低轮 FOX 分组密码的碰撞-积分攻击

吴文玲[1],卫宏儒[2]
(1. 中国科学院软件研究所信息安全国家重点实验室,北京 100080;2. 北京科技大学应用科学学院,北京 100083)

摘 要: FOX 是最近推出的系列分组密码,它的设计思想基于可证安全的研究结果,且在各种平台上的性能优良. 本文利用碰撞攻击和积分攻击相结合的技术分析 FOX 的安全性,结果显示碰撞-积分攻击比积分攻击有效,攻击对 4 轮 FOX64 的计算复杂度是 $2^{45.4}$,对 5 轮 FOX64 的计算复杂度是 $2^{109.4}$,对 6 轮 FOX64 的计算复杂度是 $2^{173.4}$,对 7 轮 FOX64 的计算复杂度是 $2^{237.4}$,且攻击所需数据量均为 2^9;也就是说 4 轮 FOX64/64、5 轮 FOX64/128、6 轮 FOX64/192 和 7 轮 FOX64/256 对本文攻击是不免疫的.

关键词: 分组密码; 攻击; 密钥; 计算复杂度; 数据复杂度
中图分类号: TP391 **文献标识码**: A **文章编号**: 0372-2112 (2005) 07-1307-04

Collision-Integral Attack of Reduced-Round FOX

WU Wen-ling[1], WEI Hong-ru[2]
(1. *State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*;
2. *School of Applied Science, University of Science and Technology Beijing, Beijing*100083, *China*)

Abstract: FOX are a family of block ciphers presented recently, which are based upon some results on proven security and have high performances on various platforms. In this paper, we construct some distinguishers between 3-round FOX and a random permutation of the blocks space. By using collision-searching techniques and integral attack, the distinguishers are used to attack on 4, 5, 6 and 7 rounds of FOX64. The four subkeys of 4-round FOX64 can be recovered with 2^9 chosen plaintexts and $2^{45.4}$ encryptions. The five subkeys of 5-round FOX64 can be recovered with 2^9 chosen plaintexts and $2^{109.4}$ encryptions. The six subkeys of 6-round FOX64 can be recovered with 2^9 chosen plaintexts and $2^{173.4}$ encryptions. The seven subkeys of 7-round FOX64 can be recovered with 2^9 chosen plaintexts and $2^{237.4}$ encryptions. Therefore, 4-round FOX64/64, 5-round FOX64/128, 6-round FOX64/192 and 7-round FOX64/256 are not immune to Collision-Integral attack.

Key words: block cipher; attack; key; data complexity; time complexity

1 引言

FOX[1]是基于 Mediacrypt[2]公司的需求而设计的系列分组密码,分组长度可以为 64 或 128 比特,密钥长度 k 满足 $0 \leq k \leq 256$,且 k 是 8 的倍数. 设计者建议 FOX64/128 和 FOX128/256 的轮数均为 16,其中 FOX64/128 中的 64 代表分组长度,128 代表密钥长度. FOX 的整体结构采用的是 Lai-Massey 方案[3];基于文献[4]对 Lai-Massey 方案的可证安全性的研究,轮函数使用 SPS 结构,并使用了三层密钥加运算. FOX 的另一个特点是它的密钥编排算法;和别的分组密码不同的是 FOX 的密钥编排算法很复杂,每一个子密钥都必须由种子密钥生成,由若干子密钥很难获取种子密钥或其它子密钥的信息. 如此设计的密钥编排算法、整体结构及轮函数保证 FOX 具有很强的可证安全的特性.

由于 FOX 推出不久,因此关于它的安全性分析仅限于设计者的分析结果,FOX 关于差分和线性密码分析的安全性源于它的整体结构、轮函数及 S 盒的特性;文献[1]还分析了 FOX 对统计攻击[5~7]、代数攻击[8]和滑动攻击[9]等的安全性. 积分(Square)攻击[10]是目前对 AES 最有效的攻击方法之一,文献[1]指出积分攻击对 4 轮 FOX64 的计算复杂度是 2^{64},5 轮 FOX64 的计算复杂度是 2^{128},6 轮 FOX64 的计算复杂度是 2^{192},7 轮 FOX64 的计算复杂度是 2^{256};也就是说 4 轮 FOX64/64、5 轮 FOX64/128、6 轮 FOX64/192 和 7 轮 FOX64/256 对积分攻击是安全的. 本文结合碰撞攻击[12]和积分攻击对 FOX 的安全性进行分析,结果显示攻击对 4 轮 FOX64 的计算复杂度是 $2^{45.4}$,对 5 轮 FOX64 的计算复杂度是 $2^{109.4}$,对 6 轮 FOX64 的计算复杂度是 $2^{173.4}$,对 7 轮 FOX64 的计算复杂度是 $2^{237.4}$;也就是说 4 轮 FOX64/64、5 轮 FOX64/128、6 轮 FOX64/192 和 7 轮 FOX64/256 对本文攻击是不免疫的.

2 FOX 分组密码简述

FOX 是一系列分组密码,为了篇幅,这里仅介绍 FOX64,即分组长度为 64 的 FOX,它的轮数随密钥长度的改变而不同,FOX64/k/r 中的 k 是密钥长度、r 是轮数.

收稿日期:2004-09-10;修回日期:2005-01-26
基金项目:国家自然科学基金(No. 60373047, No. 90304007);973 项目(No. 2004CB318004);863 项目(No. 2003AA144030)

2.1 轮函数

$f:\{0,1\}^{32}\times\{0,1\}^{64}\to\{0,1\}^{32}$ $X_{(32)}\times K_{(64)}\to Y_{(32)}$, $K_{(64)}=K_{0(32)}\|K_{1(32)}$, $Y_{(32)}=\text{sigma}\,4(\text{mu}\,4(\text{sigma}\,4(X_{(32)}\oplus K_{0(32)}))\oplus K_{1(32)})\oplus K_{0(32)}$

sigma $4:\{0,1\}^{32}$ 由 4 个 8×8 的 S 盒并置而成.

$$\text{mu}\,4:\{0,1\}^{32}\begin{pmatrix}Y_{0(8)}\\Y_{1(8)}\\Y_{2(8)}\\Y_{3(8)}\end{pmatrix}=\begin{pmatrix}1&1&1&\alpha\\1&z&\alpha&1\\z&\alpha&1&1\\\alpha&1&z&1\end{pmatrix}\begin{pmatrix}X_{0(8)}\\X_{1(8)}\\X_{2(8)}\\X_{3(8)}\end{pmatrix}$$

$z=\alpha^{-1}+1$, α 是不可约多项式的一个根.

2.2 加密和解密算法

64 比特明文 $P_{(64)}$, 经 FOX64/k/r 加密得 64 比特密文 $C_{(64)}$:

$C_{(64)}=\text{Im}id64(\text{Im}or64(\cdots(\text{Im}or64(P_{(64)},K_{0(64)}),\cdots,K_{r-2(64)}),K_{r-1(64)})$

其中 $K_{(64r)}=K_{0(64)}\|K_{1(64)}\|\cdots\|K_{r-1(64)}$ 是种子密钥经密钥编排算法生成的子密钥流.

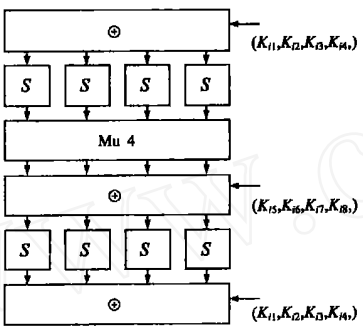

图 1 轮函数

解密为: $P_{(64)}=\text{Im}id64(\text{Im}o64(\cdots(\text{Im}o64(C_{(64)},K_{r-1(64)}),\cdots,K_{1(64)}),K_{0(64)})$

其中 $\text{Im}or64:\{0,1\}^{64}\times\{0,1\}^{64}\to\{0,1\}^{64}$ $X_{64}\times K_{(64)}\to Y_{(64)}$

$X_{(64)}=X_{0(32)}\|X_{1(32)}$, $Y_{(64)}=or(X_{0(32)}\oplus\phi)\|(X_{1(32)}\oplus\phi)$

$\phi=f(X_{0(32)}\oplus X_{1(32)},K_{64})$.

$or:\{0,1\}^{32}\to\{0,1\}^{32}$

$X_{(32)}=X_{0(16)}\|X_{1(16)}\to Y_{(32)}=X_{1(16)}\|X_{0(16)}\oplus X_{1(16)}$

$\text{Im}id64$ 及 $\text{Im}o64$ 和 $\text{Im}or64$ 的区别仅在于 or 处, $\text{Im}id64$ 使用恒等变换, $\text{Im}o64$ 使用 or^{-1}. 本文攻击假定每个子密钥是独立的, 所以不涉及密钥编排算法.

3 三轮区分器

对于明文 P, u, v 和 w 分别表示三轮 FOX 的第一、第二轮和最后的输出. 对任意比特串 g, 用 g_l 表示它的左半边, g_r 表示它的右半边, $\Delta g=g_l\oplus g_r$. 如图 2 所示, 我们选取明文 P, 使得

$$P_l=(c_1,c_2,c_3,x),\quad P_r=(c_1,c_2,c_3,x)$$

其中 c_1,c_2,c_3,x 取自 $\{0,1\}^8$.

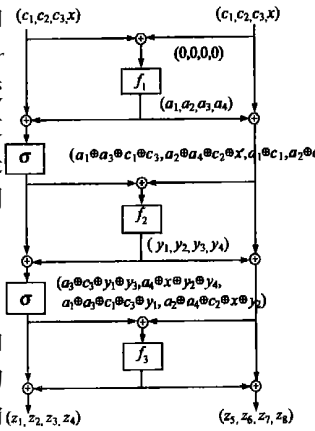

图 2 3轮区分器

第一轮的输出为:

$u_l=(a_3\oplus c_3,a_4\oplus x,a_1\oplus a_3\oplus c_1\oplus c_3,a_2\oplus a_4\oplus c_2\oplus x)$,

$u_r=(a_1\oplus c_1,a_2\oplus c_2,a_3\oplus c_3,a_4\oplus x)$

因此 $\Delta u=(a_1\oplus a_3\oplus c_1\oplus c_3,a_2\oplus a_4\oplus c_2\oplus x,a_1\oplus c_1,a_2\oplus c_2)$

在第二轮中, 轮函数 f_2 做如下变换:

$\Delta u=(a_1\oplus a_3\oplus c_1\oplus c_3,a_2\oplus a_4\oplus c_2\oplus x,a_1\oplus c_1,a_2\oplus c_2)\xrightarrow{f_2}(y_1,y_2,y_3,y_4)$

第二轮的输出为:

$v_l=(a_1\oplus a_3\oplus c_1\oplus c_3\oplus y_3,a_2\oplus a_4\oplus c_2\oplus x\oplus y_4,a_1\oplus c_1\oplus y_1\oplus y_3,a_2\oplus c_2\oplus y_2\oplus y_4,)$,

$v_r=(a_1\oplus c_1\oplus y_1,a_2\oplus c_2\oplus y_2,a_3\oplus c_3\oplus y_3,a_4\oplus x\oplus y_4)$.

因此, $\Delta v=(a_3\oplus c_3\oplus y_1\oplus y_3,a_4\oplus x\oplus y_2\oplus y_4,a_1\oplus a_3\oplus c_1\oplus c_3\oplus y_1,a_2\oplus a_4\oplus c_2\oplus x\oplus y_2)$.

对于三轮 FOX, 最后一轮没有 or 变换, 所以 $\Delta w=\Delta v$. 因此, 我们可以给出如下引理:

引理 令 $P=(L_0,R_0)$ 和 $P^*=(L_0^*,R_0^*)$ 是 3-轮 FOX 的两个输入, 相应的输出为 $C=(w_1,w_2,w_3,w_4,w_5,w_6,w_7,w_8)$ 和 $C^*=(w_1^*,w_2^*,w_3^*,w_4^*,w_5^*,w_6^*,w_7^*,w_8^*)$, 如果 $L_0=R_0$, $L_0^*=R_0^*$, 且 R_0 和 R_0^* 仅第 4 字节不同; 则 C 和 C^* 有如下性质:

(1) $w_3\oplus w_7\neq w_3^*\oplus w_7^*$

(2) $w_1\oplus w_3\oplus w_5\oplus w_7\neq w_1^*\oplus w_3^*\oplus w_5^*\oplus w_7^*$

(3) $w_2\oplus w_4\oplus w_6\oplus w_8\neq w_2^*\oplus w_4^*\oplus w_6^*\oplus w_8^*$

证明: 见图 2. 令 $L_0=R_0=(c_1,c_2,c_3,x)$, $L_0^*=R_0^*=(c_1,c_2,c_3,x^*)$, 且 $x\neq x^*$.

在第一轮中, 轮函数的输入均为 $(0,0,0,0)$, 因此, 对明文 P 和 P^*, 轮函数 f_1 的输出均为 (a_1,a_2,a_3,a_4).

在第二轮中, $\Delta u=(a_1\oplus a_3\oplus c_1\oplus c_3,a_2\oplus a_4\oplus c_2\oplus x,a_1\oplus c_1,a_2\oplus c_2)$, $\Delta u^*=(a_1\oplus a_3\oplus c_1\oplus c_3,a_2\oplus a_4\oplus c_2\oplus x^*,a_1\oplus c_1,a_2\oplus c_2)$, Δu 和 Δu^* 仅有一块不同, 由轮函数的结构(图 1)可推出: $y_1\neq y_1^*$, $y_2\neq y_2^*$, $y_3\neq y_3^*$, $y_4\neq y_4^*$.

在第三轮中, $\Delta w=(a_3\oplus c_3\oplus y_1\oplus y_3,a_4\oplus x\oplus y_2\oplus y_4,a_1\oplus a_3\oplus c_1\oplus c_3\oplus y_1,a_2\oplus a_4\oplus c_2\oplus x\oplus y_2)$, $\Delta w^*=(a_3\oplus c_3\oplus y_1^*\oplus y_3^*,a_4\oplus x^*\oplus y_2^*\oplus y_4^*,a_1\oplus a_3\oplus c_1\oplus c_3\oplus y_1^*,a_2\oplus a_4\oplus c_2\oplus x^*\oplus y_2^*)$, 因此, $w_3\oplus w_7=a_1\oplus a_3\oplus c_1\oplus c_3\oplus y_1\neq a_1\oplus a_3\oplus c_1\oplus c_3\oplus y_1^*=w_3^*\oplus w_7^*$. 同理可得(3)和(4).

4 FOX的碰撞-积分攻击

4.1 四轮 FOX的碰撞-积分攻击

令 t 是第三轮的输出, w 和 t 的区别是: w 没有经过第三轮的 or 变换; z 表示四轮 FOX 的输出, 注意最后一轮没有 or 变换, 所以

$\Delta z=\Delta t=(t_1\oplus t_5,t_2\oplus t_6,t_3\oplus t_7,t_4\oplus t_8)$

$=(w_3\oplus w_5,w_4\oplus w_6,w_1\oplus w_3\oplus w_7,w_2\oplus w_4\oplus w_8)$,

如果是选择明文攻击, 则密文 z 已知, 即 Δz 已知. 进一步, 我们知道 $w_1\oplus w_3\oplus w_7$; 如果我们能从密文预测 w_5, 则可以预测 $w_1\oplus w_3\oplus w_5\oplus w_7$, 利用此我们就可以对四轮 FOX 进行攻击. 如果记 $f_4(\Delta z)=(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$, 则 $w_5=z_5\oplus\alpha_1$; 分析 f_4 的结构, 要想推出 α_1, 需预测 40 比特子密钥, 即第 4 个子密

钥 K_4 的第 1、2、3、4 和 5 字节 $(K_{41}, K_{42}, K_{43}, K_{44}, K_{45})$.

算法 1

第一步,选取 122 个明文 $P^i(1 \le i \le 122)$,使得 $P^i = (0,0,0,x_i,0,0,0,x_i)$, $x_i \in \{0,1\}^8$,相应的密文记为 $z^i(1 \le i \le 122)$.

第二步,对 $(K_{41}, K_{42}, K_{43}, K_{44}, K_{45})$ 的每一个候选值,由 Δz^i 计算 α_1^i,并计算 $\varepsilon^i = z_3^i \oplus z_5^i \oplus z_7^i \oplus \alpha_1^i$. 检查 122 个 ε^i 是否有碰撞,如果有,丢掉相应的候选密钥,否则,输出相应的候选密钥.

第三步,对第二步输出的候选密钥,选择其它的明文,重复第二步,直到输出值唯一.

给 256 个箱子中扔 122 个球,至少存在一个碰撞的概率大于 $1 - e^{-122 \times 121/2^9} > 1 - 2^{-41.5}$,因此,第二步输出错误子密钥的概率小于 $2^{-41.5}$. 因为正确密钥肯定通过,而 $2^{40} - 1$ 个错误密钥中通过碰撞检测的个数平均为 $(2^{40} - 1) \times 2^{-41.5} \approx 0.35$,故通过检测的候选密钥的个数约为 1.35;因此,第三步只需几个明文即可. 攻击需 128 个选择明文,$2^{40} \times 2^7/4 = 2^{45}$ 次加密.

下面进一步预测 K_6. 类似算法 1,只是 $(K_{41}, K_{42}, K_{43}, K_{44}, K_{45})$ 已知,所以第二步的候选值是 2^8 个,所以只需 64 个选择明文,而且可以用算法 1 中的选择明文. 这里利用引理中的第三个不等式,因此计算 $\varepsilon^i = z_4^i \oplus z_6^i \oplus z_8^i \oplus \alpha_2^i$. 攻击需 $2^8 \times 64/4 = 2^{12}$ 次加密.

$(K_{41}, K_{42}, K_{43}, K_{44}, K_{45})$ 已知,可以从 Δz^i 计算 α_1^*, $w_5^i = \alpha_1^i \oplus z_5^i$. 对 K_{47} 的每个候选值,从 Δz^i 计算 α_3^i, $w_7^i = \alpha_3^i \oplus z_7^i$,又 $w_3^i \oplus w_5^i = z_1^i \oplus z_5^i$,所以 $w_3^i \oplus w_7^i = z_1^i \oplus z_7^i \oplus \alpha_1^i \oplus \alpha_3^i$. 这样类似算法 1,通过计算 $\varepsilon^i = z_1^i \oplus z_7^i \oplus \alpha_1^i \oplus \alpha_3^i$ 预测 K_{47}. 攻击需 $2^8 \times 64/4 = 2^{12}$ 次加密.

对于 K_{48},不能用类似的方法预测,我们可以用积分攻击来预测.

算法 2

第一步,选取 256 个明文 $P^i(1 \le i \le 256)$,使得 $P^i = (0,0,0,x_i,0,0,0,x_i)$, $x_i \in \{0,1\}^8$

第二步,对 K_{48} 的每个候选值,计算 $\varepsilon^i = z_2^i \oplus z_8^i \oplus \alpha_2^i \oplus \alpha_4^i$,检查 $\bigoplus_{1 \le i \le 256} \varepsilon^i = 0$ 是否成立,如果否,丢掉相应的候选值,如果是,输出相应的候选值.

第三步,对第二步输出的值,选取另一组明文(比如 $P^i = (c,c,c,x_i,c,c,c,x_i)$,变量 $x_i \in \{0,1\}^8$, c 是常数),重复第二步.

已知 $(K_{41}, K_{42}, K_{43}, K_{44}, K_{45}, K_{46}, K_{47})$ 和密文 z^i,则可计算 α_2^i;又 $z_2^i \oplus \alpha_2^i \oplus z_8^i \oplus \alpha_4^i = w_4^i \oplus w_8^i = a_2 \oplus a_4 \oplus c \oplus x^i \oplus y_2^i$,而当 x^i 遍历 $\{0,1\}^8$ 时,y_2^i 也遍历 $\{0,1\}^8$,因此对正确的子密钥 $\bigoplus_{1 \le i \le 256} \varepsilon^i = 0$. 算法 2 需要不超过 2^9 个选择明文,计算复杂度小于 $2^9 \times 2^8/4 = 2^{15}$ 次加密. 由于算法 2 可以使用算法 1 选好的明文,所以预测 K_4 的数据复杂度为 2^9,计算复杂度为 $2^{45} + 2^{15} + 2^{13}$.

现已知子密钥 K_4,因此,可以从密文做一轮解密得到第三轮(没有经过 *or* 变换)的输出 w. 记 $f_3(\Delta v) = (\beta_1, \beta_2, \beta_3, \beta_4)$,则 $v_1 = w_1 \oplus \beta_1$, $v_2 = w_2 \oplus \beta_2$, $v_3 = w_3 \oplus \beta_3$, $v_4 = w_4 \oplus \beta_4$, $v_5 = w_5 \oplus \beta_1$, $v_6 = w_6 \oplus \beta_2$, $v_7 = w_7 \oplus \beta_3$, $v_8 = w_8 \oplus \beta_4$.

记 $f_2(\Delta u) = (y_1, y_2, y_3, y_4)$,所以 $v_1 = u_3 \oplus y_3$, $v_2 = u_4 \oplus y_4$, $v_3 = u_1 \oplus y_1 \oplus u_3 \oplus y_3$, $v_4 = u_2 \oplus y_2 \oplus u_4 \oplus y_4$, $v_5 = u_5 \oplus y_1$, $v_6 = u_6 \oplus y_2$, $v_7 = u_7 \oplus y_3$, $v_8 = u_8 \oplus y_4$.

而由图 2 可知:$u_1 \oplus u_5 = a_1 \oplus a_3$, $u_2 \oplus u_6 = a_2 \oplus a_4 \oplus c \oplus x$, $u_3 \oplus u_7 = a_1 \oplus c$, $u_4 \oplus u_8 = a_2 \oplus c$.

因此有下等式:

$$w_1 \oplus w_3 \oplus w_5 \oplus \beta_3 = a_1 \oplus a_3 \quad (1)$$

$$w_1 \oplus w_7 \oplus \beta_1 \oplus \beta_3 = a_1 \oplus c \quad (2)$$

$$w_2 \oplus w_8 \oplus \beta_2 \oplus \beta_4 = a_2 \oplus c \quad (3)$$

$$w_2 \oplus w_4 \oplus w_6 \oplus \beta_4 = a_2 \oplus a_4 \oplus c \oplus x \quad (4)$$

由此我们构造预测子密钥 K_3 的算法:

算法 3

第一步,用算法 1 中已选取的 6 个明文 $P^i(1 \le i \le 6)$,使得 $P^i = (0,0,0,x_i,0,0,0,x_i)$, $x_i \in \{0,1\}^8$,相应的密文记为 $z^i(1 \le i \le 6)$,第三轮输出为 $w^i(1 \le i \le 6)$.

第二步,对 $(K_{31}, K_{32}, K_{33}, K_{34}, K_{37})$ 的每一个候选值,由 Δw^i 计算 β_3,并计算

$$\delta = w_1^i \oplus w_3^i \oplus w_5^i \oplus \beta_3$$

检查 6 个 δ 是相同,如果不是,丢掉相应的候选密钥;如果是,输出相应的候选密钥.

第三步,对第二步输出候选值,选择其它的明文,重复第二步,直到输出值唯一.

6 个字节相同的概率是 2^{-40},因此,第三步最多需要 2 个明文. 因此预测 $(K_{31}, K_{32}, K_{33}, K_{34}, K_{37})$ 的数据复杂度为 8,计算复杂度 $8 \times 2^{40}/4 = 2^{41}$.

利用等式(2)和算法 3 预测的 $(K_{31}, K_{32}, K_{33}, K_{34}, K_{37})$,通过计算 $\delta = w_1^i \oplus w_7^i \oplus \beta_1 \oplus \beta_3$ 预测 K_{35},且可以用上面已选的明文,计算复杂度 $4 \times 2^8/4 = 2^8$.

利用等式(4)和算法 3 预测的 $(K_{31}, K_{32}, K_{33}, K_{34})$,通过计算 $\delta = w_2^i \oplus w_4^i \oplus \beta_4$ 预测 K_{38},这里是检查 δ 是否有碰撞,如果有,丢掉相应的候选密钥,否则,输出相应的候选密钥,需要 64 个选择明文,计算复杂度为 $2^6 \times 2^8/4 = 2^{12}$.

利用等式(2)和算法 3 预测的 $(K_{31}, K_{32}, K_{33}, K_{34}, K_{38})$,通过计算 $\delta = w_2^i \oplus w_8^i \oplus \beta_2 \oplus \beta_4$ 来预测 K_{36},且可以用上面选好的明文,计算复杂度为 $4 \times 2^8/4 = 2^8$.

综上所述,预测 K_3 的计算复杂度小于 $2^{41} + 2^{12} + 2^9$. 类似预测 K_2 和 K_1 的计算复杂度均小于 $2^{41} + 2^{12} + 2^9$. 因此,恢复 4 个轮子密钥的计算复杂度小于 $2^{45} + 2^{42} + 2^{41} + 2^{16} < 2^{45.4}$ 次加密,数据复杂度小于 2^9.

4.2 其它低轮 FOX 的碰撞-积分攻击

对 5 轮 FOX64,预测 K_5 子密钥,类似 4.1 的攻击方法,计算复杂度小于 $2^{109.4}$,数据复杂度小于 2^9. 同理,对 6 轮 FOX64,预测 K_5 和 K_6 两轮子密钥,攻击的计算复杂度是 $2^{173.4}$,数据复杂度小于 2^9. 对 7 轮 FOX64,预测 K_5、K_6 和 K_7 三轮子密钥,攻击的计算复杂度是 $2^{237.4}$;数据复杂度小于 2^9.

5 结束语

FOX 是最近推出的系列分组密码,它的设计思想基于一些可证安全的研究结果,它在各种平台上的性能比较好.由于 FOX 公布不久,因此关于它的安全性分析仅限于设计者的分析结果,本文利用碰撞攻击和积分攻击相结合的技术分析 FOX 的安全性,结果显示碰撞-积分攻击比积分攻击有效,攻击对 4 轮 FOX64 的计算复杂度是 $2^{45.4}$,对 5 轮 FOX64 的计算复杂度是 $2^{109.4}$,对 6 轮 FOX64 的计算复杂度是 $2^{173.4}$,对 7 轮 FOX64 的计算复杂度是 $2^{237.4}$,且攻击所需数据量均为 2^9;也就是说 4 轮 FOX64/64、5 轮 FOX64/128、6 轮 FOX64/192 和 7 轮 FOX64/256 对本文攻击是不免疫的.

参考文献:

[1] P Junod, S Vaudenay. FOX: a new family of block ciphers[A]. Selected Areas in Cryptography-SAC 2004[C]. Berlin: Springer-Verlag, 2004. 131 - 146.

[2] Mediacrypt AG[EB/OL]. http://www.mediacrypt.com

[3] X Lai, J Massey. A proposal for a new block encryption standard[A]. Advances in Cryptology-EUROCRYPT'90[C]. Berlin: Springer-Verlag, 1991. 389 - 404.

[4] S Vaudenay. On the lai-massey scheme[A]. Advances in Cryptology-ASIACRYPT'99[C]. Berlin: Springer-Verlag, 1999. 8 - 19.

[5] E Biham, A Biryukov, A Shamir. Enhancing differential-linear cryptanalysis[A]. Advances in Cryptology-ASIACRYPT'02[C]. Berlin: Springer-Verlag, 2002. 254 - 266.

[6] L Knudsen. Truncated and higher order differentials[A]. Fast Software Encryption-FSE'95[C]. Berlin: Springer-Verlag, 1995. 196 - 211.

[7] E Biham, A Biryukov, A Shamir. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials[A]. Advances in Cryptology-EUROCRYPT'99[C]. Berlin: Springer-Verlag, 1999. 12 - 23.

[8] N Courtois, J Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations[A]. Advances in Cryptology-ASIACRYPT'02[C]. Berlin: Springer-Verlag, 2002. 267-287.

[9] A Biryukov, D Wagner. Slide attacks[A]. Fast Software Encryption-FSE'99[C]. Berlin: Springer-Verlag, 1999. 245 - 259.

[10] A Biryukov, D Wagner. Advanced slide attacks[A]. Advances in Cryptology-EUROCRYPT'00[C], Berlin: Springer-Verlag, 2000. 589 - 606.

[11] L Knudsen, D Wagner. Integral cryptanalysis (extended abstract)[A]. Fast Software Encryption-FSE2002[C]. Berlin: Springer-Verlag, 2002. 112 - 127.

[12] Wenling Wu, Dengguo Feng, Hua Chen. Collision attack and pseudrandomness of reduced-round camellia[A]. Selected Areas in Cryptography-SAC 2004[C]. Heidelberg: Springer-Verlag, 2005. 275 - 290.

作者简介:

吴文玲 女,1966 年 8 月出生于陕西省蒲城县,现为中国科学院软件研究所研究员、博导,从事密码学与信息安全的研究工作,主持和参加了多项有关密码学与信息安全方面的国家级和省部级项目,在国内外各种学术会议和学术刊物上发表 80 余篇论文,与他人合作出版著作三部.

卫宏儒 男,1963 年 7 月出生于陕西省扶风县,现为北京科技大学应用科学学院副教授,从事密码学与信息安全方面的研究与教学工作,发表十余篇相关论文.