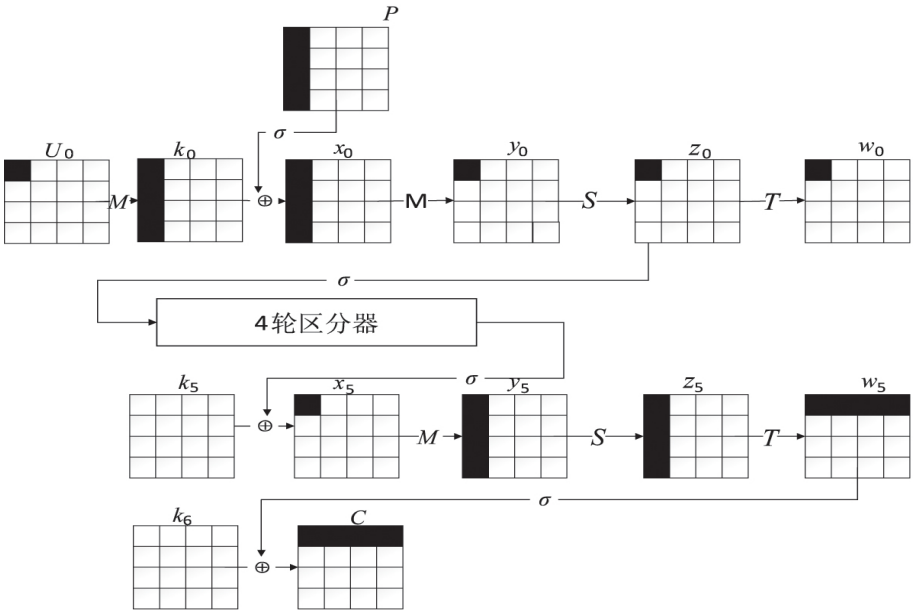

图 2　6 轮中间相遇攻击图

为了评估在线阶段的时间复杂度,文中计算了 6 轮 Square 加密的数量。在步骤 1 中,时间复杂度约为 $2^{81} \times 2^{32} = 2^{113}$。步骤 2 的时间复杂度由步骤 d 决定,其时间复杂度等于 $2^{48} \times 2^{16} \times 2^{7} \times \frac{1+4}{6 \times 16} = 2^{66.74}$。在步骤 3 中,时间复杂度为 $2^{12 \times 8} = 2^{96}$。预计算的时间复杂度为 $2^{11 \times 8} \times 2^{7} \times \frac{1+4+16+4}{6 \times 16} = 2^{93.06}$。综上,总的攻击时间复杂度为 2^{113},数据复杂度为 $2^{81} \times 2^{32} = 2^{113}$,存储复杂度为 2^{91}。

3.3 对 6 轮 Square 攻击的改进

通常,为了降低存储复杂度,可利用多重集把对 6 轮 Square 中间相遇攻击决定区分器的参数减少一个(w_0 [0] 可不要)。

由定义 2 可知,一个 δ 集会遍历 256 个状态,第 m 轮 σ 变换进行之后遍历的仍然是 256 个状态,同样进行 M 变换后还是遍历 256 个状态,对差分 $\Delta X_m^i = X_m^i \oplus X_m^0 (0 \leqslant i \leqslant 255)$ 进行考虑,因为攻击过程中考虑的多重集是无序的,它也遍历 256 个状态,所以差分集合 $\{\Delta X_m^0, \Delta X_m^1, \cdots, \Delta X_m^{255}\}$ 不影响后面的攻击。而对 6 轮 Square 攻击的改进就可以考虑推论 1 中 $\{w_0^0, w_0^1, \cdots, w_0^{255}\}$ 的所有值,因为 w_{m+1}^0 是活动字节,通过选择 w_m^0 使 $w_{m+1,0}^0 = 0$ 是可能的,所以就可以把决定区分器的参数个数减少一个,为 10 个。首先,使用一个杂凑表将多重集的所有可能值存储起来,因为在构建预计算区分器的过程中都会用到。然后,筛选出符合要求的若干明文进行加密。同时对特定的密钥进行搜寻,再解密部分密文从而得到多重集,同时检验前一步骤中获取的多重集是不是在预计算阶段的杂凑表中。如果在,那么猜测出来的密钥值在很大程度上是无误的。文中改进的 6 轮 Square 密码算法的中间相遇攻击使用的多重集为 $(X_5^0[0] \oplus X_5^i[0], X_5^1[0] \oplus X_5^i[0], \cdots, X_5^{255}[0] \oplus X_5^i[0])$,它对应的 δ 集是 $\{w_0^0, w_0^1, \cdots, w_0^{255}\}$,最终将攻击预计算的存储复杂度降为 $2^{10 \times 8} \times 2^8 \times 8/2^7 = 2^{84}$。

根据 6 轮 Square 的中间相遇攻击过程可以看出,主要的时间复杂度是对选择的明文进行加密,所以还可以通过减少选择明文的数量以降低时间复杂度。在图 1 对 6 轮 Square 攻击的截断差分特性中,因为对 w_0 活动字节的差分位置有 0、1、2、3 这 4 个位置,每个选择可构造 2^2 个对应差分表,而 x_5 处的活动字节也会有 0、1、2、3 这四个位置,所以就可以构造出 $4 \times 4 = 16$ 条不同的区分器链,从而可将数据复杂度减少 2^4,但需要额外的预计算表,即存储复杂度则会增加 2^4。

综上,整个攻击的时间复杂度为 2^{109},数据复杂度降为 2^{109},存储复杂度为 2^{84}。

4 结束语

利用差分枚举技术和反弹式思想,文中构造了 Square 算法的 4 轮区分器,又通过多重集降低了预计算的存储复杂度以及通过减少明文数量降低了数据复杂度。研究了 Square 密码算法的中间相遇攻击技术,首次给出了 6 轮 Square 算法的中间相遇攻击,整个攻击的数据复杂度为 2^{109},时间复杂度 2^{109},存储复杂度为 2^{84}。

参考文献:

[1] DAEMEN J, KNUDSEN L, RIJMEN V. The block cipher SQUARE [C] //FSE 1997. Berlin: Springer, 1997: 150-153.

[2] KATZ J, LINDELL Y. Introduction to modern cryptography: principles and protocols [M]. [s. l.]: Chapman and Hall/CRC, 2007.

[3] 冯国登,吴文玲. 分组密码的分析和设计 [M]. 北京: 清华大学出版社, 2000.

[4] 李 超,孙 兵. 分组密码的攻击方法与实例分析 [M]. 北京: 科学出版社, 2010.

[5] DERBEZ P, PERRIN L. Meet-in-the-Middle attacks and structural analysis of round-reduced PRINCE [C] //FSE 2015. Istanbul, Turkey: [s. n.], 2015: 190-216.

[6] BIRYUKOV A, DERBEZ P, PERRIN L. Differential analysis and Meet-in-the-Middle attack against round-reduced TWINE [C] //FSE 2015. Berlin: Springer, 2015: 3-27.

[7] 崔竞一,郭建胜,刘翼鹏. Crypton 算法的不可能差分分析 [J]. 计算机研究与发展, 2017, 54(7): 1525-1536.

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

[9] DEMIRCI H, SELCUK A A. A meet-in-the-middle attack on 8-round AES [C] //FSE 2008. [s. l.]: [s. n.], 2008: 116-126.

[10] DUNKELMAN O, KELLER N, SHAMIR A. Improved single-key attacks on 8-round AES [C] //ASIACRYPT 2010. [s. l.]: [s. n.], 2010: 158-176.

[11] 李永光,曾 光,韩文报. 11 轮 3D 密码算法的中间相遇攻击 [J]. 信息工程大学学报, 2015, 16(2): 133-138.

[12] 刘 超,廖福成,卫宏儒. 对简化轮数的 Crypton 算法的中间相遇攻击 [J]. 软件工程与应用, 2012, 1(2): 17-23.

[13] LIN Li, WU Wenling. Improved meet-in-the-middle attacks on reduced-round Kalyna-128/256 and Kalyna-256/512 [J]. Designs Codes & Cryptography, 2017, 86(4): 1-21.

[14] 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.

[15] 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(6): 132-139.