

FOX 算法的中间相遇攻击

李荣佳, 金晨辉

(解放军信息工程大学三院, 河南 郑州 450002)

摘 要: 研究了 FOX 分组密码算法在中间相遇攻击下的安全性。首先, 分别构造了 FOX64 和 FOX128 的 3 轮中间相遇区分器, 实施了 6 轮中间相遇攻击, 得到对 6 轮 FOX64 和 FOX128 较好的攻击结果。其次, 将 FOX128 的中间相遇区分器扩展到 4 轮, 并结合时间存储数据折衷的方法, 攻击了 7 轮 FOX128, 与已有的攻击结果相比, 攻击的时间复杂度和存储复杂度略大, 而数据复杂度明显降低。

关键词: 分组密码; 密码分析; 中间相遇攻击; FOX 算法

中图分类号: TP918.1

文献标识码: A

Meet-in-the-middle attacks on FOX block cipher

LI Rong-jia, JIN Chen-hui

(The Third College, PLA Information Engineering University, Zhengzhou 450002, China)

Abstract: The security of the block cipher FOX against meet-in-the-middle attack was analyzed. Firstly, 3-round meet-in-the-middle distinguishers was constructed and 6-round meet-in-the-middle attacks for FOX64 and FOX128 was proposed. The two attacks were better attacks for 6-round FOX64 and FOX128, respectively. Secondly, the meet-in-the-middle distinguisher was extended of FOX128 to 4 rounds and proposed 7-round meet-in-the-middle attack combined with time/memory/data tradeoff. Compared to the currently known attacks on 7-round FOX128, The attack has a greater time and memory complexity, however the data complexity is much smaller.

Key words: block cipher, cryptanalysis, meet-in-the-middle attack, FOX

1 引言

FOX 密码算法^[1]是由 Junod 和 Vaudenay 在 2004 年提出的系列分组密码算法, 分组规模可为 64 bit 和 128 bit, 分别记为 FOX64 和 FOX128。FOX 密码的整体结构采用 Lai-Massey 结构, 其轮函数采用 SPS 结构。FOX 密码算法在各平台都具有不错的性能, 广泛地应用于欧洲有线电视等安全产品中。

对 FOX 算法的主要攻击方法有积分攻击、不可能差分攻击、差分—碰撞攻击等。文献[2]利用 3 轮积分区分器分析了 FOX 算法。文献[3]分析了 FOX 算法抵抗不可能差分攻击的能力。文献[4]构造了 4 轮区分器并给出了对 FOX 算法的差分—碰撞攻击结果。文献[5,6]分别对 FOX64 算法进行了零相

关—积分分析和多维零相关线性分析。在 2014 年 FSE 会议上, 文献[7]提出的对 FOX 算法的全子密钥恢复攻击, 目前取得对 FOX 算法的较好攻击结果。

中间相遇攻击由 Diffie 和 Hellman 在分析 DES 算法的安全性时首次提出。近几年, 中间相遇攻击被应用于 AES 算法的分析中, 并得到了较好的攻击结果。在文献[8]中, Demirci 和 Selçuk 首次将中间相遇攻击用于分析 AES, 利用 4 轮中间相遇区分器攻击了 7 轮 AES-192 和 8 轮 AES-256。文献[9]有效地降低了 Demirci 和 Selçuk 攻击的存储和时间复杂度。在 2013 年欧密会上, Derbez 等^[10]利用中间相遇攻击取得了在单密钥条件下对 AES-128 较好的攻击结果。文献[11]利用中间相遇攻击, 结合密钥制约关系, 攻击了 9 轮 AES-192。

收稿日期: 2015-06-26; 修回日期: 2016-01-20

基金项目: 国家自然科学基金资助项目 (No.61272488, No.61402523)

Foundation Item: The National Natural Science Foundation of China (No.61272488, No.61402523)

本文着重研究对 FOX 算法的中间相遇攻击。首先, 本文通过构造 3 轮的中间相遇区分器, 对 6 轮 FOX64 和 FOX128 算法实施了攻击, 得到对 6 轮 FOX64 和 FOX128 较好的攻击结果。其次, 对于 FOX128, 本文将其中间相遇攻击区分器扩展到 4 轮, 并结合时间存储数据折衷的方法, 对 7 轮 FOX128 实施了攻击, 与已有的攻击结果相比, 此攻击的时间复杂度和存储复杂度略大, 而数据复杂度明显降低。表 1 和表 2 将本文对 FOX 算法的攻击结果与此前的主要攻击结果进行了对比。

表 1 FOX64 的主要分析结果

分析方法	轮数	时间复杂度	存储复杂度	数据复杂度	文献
积分攻击	6	$2^{173.4}$	—	2^9	文献[2]
不可能差分	6	2^{133}	—	2^{40}	文献[3]
差分—碰撞	6	$2^{170.8}$	—	7	文献[4]
全子密钥恢复	6	2^{124}	2^{124}	17	文献[7]
多维零相关	6	$2^{119.4}$	—	$2^{60.1}$	文献[5]
零相关—积分	6	$2^{116.7}$	—	2^{50}	文献[6]
中间相遇	6	$2^{114.4}$	$2^{89.9}$	16	本文

注: —表示文中没有给出, 下同。

表 2 FOX128 的主要分析结果

分析方法	轮数	时间复杂度	存储复杂度	数据复杂度	文献
积分攻击	5	$2^{205.6}$	—	2^9	文献[2]
不可能差分	5	2^{135}	—	2^{72}	文献[3]
差分—碰撞	5	$2^{204.5}$	—	11	文献[4]
全子密钥恢复	6	2^{221}	2^{221}	26	文献[7]
中间相遇	6	$2^{211.1}$	$2^{153.6}$	25	本文
全子密钥恢复	7	2^{242}	2^{242}	2^{63}	文献[7]
中间相遇	7	$2^{248.8}$	$2^{246.6}$	$2^{42.6}$	本文

2 预备知识

2.1 符号表示

- $x||y$: x 与 y 级联。
- P^i : 第 i 个明文。
- X_j^i : 第 i 个明文在第 j 轮所对应的中间状态。
- ΔX_j^i : X_j^i 与 X_j^0 的差分。
- S: FOX 密码的 S 盒运算。
- $X[i]$: X 的第 i 个字节。
- $X[i, \dots, j]$: X 的第 i 个到第 j 个字节。
- $X[i, \dots, j] = Y[i, \dots, j]$: X 的第 i 个到第 j 个字节

与 Y 的第 i 个到第 j 个字节对应相等。

2.2 FOX64 密码简介

FOX64 采用了 16 轮迭代的 Lai-Massey 结构, 其第 i 轮的 64 bit 输入可以表示为 2 个 32 bit $L_{i-1} || R_{i-1}$ 。类似地, 第 i 轮的 64 bit 的子密钥 K_i 也可以表示为 2 个 32 bit $LK_i || RK_i$, FOX64 的具体结构如图 1 所示。对于 $x, y \in \{0, 1\}^{16}$, 令 $\text{or}(x || y) = (y || x \oplus y)$ 。设 f_{32} 为其轮函数, 则

$$L_i || R_i = \text{or}(L_{i-1} \oplus f_{32}(L_{i-1} \oplus R_{i-1}, K_i) || (R_{i-1} \oplus f_{32}(L_{i-1} \oplus R_{i-1}, K_i)))$$

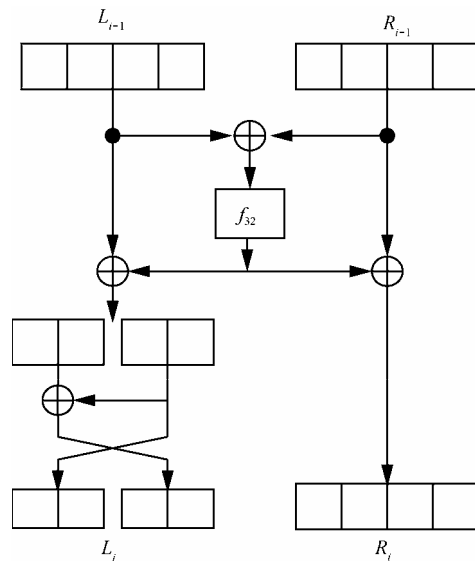


图 1 1 轮 FOX64

轮函数 f_{32} 采用 SPS 结构, 包括子密钥加、代替变换 sigma_4 和扩散变换 mu_4 这 3 个变换, 可以表示为

$$f_{32}(a, K_i) = \text{sigma}_4(\text{mu}_4(\text{sigma}_4(a \oplus LK_i) \oplus RK_i) \oplus LK_i)$$

代替变换 sigma_4 由 4 个 8 bit S 盒并置而成, 扩散变换 mu_4 是一个 4×4 的 MDS 矩阵运算。

2.3 FOX128 密码简介

与 FOX64 类似, FOX128 也采用 16 轮迭代的 Lai-Massey 结构, 其第 i 轮的 128 bit 输入表示为 4 个 32 bit $LL_{i-1} || RL_{i-1} || LR_{i-1} || RR_{i-1}$, 128 bit 的子密钥 K_i 表示为 2 个 64 bit $LK_i || RK_i$ 。FOX128 的具体结构如图 2 所示, 设 f_{64} 为其轮函数, 则

$$LL_i || LR_i = \text{or}(LL_{i-1} \oplus \phi_L) || (LR_{i-1} \oplus \phi_L)$$

$$RL_i || RR_i = \text{or}(RL_{i-1} \oplus \phi_R) || (RR_{i-1} \oplus \phi_R)$$

其中, $\phi_L || \phi_R = f_{64}((LL_{i-1} \oplus LR_{i-1}) || (RL_{i-1} \oplus RR_{i-1}), K_i)$ 。

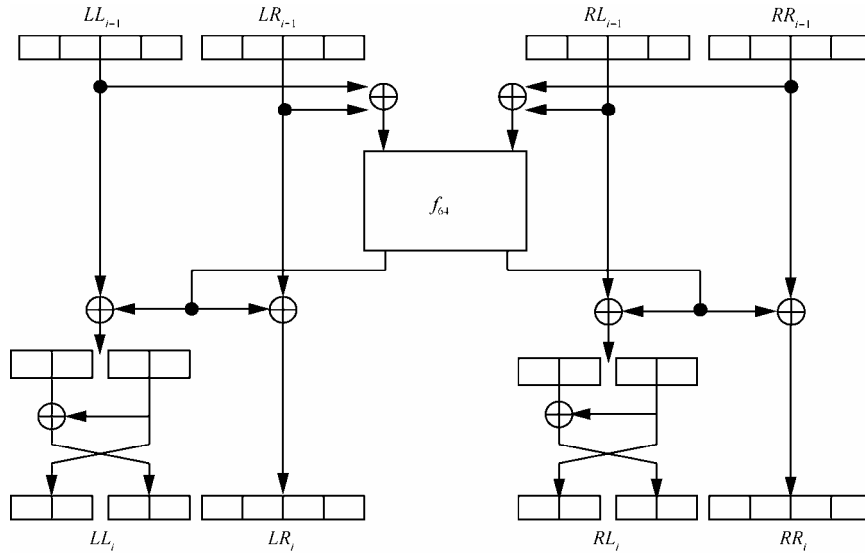


图 2 1 轮 FOX128

轮函数 f_{64} 与 f_{32} 类似, 采用 SPS 结构, 包括子密钥加、代替变换 σ_8 和扩散变换 μ_8 这 3 个变换, 可以表示为

$$f_{64}(b, K_i) = \sigma_8(\mu_8(\sigma_8(b \oplus LK_i)) \oplus RK_i) \oplus LK_i$$

代替变换 σ_8 由 8 个 8 bit S 盒并置而成, 扩散变换 μ_8 是一个 8×8 的 MDS 矩阵运算。

本文把轮函数 f_{32} 和 f_{64} 的输入记作 x , 经过第 1 层子密钥加、第 1 层代替变换、扩散变换、第 2 层子密钥加、第 2 层代替变换、第 3 层子密钥加后的状态分别记作 y 、 z 、 w 、 q 、 r 、 s 。

一轮 FOX64 算法具有如下性质。

性质 1^[6] 设 1 轮 FOX64 的输入为 $L_{i-1} \parallel R_{i-1}$, 输出为 $L_i \parallel R_i$, 则有

$$L_{i-1}[0] \oplus R_{i-1}[0] = L_i[2] \oplus L_i[0] \oplus R_i[0]$$

$$L_{i-1}[2] \oplus R_{i-1}[2] = L_i[0] \oplus R_i[2]$$

类似地, FOX128 算法具有如下性质。

性质 2^[6] 设 1 轮 FOX128 的输入为 $LL_{i-1} \parallel RL_{i-1} \parallel LR_{i-1} \parallel RR_{i-1}$, 输出为 $LL_i \parallel LR_i \parallel RL_i \parallel RR_i$, 则有

$$LL_{i-1}[0] \oplus LR_{i-1}[0] = LL_i[2] \oplus LL_i[0] \oplus LR_i[0]$$

$$LL_{i-1}[2] \oplus LR_{i-1}[2] = LL_i[0] \oplus LR_i[2]$$

3 FOX64 的中间相遇攻击

3.1 FOX64 的 3 轮区分器

本文给出 FOX64 的 3 轮中间相遇区分器, 如图 3 所示, 其中白块表示差分为 0, 黑块表示可以求出的差分, 下同。

定理 1 给定 FOX64 的 16 个明文 $\{P^0, P^1, \dots, P^{15}\}$, 满足 $P^i[2] = P^i[6] = i$, $P^j[0, 1, 3, 4, 5, 7] = P^0[0, 1, 3, 4, 5, 7]$ ($0 \leq i, j \leq 15$)。若对这 16 个明文进行 3 轮 FOX64 加密, 则有序序列 $(L_3^i[0, 2] \oplus R_3^i[0, 2] \oplus L_2^i[0, 2] \oplus R_2^i[0, 2], L_3^i[0, 2] \oplus R_3^i[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2], \dots, L_3^i[0, 2] \oplus R_3^i[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2])$ 只有 2^{88} 种可能的取值。

证明 序列 $(L_3^i[0, 2] \oplus R_3^i[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2], L_3^i[0, 2] \oplus R_3^i[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2], \dots, L_3^i[0, 2] \oplus R_3^i[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2])$ 由如下 11 个字节决定: $y_2^0[0] \parallel q_2^0[0, 1, 2, 3] \parallel y_3^0[0, 1, 2, 3] \parallel q_3^0[0, 2]$ 。

对于 $0 \leq i, j \leq 15$, 满足 $P^i[2] = P^i[6] = i$, $P^i[0, 1, 3, 4, 5, 7] = P^0[0, 1, 3, 4, 5, 7]$, 有 $\Delta P^i = 00i000i0$ 。因此求出 $\Delta x_1^i = 0$, $\Delta s_1^i = 0$, 进一步可得 $\Delta L_1^i = \text{or}(00i0) = i0i0$, $\Delta R_1^i = 00i0$ 。因为 $\Delta y_2^i = \Delta x_2^i = \Delta L_1^i \oplus \Delta R_1^i = i000$, 所以 $\Delta z_2^i[1, 2, 3] = 000$ 。由于 $\Delta y_2^0[0]$ 已知, 利用 $\Delta z_2^i[0] = S(y_2^i[0]) \oplus S(\Delta y_2^i[0] \oplus y_2^0[0])$, 求出 $\Delta z_2^i[0]$ 。又因为扩散变换 μ_4 是线性变换, 所以求出 Δq_2^i 。由于 $q_2^0[0, 1, 2, 3]$ 已知, 利用 $\Delta s_2^i[t] = \Delta r_2^i[t] = S(q_2^i[t]) \oplus S(\Delta q_2^i[t] \oplus q_2^0[t])$ ($t = 0, 1, 2, 3$), 求出 Δs_2^i 。于是利用 $\Delta L_2^i = \text{or}(\Delta L_1^i \oplus \Delta s_2^i)$ 和 $\Delta R_2^i = \Delta R_1^i \oplus \Delta s_2^i$, 可以求出 $\Delta L_2^i \parallel \Delta R_2^i$ 。类似地, 本文求得 Δx_3^i 和 Δy_3^i , 由于 $y_3^0[0, 1, 2, 3]$ 已知, 可以利用 $\Delta z_3^i[t] = S(y_3^i[t]) \oplus S(\Delta y_3^i[t] \oplus y_3^0[t])$ ($t = 0, 1, 2, 3$), 求出 Δz_3^i 。因为扩散变换 μ_4 是线性变换, 所以求出 Δq_3^i 。由于 $q_3^0[0, 2]$ 已知, 利用 $\Delta s_3^i[t] = S(q_3^i[t]) \oplus S(\Delta q_3^i[t] \oplus q_3^0[t])$, 求出 $\Delta s_3^i[0, 2]$ 。进一步利用 $\Delta L_3^i[0] = \Delta L_2^i[2] \oplus \Delta s_3^i[2]$,

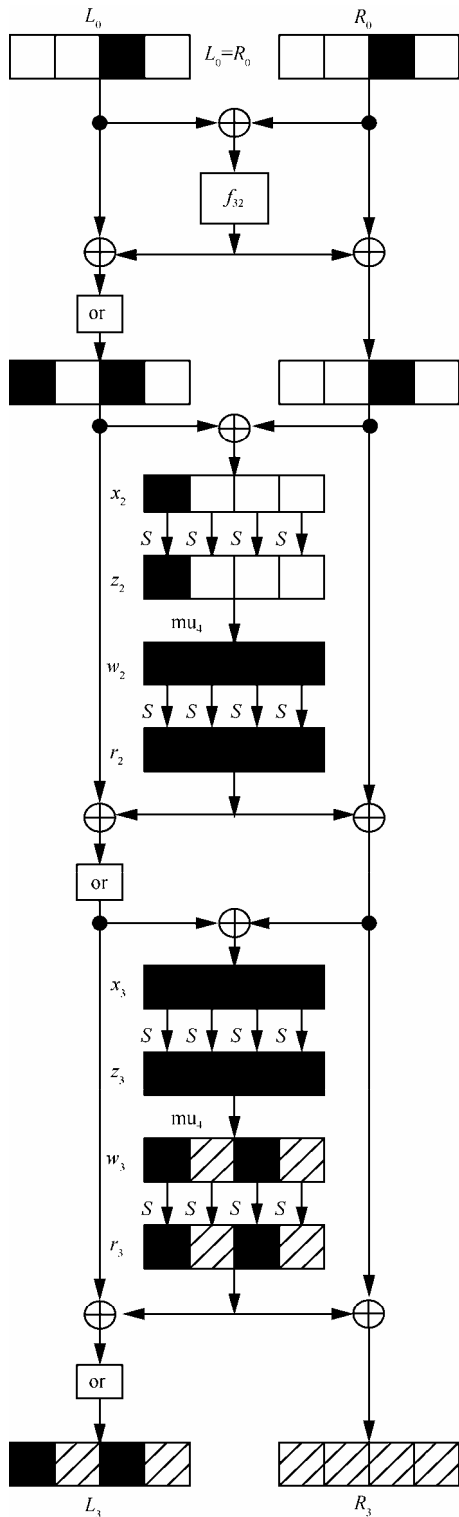


图 3 FOX64 的 3 轮中间相遇区分器

$\Delta L_3^i[2] = \Delta L_2^i[0] \oplus \Delta S_3^i[0] \oplus \Delta L_2^i[2] \oplus \Delta S_3^i[2]$ 和 $\Delta R_3^i[0, 2] = \Delta R_2^i[0, 2] \oplus \Delta S_3^i[0, 2]$, 求出 $\Delta L_3^i[0, 2] \parallel \Delta R_3^i[0, 2]$ 。最后, 本文求出序列 $(L_3^1[0, 2] \oplus R_3^1[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2], L_3^2[0, 2] \oplus R_3^2[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2], \dots, L_3^{15}[0, 2] \oplus R_3^{15}[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2])$, 因此该序列

只有 2^{88} 种可能的取值。

3.2 FOX64 的 6 轮中间相遇攻击

攻击分为 2 阶段: 预计算阶段和在线阶段。

预计算阶段: 根据定理 1, 计算序列 $(L_3^1[0, 2] \oplus R_3^1[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2], L_3^2[0, 2] \oplus R_3^2[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2], \dots, L_3^{15}[0, 2] \oplus R_3^{15}[0, 2] \oplus L_3^0[0, 2] \oplus R_3^0[0, 2])$ 的 2^{88} 种可能取值, 并存储在预计算表 H_1 中。

在线阶段: 在线阶段的具体攻击步骤如下。

步骤 1 选择 16 个明文 $\{P^0, P^1, \dots, P^{15}\}$, 满足 $P^i[2] = P^i[6] = i$, $P^i[0, 1, 3, 4, 5, 7] = P^0[0, 1, 3, 4, 5, 7]$ ($0 \leq i, j \leq 15$), 并获取相应的密文。

步骤 2 猜测子密钥 $LK_5 \parallel RK_5[0, 2] \parallel LK_6 \parallel RK_6$, 脱密 16 个密文, 得到 $L_4^i[0, 2] \parallel R_4^i[0, 2]$ 。

步骤 3 由性质 1, 计算 $L_3^i[0, 2] \oplus R_4^i[0, 2]$, 进而得到序列 $(L_3^1[0, 2] \oplus R_4^1[0, 2] \oplus L_3^0[0, 2] \oplus R_4^0[0, 2], L_3^2[0, 2] \oplus R_4^2[0, 2] \oplus L_3^0[0, 2] \oplus R_4^0[0, 2], \dots, L_3^{15}[0, 2] \oplus R_4^{15}[0, 2] \oplus L_3^0[0, 2] \oplus R_4^0[0, 2])$ 。

步骤 4 检测步骤 3 中求得的序列是否在预计算表 H_1 中。若在, 则判定猜测的密钥为正确密钥; 否则, 判定为错误密钥。故一个错误密钥通过检测的概率为 $\frac{2^{88}}{2^{15 \times 2 \times 8}} = 2^{-152}$, 最终保留的密钥个数为 $1 + 2^{112} \times 2^{-152} \approx 1$ 。

预计算阶段的时间复杂度大约为 $\frac{2^{88} \times 2^4 \times 2}{6} \approx 2^{90.4}$ 次 6 轮 FOX64 加密, 在线阶段的时间复杂度大约为 $\frac{2^{112} \times 2^4 \times 2}{6} = 2^{114.4}$ 次 6 轮 FOX64 加密, 故攻击的时间复杂度约等于在线阶段的时间复杂度。攻击的存储复杂度大约为 $\frac{2^{88} \times 240}{64} = 2^{89.9}$ 个 64 bit。攻击所需的数据量为 16 个选择明文。

4 FOX128 的中间相遇攻击

4.1 FOX128 的 3 轮区分器

本文给出 FOX128 的 3 轮中间相遇区分器, 如图 4 所示。

定理 2 给定 FOX128 的 25 个明文 $\{P^0, P^1, \dots, P^{24}\}$ 满足 $P^i[2] = P^i[6] = i$, $P^i[0, 1, 3, 4, 5, 7, \dots, 15] = P^0[0, 1, 3, 4, 5, 7, \dots, 15]$ ($0 \leq i, j \leq 24$)。若对这 25 个明文进行 3 轮 FOX128 加密, 则有序序列 $(LL_3^1[0, 2] \oplus LR_3^1[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2], LL_3^2[0, 2] \oplus LR_3^2[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2], \dots, LL_3^{24}[0, 2] \oplus LR_3^{24}[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2])$

$LR_3^{24}[0,2] \oplus LL_3^0[0,2] \oplus LR_3^0[0,2]$) 只有 2^{152} 种可能的取值。

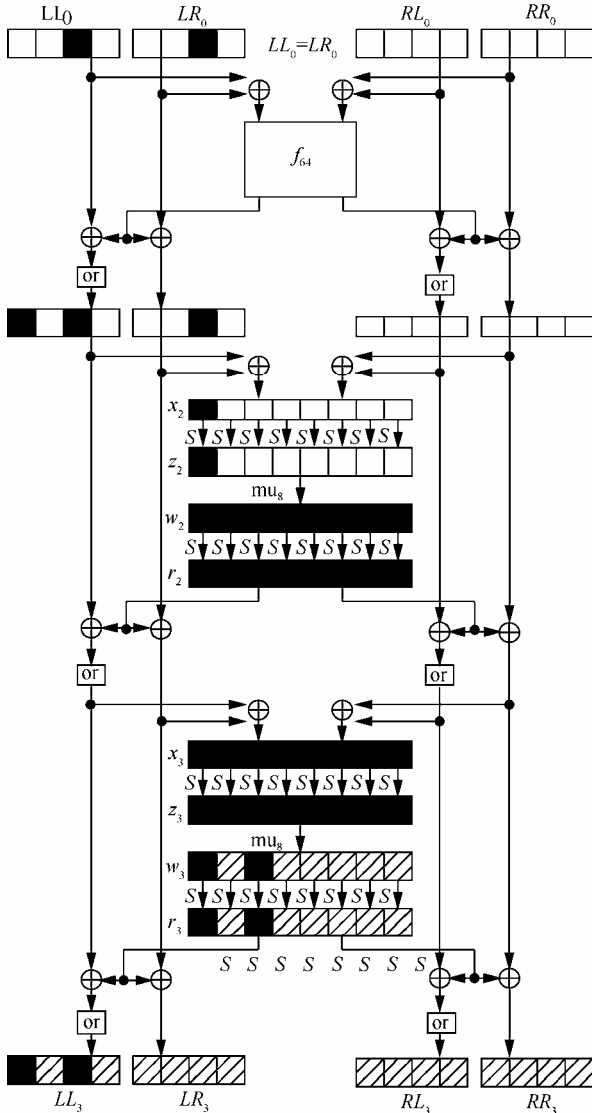


图 4 FOX128 的 3 轮中间相遇区分器

证明 序列 $(LL_3^1[0,2] \oplus LR_3^1[0,2] \oplus LL_3^0[0,2] \oplus LR_3^0[0,2], LL_3^2[0,2] \oplus LR_3^2[0,2] \oplus LL_3^1[0,2] \oplus LR_3^1[0,2], \dots, LL_3^{24}[0,2] \oplus LR_3^{24}[0,2] \oplus LL_3^0[0,2] \oplus LR_3^0[0,2])$ 由如下 19 个字节决定: $y_2^0[0] \parallel q_2^0 \parallel y_3^0 \parallel q_3^0[0,2]$ 。

对于 $0 \leq i, j \leq 24$, 满足 $P^i[2] = P^j[6] = i$, $P^i[0, 1, 3, 4, 5, 7, \dots, 15] = P^j[0, 1, 3, 4, 5, 7, \dots, 15]$, 本文有 $\Delta P^i = 00i000i000000000$ 。因此求出 $\Delta x_1^i = 0$, $\Delta s_1^i = 0$ 。进一步可得, $\Delta LL_1^i = \text{or}(00i0) = i0i0$, $\Delta LR_1^i = 00i0$, $\Delta RL_1^i = \Delta RR_1^i = 0000$ 。因为 $\Delta y_2^i = \Delta x_2^i = i0000000$, 所以 $\Delta z_2^i[1, 2, 3, 4, 5, 6, 7] = 00000000$ 。由于 $\Delta y_2^0[0]$ 已知, 利用 $\Delta z_2^i[0] = S(y_2^0[0]) \oplus S(\Delta y_2^i[0]) \oplus$

$y_2^0[0]$, 求出 $\Delta z_2^i[0]$ 。又因为扩散变换 mu_8 是线性变换, 所以求出 Δq_2^i 。由于 q_2^0 已知, 利用 $\Delta s_2^i[t] = \Delta r_2^i[t] = S(q_2^0[t]) \oplus S(\Delta q_2^i[t] \oplus q_2^0[t])$ ($t = 0, 1, 2, 3, 4, 5, 6, 7$), 求出 Δs_2^i 。进一步求出 $\Delta LL_2^i \parallel \Delta LR_2^i \parallel \Delta RL_2^i \parallel \Delta RR_2^i$ 。类似地, 本文求得 Δx_3^i 和 Δy_3^i , 由于 y_3^0 已知, 利用 $\Delta z_3^i[t] = S(y_3^0[t]) \oplus S(\Delta y_3^i[t] \oplus y_3^0[t])$, 求出 Δz_3^i 。因为扩散变换 mu_8 是线性变换, 所以求出 Δq_3^i 。由于 $q_3^0[0, 2]$ 已知, 求出 $\Delta s_3^i[0, 2]$ 。进一步求出 $\Delta LL_3^i[0, 2] \parallel \Delta LR_3^i[0, 2]$ 。最后, 本文求出序列 $(LL_3^1[0, 2] \oplus LR_3^1[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2], LL_3^2[0, 2] \oplus LR_3^2[0, 2] \oplus LL_3^1[0, 2] \oplus LR_3^1[0, 2], \dots, LL_3^{24}[0, 2] \oplus LR_3^{24}[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2])$, 因此该序列只有 2^{152} 种可能的取值。

4.2 FOX128 的 6 轮中间相遇攻击

攻击分为 2 阶段: 预计算阶段和在线阶段。

预计算阶段: 根据定理 2, 计算序列 $(LL_3^1[0, 2] \oplus LR_3^1[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2], LL_3^2[0, 2] \oplus LR_3^2[0, 2] \oplus LL_3^1[0, 2] \oplus LR_3^1[0, 2], \dots, LL_3^{24}[0, 2] \oplus LR_3^{24}[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2])$ 的 2^{152} 种可能取值, 并存储在预计算表 H_2 中。

在线阶段: 在线阶段的具体攻击步骤如下。

步骤 1 选择 25 个明文 $\{P^0, P^1, \dots, P^{24}\}$, 满足 $P^i[2] = P^j[6] = i$, $P^i[0, 1, 3, 4, 5, 7, \dots, 15] = P^j[0, 1, 3, 4, 5, 7, \dots, 15]$ ($0 \leq i, j \leq 24$), 并获取相应的密文。

步骤 2 猜测子密钥 $LK_5 \parallel RK_5[0, 2] \parallel LK_6 \parallel RK_6$, 脱密 25 个密文, 得到 $LL_4^i[0, 2] \parallel LR_4^i[0, 2]$ 。

步骤 3 由性质 2 计算 $LL_3^i[0, 2] \oplus LR_3^i[0, 2]$, 进而得到序列 $(LL_3^1[0, 2] \oplus LR_3^1[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2], LL_3^2[0, 2] \oplus LR_3^2[0, 2] \oplus LL_3^1[0, 2] \oplus LR_3^1[0, 2], \dots, LL_3^{24}[0, 2] \oplus LR_3^{24}[0, 2] \oplus LL_3^0[0, 2] \oplus LR_3^0[0, 2])$ 。

步骤 4 检测步骤 3 中求得的序列是否在预计算表 H_2 中。若在, 则判定猜测的密钥为正确密钥; 否则, 判定为错误密钥。故一个错误密钥通过检测的概率为 $\frac{2^{152}}{2^{24} \times 2^{28}} = 2^{-232}$, 最终保留的密钥个数为 $1 + 2^{208} \times 2^{-232} \approx 1$ 。

预计算阶段的时间复杂度大约为 $\frac{2^{152} \times 25 \times 2}{6} = 2^{155.1}$ 次 6 轮 FOX128 加密, 在线阶段的时间复杂度大约为 $\frac{2^{208} \times 25 \times 2}{6} = 2^{211.1}$ 次 6 轮 FOX128 加密, 故攻击的时间复杂度约等于在线阶段的时间复杂度。

攻击的存储复杂度大约为 $\frac{2^{152} \times 384}{128} = 2^{153.6}$ 个 128 bit。

攻击所需的数据量为 25 个选择明文。

4.3 FOX128 的 7 轮中间相遇攻击

在 FOX128 的 3 轮中间相遇区分器的中间增加 1 轮，本文得到如下的 4 轮中间相遇区分器。

定理 3 给定 FOX128 的 25 个明文 $\{P^0, P^1, \dots, P^{24}\}$ ，满足 $P^i[2] = P^i[6] = i$ ， $P^j[0,1,3,4,5,7, \dots, 15] = P^0[0,1,3,4,5,7, \dots, 15]$ ($0 \leq i, j \leq 24$)。若对这 25 个明文进行 4 轮 FOX128 加密，则序列 $(LL_4^1[0,2] \oplus LR_4^1[0,2] \oplus LL_4^0[0,2] \oplus LR_4^0[0,2], LL_4^2[0,2] \oplus LR_4^2[0,2] \oplus LL_4^0[0,2] \oplus LR_4^0[0,2], \dots, LL_4^{24}[0,2] \oplus LR_4^{24}[0,2] \oplus LL_4^0[0,2] \oplus LR_4^0[0,2])$ 只有 2^{280} 种可能的取值。

如果本文采用 4.2 节中的方法，在预计算阶段计算所有的 2^{280} 种可能的取值，那么预计算阶段的时间复杂度大约为 $\frac{2^{280} \times 25 \times 3}{7} = 2^{283.4}$ 次 7 轮

FOX128 加密，存储复杂度大约为 $\frac{2^{280} \times 384}{128} = 2^{281.6}$

个 128 bit，这一复杂度超出了穷举攻击的复杂度。因此，本文利用时间存储数据折衷的方法，预计算阶段，只计算并存储序列的 2^{280} 种可能取值中的 $\frac{1}{2^{35}}$ 。同时，在线阶段，本文需要选择大约 $25 \times 2^{38} \approx 2^{42.6}$ 个明文，重复 2^{38} 次攻击。于是正确密钥可以在预计算表中找到匹配的概率变为

$$1 - (1 - 2^{-35})^{2^{38}} \approx 0.9997$$

也就是说，本文攻击成功的概率为 99.97%。经过时间存储数据折衷后，预计算阶段的时间复杂度降低为 $2^{283.4} \times 2^{-35} = 2^{248.4}$ 次 7 轮 FOX128 加密，而在线阶段的时间复杂度提高到 $\frac{2^{208} \times 25 \times 2}{7} \times 2^{38} = 2^{248.8}$

次 7 轮 FOX128 加密，故攻击的时间复杂度约等于在线阶段的时间复杂度。攻击的存储复杂度降低为 $2^{281.6} \times 2^{-35} = 2^{246.6}$ 个 128 bit。攻击所需的数据量变为 $2^{42.6}$ 个选择明文。

5 结束语

本文主要研究了对 FOX64 和 FOX128 算法的中间相遇攻击，首先，本文分别构造了 FOX64 和 FOX128 的 3 轮中间相遇区分器，通过猜测最后 2 轮的部分子密钥，利用 1 轮 FOX 算法的输入与输出

的关系，实施了 6 轮攻击，此攻击是目前为止对 6 轮 FOX64 和 FOX128 较好的攻击结果。其次，本文将 FOX128 的中间相遇攻击区分器扩展到 4 轮，并结合时间存储数据折衷的方法，对 7 轮 FOX128 实施了攻击，与已有的攻击结果相比，此攻击的时间复杂度和存储复杂度略大，而数据复杂度得到明显降低。

参考文献:

- [1] JUNOD P, VAUDENAY S. FOX: a new family of block ciphers[C]// Lecture Notes in Computer Science, 2004. c2004:131-146.
- [2] WU W, ZHANG W, FENG D. Integral cryptanalysis of reduced FOX block cipher[J]. Lecture Notes in Computer Science. 2005, 3935(1): 229-241.
- [3] WU Z M, LAI X J, ZHU B, et al. Impossible differential cryptanalysis of FOX [EB/OL]. IACR Cryptology ePrint Archive, 2009.
- [4] CHEN J, HU Y P, ZHANG Y Y, et al. Differential collision attack on reduced fox block cipher[J]. China Communications. 2012, 9(7):71-76.
- [5] 郭瑞, 金晨辉. 低轮 FOX64 算法的零相关-积分分析[J]. 电子与信息学报, 2015, 37(2): 417-422.
- [6] GUO R, JIN C H. Integral cryptanalysis of reduced round FOX64[J]. Journal of Electronics & Information Technology. 2015, 37(2): 417-422
- [7] 伊文坛, 陈少真. FOX 密码的多维零相关线性分析[J]. 密码学报, 2015, 2(1): 27-39.
- [8] YI W T, CHEN S Z. Multidimensional zero-correlation linear attacks on Fox block cipher[J]. Journal of Cryptologic Research, 2015, 2(1): 27-39.
- [9] ISOBE T, SHIBUTANI K. Improved all-subkeys recovery attacks on FOX, KATAN and SHACAL-2 block ciphers [C]//FSE 2014. c2014: 104-126.
- [10] DEMIRCI H, SELÇUK A. A Meet-in-the-middle attack on 8-round AES[C]//Lecture Notes in Computer Science, Lausanne, Switzerland, c2008:116-126.
- [11] DUNKELMAN O, KELLER N, SHAMIR A. Improved single-key attacks on 8-round AES-192 and AES-256[J]. Journal of Cryptology, 2010, 28(3): 158-176.
- [12] DERBEZ, P, FOUQUE P A, JEAN J. Improved key recovery attacks on reduced-round AES in the single-key setting[J]. Lecture Notes in Computer Science, 2013, 788: 371-387.
- [13] LI L B, JIA K T, WANG X Y. Improved single-key attacks on 9-round AES-192/256[M]//Fast Software Encryption. Springer Berlin Heidelberg, 2014:127-146.

作者简介:



李荣佳 (1991-), 男, 山东泗水人, 解放军信息工程大学硕士生, 主要研究方向为对称密码设计与分析。

金晨辉 (1965-), 男, 河南扶沟人, 解放军信息工程大学教授, 主要研究方向为密码学与信息安全。