第 38 卷 第 3 期 计 算 机 工 程 2012 年 2 月
Vol.38 No.3 Computer Engineering February 2012

• 安全技术 • 文章编号：1000—3428(2012)03—0141—04 文献标识码：A 中图分类号：TP309.2

CLEFIA-128 算法的不可能差分密码分析

郑秀林[1,2]，连至助[1,2]，鲁艳蓉[1,2]，袁 征[1]

(1. 北京电子科技学院信息安全系，北京 100070；2. 西安电子科技大学通信工程学院，西安 710071)

摘 要：研究 13 轮 CLEFIA-128 算法，在 9 轮不可能差分攻击的基础上，提出一种未使用白化密钥的不可能差分密码分析方法。猜测每个密钥，筛选满足轮函数中 S 盒输入输出差分对的数据对。利用轮密钥之间的关系减少密钥猜测量，并使用 Early Abort 技术降低计算复杂度。计算结果表明，该方法的数据复杂度和时间复杂度分别为 2^{120} 和 $2^{125.5}$。

关键词：分组密码；CLEFIA-128 算法；密码分析；不可能差分密码分析；Early Abort 技术

Impossible Differential Cryptanalysis of CLEFIA-128 Algorithm

ZHENG Xiu-lin[1,2], LIAN Zhi-zhu[1,2], LU Yan-rong[1,2], YUAN-Zheng[1]

(1. Department of Information Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China;
2. School of Telecommunications Engineering, Xidian University, Xi'an 710071, China)

【Abstract】This paper presents an impossible differential cryptanalysis of 13-round CLEFIA-128 no whitening key, which use the 9-round impossible differential. In the process of cryptanalysis, it guesses each key and filter the data pairs using the output and input differences of S-box. It utilizes the keys relations to reduce the number of guessed keys, and introduces the early abort technique to reduce the time complexity. Computing result shows that the complexity of the cryptanalysis is about 2^{120} data and $2^{125.5}$ encryptions

【Key words】block cipher; CLEFIA-128 algorithm; cryptanalysis; impossible differential cryptanalysis; Early Abort

DOI: 10.3969/j.issn.1000-3428.2012.03.048

1 概述

CLEFIA[1]是 Sony 公司提出的一个 128 bit 的分组密码算法，它支持 128 bit、192 bit 和 256 bit 密钥，分别记为 CLEFIA-128、CLEFIA-196 和 CLEFIA-256，相应的迭代轮数为 18 轮、22 轮和 26 轮。设计者声称 CLEFIA 能抗当前所有的已知的攻击方法[1-2]，并且给出了 9 轮 CLEFIA-128 的不可能差分攻击。

不可能差分攻击是利用一个概率为 0 的差分路径实现过滤攻击。目前的研究都是在 9 轮不可能差分的基础上对 CLEFIA 算法进行的不可能差分攻击。其中，文献[3-4]成功攻击了 12 轮 CLEFIA-128/196/256、13 轮 CLEFIA-196/256 和 14 轮的 CLEFIA-256。文献[5]称找到一个 14 轮 CLEFIA-128 的不可能差分攻击，但未能证实该攻击是否成功。

Early Abort 技术[6]的原理是每次通过猜测一小部分密钥(而不是猜测所有密钥)，对候选的数据的部分信息进行加解密，判断是否满足要求，保留的符合要求的数据对，在此基础上，继续猜测剩余的部分密钥，重复上述判断。由于每一次猜测密钥都能过滤掉一部分数据对，从而降低下一步的计算复杂度。

本文在一个 9 轮不可能差分的基础上，给出未使用白化密钥的 13 轮 CLEFIA-128 的不可能差分攻击，攻击中利用了第 1 轮~第 2 轮密钥与第 13 轮密钥间的关系，从而减少密钥猜测量。同时，使用 Early Abort 技术降低计算复杂度。

2 CLEFIA-128 算法

CLEFIA 算法采用具有 4 个分支的广义 Feistel 结构，每个分支 32 bit 数据，且白化密钥作用在开头和结尾轮。本文只给出加密流程，如图 1 所示。

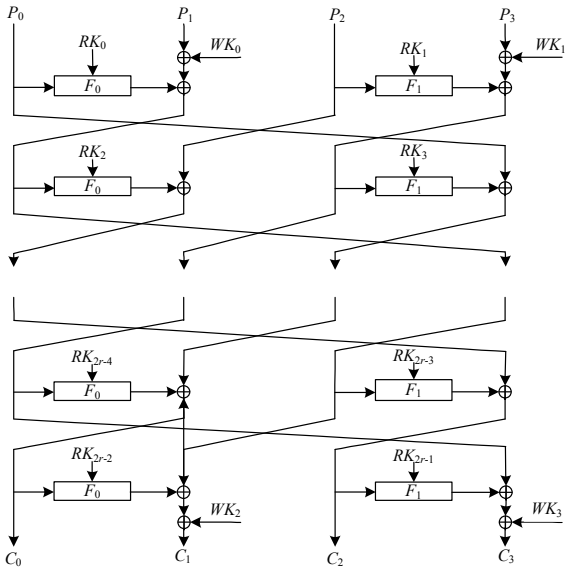

图 1 CLEFIA 算法的加密流程

2.1 符号说明

对算法中符号的说明如下：

$P = P_0 | P_1 | P_2 | P_3$：128 bit 明文 P 由 4 个 32 bit 的字 P_i 连接；

$C = C_0 | C_1 | C_2 | C_3$：128 bit 密文 C 由 4 个 32 比特字 C_i

基金项目：国家自然科学基金资助项目(61070250)；北京市自然科学基金资助项目(4102055)

作者简介：郑秀林(1956－)，男，教授，主研方向：对称密码学；连至助、鲁艳蓉，硕士研究生；袁 征，副教授

收稿日期：2011-08-16 **E-mail：**lzz600@126.com

连接；

C^r, C^{r^*}：第 r 轮输出；

$C_i^r, C_i^{r^*}$：C^r, C^{r^*} 的第 i 个 32 比特字 $(0 \leqslant j < 4)$；

$C_{i,j}^r$：C_i^r 的第 j 个字节 $(0 \leqslant j < 4)$；

ΔC：密文差分；

ΔC_i^r：第 r 轮输出中第 i 个 32 比特字的差分；

$RK_{i,j}$：轮密钥 RK_i 的第 j 个字节；

$X[a-b]$：$\boldsymbol{X}$ 中的第 a bit 到第 b bit；

x^t：向量 x 的转置；

$\boldsymbol{M}_i(\boldsymbol{a})^t$，$\boldsymbol{M}_i^{-1}(\boldsymbol{a})^{\mathrm{T}}$：矩阵 $\boldsymbol{M}_i$ 和它的逆矩阵 $\boldsymbol{M}_i^{-1}$ 对输入向量 $\boldsymbol{a}$ 的转置的运算。

2.2 CLEFIA 算法简介

$WK_0, WK_1, WK_2, WK_3 \in \{0,1\}^{32}$ 是白化密钥，$RK_i \in \{0,1\}^{32}$，$(0 \leqslant i \leqslant 2r)$ 为轮密钥，由密钥扩展算法得到。算法具体如下：

(1) $C_0^0 = P_0, C_1^0 = P_1 \oplus WK_0, C_2^0 = P_2, C_3^0 = P_3 \oplus WK_1$；

(2)对 $i = 1, 2, \cdots, r-1$ 做如下计算：

$C_0^i = C_1^{i-1} \oplus F_0(C_0^{i-1}, RK_{2i-2}), C_1^i = C_2^{i-1}$

$C_2^i = C_3^{i-1} \oplus F_1(C_2^{i-1}, RK_{2i-1}), C_3^i = C_0^{i-1}$

(3)最后一轮 r：

$C_0^r = C_0^{r-1}, C_1^r = C_1^{r-1} \oplus F_0(C_0^{r-1}, RK_{2i-2}) \oplus WK_2$

$C_2^r = C_2^{r-1}, C_3^i = C_3^{r-1} \oplus F_1(C_2^{r-1}, RK_{2i-1}) \oplus WK_3$

F_0 和 F_1 的定义表述如下：

$$F_0\left(C_0^{i-1}, RK_{2i-2}\right) = M_0 \begin{pmatrix} S_0\left(C_{0,0}^{i-1} \oplus RK_{2i-2,0}\right) \\ S_1\left(C_{0,1}^{i-1} \oplus RK_{2i-2,1}\right) \\ S_0\left(C_{0,2}^{i-1} \oplus RK_{2i-2,2}\right) \\ S_1\left(C_{0,3}^{i-1} \oplus RK_{2i-2,3}\right) \end{pmatrix}$$

$$F_1\left(C_2^{i-1}, RK_{2i-1}\right) = M_1 \begin{pmatrix} S_1\left(C_{2,0}^{i-1} \oplus RK_{2i-1,0}\right) \\ S_0\left(C_{2,1}^{i-1} \oplus RK_{2i-1,1}\right) \\ S_1\left(C_{2,2}^{i-1} \oplus RK_{2i-1,2}\right) \\ S_0\left(C_{2,3}^{i-1} \oplus RK_{2i-1,3}\right) \end{pmatrix}$$

其中，S_0 和 S_1 是 2 个非线性 8×8 的 S 盒，$\boldsymbol{M}_0$ 和 $\boldsymbol{M}_1$ 是 4×4 的 Hadamard 字节矩阵。矩阵中的元素属于由不可约多项式 $z^8 + z^4 + z^3 + z^2 + 1$ 确定的有限域 F_{2^8}。

$$\boldsymbol{M}_0 = \begin{pmatrix} 01 & 02 & 04 & 06 \\ 02 & 01 & 06 & 04 \\ 04 & 06 & 01 & 02 \\ 06 & 04 & 02 & 01 \end{pmatrix}, \boldsymbol{M}_1 = \begin{pmatrix} 01 & 08 & 02 & 0a \\ 08 & 01 & 0a & 02 \\ 02 & 0a & 01 & 08 \\ 0a & 02 & 08 & 01 \end{pmatrix}$$

2.3 r 轮 CLEFIA-128 密钥扩展

定义比特置换 $\Sigma: \{0,1\}^{128} \to \{0,1\}^{128}$ 为：

$X_{(128)} \mapsto X[7-63] | X[121-127] | X[0-6] | X[64-120]$

CLEFIA-128 算法的密钥扩展过程如下[1-2]：

(1)生成 128 bit L：用初始 128 bit 密钥 K 做明文，预定 32 bit 常数 $CON_j (0 \leqslant j < 24)$ 做“轮密钥”，进行没有白化密钥的 12 轮 CLEFIA-128 加密，密文为 L。

(2)定义白化密钥：$WK_0 | WK_1 | WK_2 | WK_3 \leftarrow K$。

(3)选取第 2i+1 轮的轮密钥 $RK_{4i} | RK_{4i+1}$ 和(2i+2)轮的轮密钥 $RK_{4i+2} | RK_{4i+3}, 0 \leqslant i \leqslant 9$，$CON_j (24 \leqslant j < 60)$ 为常数。

具体密钥扩展算法如下：

for i = 0,1,⋯,8, do

{T = L⊕(CON$_{24+4i}$ | CON$_{24+4i+1}$ | CON$_{24+4i+2}$ | CON$_{24+4i+3}$)　i 为奇数：

T = T⊕K

L = Σ(L)

RK$_{4i}$ | RK$_{4i+1}$ | RK$_{4i+2}$ | RK$_{4i+3}$ = T}

3 CLEFIA-128 轮密钥关系及不可能差分路径

3.1 轮密钥间关系

文献[4]给出的经密钥扩展得到密钥关系盒定理如下：

$RK_0 | RK_1 | RK_2 | RK_3 \leftarrow L \oplus (CON_{24} | CON_{25} | CON_{26} | CON_{27})$

$RK_{20} | RK_{21} | RK_{22} | RK_{23} \leftarrow$

$\Sigma^5(L) \oplus K \oplus (CON_{44} | CON_{45} | CON_{46} | CON_{47})$

$RK_{24} | RK_{25} | RK_{26} | RK_{27} \leftarrow$

$\Sigma^6(L) \oplus (CON_{48} | CON_{49} | CON_{50} | CON_{51})$

定理[4] 设 C_1、C_2 为常数，则存在以下关系：

$RK_{24} \oplus C_1 = RK_1[10-31] | RK_3[25-31] | RK_3[18-20]$

$RK_{25} \oplus C_2 = RK_3[21-24] | RK_3[11-17] | RK_3[4-10] |$

$RK_2[29-31] | RK_3[0-3] | RK_2[22-28]$

其中，

$C_1 = CON_{48} \oplus (CON_{25}[10-31] | CON_{27}[25-31] | CON_{27}[18-20])$

$C_2 = CON_{49} \oplus (CON_{27}[21-24] | CON_{27}[11-17] | CON_{27}[4-10] |$

$CON_{26}[29-31] | CON_{27}[0-3] | CON_{26}[22-28])$

由上述定理可归纳以下性质：

性质 1 若已知 32 bit 的 RK_{24}，那么可计算出 22 bit 的 $RK_1[10\text{-}31]$，而 $RK_1[0\text{-}9]$ 未知。

性质 2 若已知 32 bit RK_{25}，可计算出 10 bit $RK_2[22\text{-}31]$。

由此可知，密钥 $(RK_{25}, RK_{24}, RK_1, RK_0, RK_{2,3})$ 可由 106 bit 的 L 计算得到。因为 RK_{23} 还与种子密钥 K 有异或关系，本文假定 RK_{23} 和 $RK_{24} | RK_{25} | RK_{26} | RK_{27}$ 相互独立。

3.2 不可能差分路径

文献[5]给出了多条 CLEFIA 的不可能差分路径，其中有如下 3 条路径适用于本文的攻击：

$(0,0,0,(0,0,0,X)) \nrightarrow (0,0,0,(Y,0,0,0))$

$(0,0,0,(0,0,0,X)) \nrightarrow (0,0,0,(0,Y,0,0))$

$(0,0,0,(0,0,0,X)) \nrightarrow (0,0,0,(0,0,Y,0))$

4 13 轮 CLEFIA-128 算法的不可能差分密码分析

将 9 轮不可能差分 $(0,0,0,(0,0,0,X)) \nrightarrow (0,0,0,(a,0,0,0))$ 用于第 3 轮~第 11 轮，本节将阐述没有白化的 13 轮 CLEFIA-128 算法的不可能差分攻击，如图 2 所示，共恢复 L 的 106 bit 的 $(RK_{25}, RK_{24}, RK_2, RK_1, RK_0)$ 和 8 bit 的 $RK_{23,0}$。

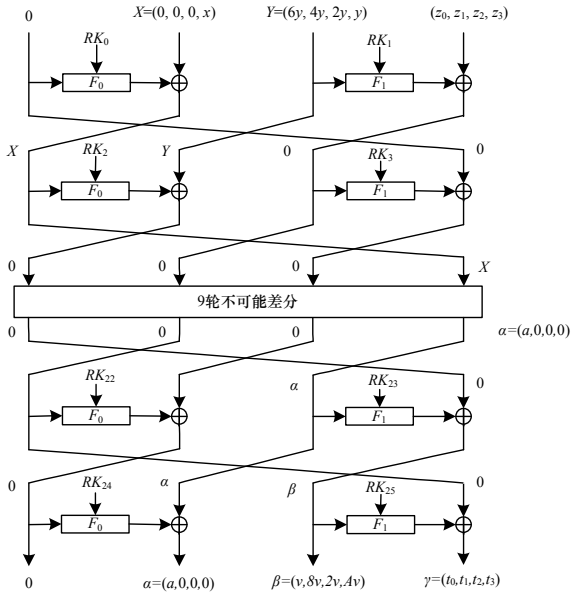

图 2 13 轮 CLEFIA-128 不可能差分攻击

具体攻击步骤如下：

(1)选择明文

令明文 $P=P_0\|P_1\|P_2\|P_3$ 的差分为 $\Delta P_0=(0,0,0,0)$，$\Delta P_1=(0,0,0,x)$，$\Delta P_2=(6y,4y,2y,y)$，$\Delta P_3=(z_0,z_1,z_2,z_3)$，定义如下明文结构：$P_0=(a_0,a_1,a_2,a_3)$，$P_1=(b_0,b_1,b_2,x)$，$P_2=(6y,4y,2y,y)$，$P_3=(z_0,z_1,z_2,z_3)$，其中，$x,\ y,z_0,z_1,z_2,z_3$ 为所有非 0 字节，$a_i(0\leqslant i<3)$ 和 $b_i(0\leqslant i<2)$ 是常数。该明文结构包括 2^{48} 个明文，可以形成 2^{95} 个明文对。本文选取 2^N 个结构，即共有 2^{N+48} 个明文、2^{N+95} 个数据对。

(2)过滤数据

经过 12 轮加密后的明文，选择密文对差分为 $\Delta C=(0,\alpha,\beta,\gamma)$，其中，$\alpha=(a,0,0,0)$，$\beta=M_1(v,0,0,0)^{\mathrm{T}}=(v,\ 8v,\ 2v,\ Av)$，$\gamma=(t_0,t_1,t_2,t_3)$，$a,v,t_0,t_1,t_2,t_3\in\{0,1\}^8$，则过滤概率约为 2^{-80}，即约剩 2^{N+15} 个数据对满足这种形式的明文差分和密文差分。

(3)恢复密钥过程

1)猜测 32 bit RK_{25}

对于剩余的数据对，密文对 $(C_2^{13},\ C_3^{13})$ 和 $(C_2^{13^*},\ C_3^{13^*})$ 的差分为 (β,γ)，计算：

$$\boldsymbol{M}_1^{-1}(\gamma)^{\mathrm{T}}=\boldsymbol{M}_1^{-1}(t_0,t_1,t_2,t_3)^{\mathrm{T}}=(e_0,e_1,e_2,e_3)^{\mathrm{T}}$$

然后执行以下步骤：

①猜测 8 bit 的 $RK_{25,0}$，计算：

$$S_1\left(C_{2,0}^{13}\oplus RK_{25,0}\right)\oplus S_1\left(C_{2,0}^{13^*}\oplus RK_{25,0}\right)=d_0$$

筛选出 $d_0=e_0$ 的数据对。

②猜测 8 bit 的 $RK_{25,1}$，计算：

$$S_0\left(C_{2,1}^{13}\oplus RK_{25,1}\right)\oplus S_0\left(C_{2,1}^{13^*}\oplus RK_{25,1}\right)=d_1$$

筛选出 $d_1=e_1$ 的数据对。

③猜测 8 bit 的 $RK_{25,2}$，计算：

$$S_1\left(C_{2,2}^{13}\oplus RK_{25,2}\right)\oplus S_1\left(C_{2,2}^{13^*}\oplus RK_{25,2}\right)=d_2$$

筛选出 $d_2=e_2$ 的数据对。

④猜测剩余的 8 bit 的 $RK_{25,3}$，计算：

$$S_0\left(C_{2,3}^{13}\oplus RK_{25,3}\right)\oplus S_0\left(C_{2,3}^{13^*}\oplus RK_{25,3}\right)=d_3$$

筛选出 $d_3=e_3$ 的数据对。

通过上述步骤过滤的概率约为 2^{-32}，因此，约剩余 2^{N-17} 个数据对。

2)猜测 32 bit 的 RK_{24}

对剩余的数据对解密，解密密文 (C_0^{13},C_1^{13}) 和 $(C_0^{13^*},C_1^{13^*})$ 得到第 12 轮部分输出 C_2^{12} 和 C_2^{12*}，易知密文差分 $C_2^{12}\oplus C_2^{12^*}=\alpha=(a,0,0,0)$。此步不过滤数据对。

3)猜测 10 bit 的 RK_1

由第 3 节性质 1 可从 RK_{24} 中计算 22 bit，因此，只猜测 $RK_1[0-9]$。数据对即明文对 (C_2^0,C_3^0) 和 $(C_2^{0^*},C_3^{0^*})$，差分为 (y,z)，计算 $M_1^{-1}(z)^t=M_1^{-1}(z_0,z_1,z_2,z_3)^{\mathrm{T}}=(e_0,e_1,e_2,e_3)^{\mathrm{T}}$，对于计算出的 $RK_{1,2}$ 和 $RK_{1,3}$，计算：

$$S_1\left(C_{2,2}^0\oplus RK_{1,2}\right)\oplus S_1\left(C_{2,2}^{0^*}\oplus RK_{1,2}\right)=d_2$$

$$S_0\left(C_{2,3}^0\oplus RK_{1,3}\right)\oplus S_0\left(C_{2,3}^{0^*}\oplus RK_{1,3}\right)=d_3$$

筛选出 $d_2=e_2$，$d_3=e_3$ 的数据对。

①猜测 2 bit 的 $RK_{1,1}$，计算：

$$S_0\left(C_{2,1}^0\oplus RK_{1,1}\right)\oplus S_0\left(C_{2,1}^{0^*}\oplus RK_{1,1}\right)=d_1$$

筛选出 $d_1=e_1$ 的数据对。

②猜测 8 bit 的 $RK_{1,0}$，计算：

$$\mathrm{S}_1\left(\mathrm{C}_{2,0}^0\oplus\mathrm{RK}_{1,0}\right)\oplus\mathrm{S}_1\left(\mathrm{C}_{2,0}^{0^*}\oplus\mathrm{RK}_{1,0}\right)=d_0$$

筛选出 $d_0=e_0$ 的数据对。

通过上述步骤过滤的概率约为 2^{-32}，因此，约剩余 2^{N-49} 个数据对。

4)猜测 32 比特 RK_0

对剩余的数据对作 1 轮加密，加密明文 (C_0^0,C_1^0) 和 $(C_0^{0^*},C_1^{0^*})$，得到第 1 轮部分输出的 C_0^1 和 $C_0^{1^*}$，易知 $C_0^1\oplus C_0^{1^*}=(0,0,0,x)$。此步也不过滤数据对。

5)根据第 3 节性质 2，从 RK_{25} 中计算 8 bit 的 $RK_{2,3}$，计算：

$$S_1\left(C_{0,3}^1\oplus RK_{2,3}\right)\oplus S_1\left(C_{0,3}^{1^*}\oplus RK_{2,3}\right)=d_3$$

其中，$C_{0,3}^1$ 和 $C_{0,3}^{1^*}$ 已由第 4)步得到。筛选满足 $d_3=C_{1,3}^1\oplus C_{1,3}^{1^*}$ 的数据对，$C_1^1=C_0^2=P_2$，$C_1^{1^*}=P_2^*$。

通过此步过滤的概率约为 2^{-8}，因此，约剩余 2^{N-57} 个数据对。

6)猜测 8 bit 的 $RK_{23,0}$

对于剩余的每个数据对，计算并检验是否有：

$$S_1\left(C_{2,0}^{12}\oplus RK_{23,0}\right)\oplus S_1\left(C_{2,0}^{12^*}\oplus RK_{23,0}\right)=C_{3,0}^{12}\oplus C_{3,0}^{12^*}$$

其中，C_2^{12} 已经由第(2)步得到，$C_{3,0}^{12}=C_{4,0}^{13}=C_{4,0}$。如果等式成立，则说明相应的数据对满足 9 轮不可能差分。所建议的密钥猜测值就是错误的，这时删除猜测密钥 RK_{25}, RK_{24}, RK_1, RK_0, $RK_{2,}$, $RK_{23,0}$。

分析完第 6)步的数据对后，约还有 $2^{114}\times\left(1-2^{-8}\right)^{2^{N-57}}$ 个猜测密钥，要完全淘汰错误密钥，则 $2^{114}\times\left(1-2^{-8}\right)^{2^{N-57}}<1$，可设 N=72，即需要 2^{72} 个明文结构，则攻击的数据复杂度为 2^{120} 个选择明文，第(3)步中各步骤的计算复杂度如下：

第 1)步：$2\times\frac{1}{8}\times\frac{1}{13}\times\sum_{i=0}^{3}\left(2^{N+15-8i}\times2^{8(i+1)}\right)<2^{91.5}$

第 2)步：$2\times\frac{1}{2}\times\frac{1}{13}\times2^{N-17}\times2^{32}\times2^{32}<2^{115.5}$

第 3)步：

$$2\times\frac{1}{8}\times\frac{1}{13}\times\left(\sum_{i=0}^{1}\left(2^{N-17-8i}\times2^{64}\right)+\sum_{i=0}^{1}\left(2^{N-33-8i}\times2^{64}\times2^{2+8i}\right)\right)<2^{1125}$$

第 4)步：$2\times\frac{1}{2}\times\frac{1}{13}\times2^{N-49}\times2^{64+10}\times2^{32}<2^{125.5}$

第 5)步：$2\times\frac{1}{8}\times\frac{1}{13}\times2^{N-49}\times2^{106}<2^{123.5}$

第 6)步：

$$2\times\frac{1}{8}\times\frac{1}{13}\times2^{114}\times\left\{1+\left(1-2^{-8}\right)+\left(1-2^{-8}\right)^2+\cdots+\left(1-2^{-8}\right)^{2^{N-57}-1}\right\}<2^{124}$$

在第(2)步中，为了得到满足特定差分要求的数据对，本文首先将每一个明文和其密文存储到 Hash 表中，索引值为密文的 10 个字节数，具体值为：C_0^{13} 的 4 个字节，C_1^{13} 的前 3 个字节 $(C_{1,0}^{13},C_{1,1}^{13},C_{1,2}^{13})$ 和 $M_1^{-1}(C_2^{13})$ 的前 3 个字节。其中，计算 $M_1^{-1}(C_2^{13})$ 约需要 $2^{N+48}\times\frac{1}{2}\times\frac{1}{13}\approx2^{115.5}$ 次加密，则第(1)步和第(2)步需要 $2^{120}+2^{115.5}\approx2^{120}$ 次加密。

通过观察可知，如果改变第 2)步和第 3)步中密钥猜测顺

序，即先猜测 32 bit 的 RK_1，再猜测 10 bit 的 RK_{24}，此时这两步的计算复杂度分别为：

$$2\times\frac{1}{8}\times\frac{1}{13}\times\sum_{i=0}^{3}\left(2^{N-17-8i}\times 2^{32}\times 2^{8(i+1)}\right)<2^{91.5}$$

$$2\times\frac{1}{2}\times\frac{1}{13}\times 2^{N-49}\times 2^{64}\times 2^{10}<2^{93.5}$$

虽然两者都小于之前的复杂度，但并没能影响最终的结果。不过在分析中还是应充分考虑到密钥猜测顺序对计算复杂度的影响，把能过滤数据对的猜测密钥步骤尽可能提前，以充分利用 Early Abort 技术。

综上所述，攻击没有白化的 13 轮 CLEFIA-128 需要的数据复杂度为 2^{120}，时间复杂度为 $2^{125.5}$ 次 13 轮加密。

目前对 CLEFIA-128 算法的不可能差分密码分析最多到达 12 轮，各文献的分析的结果比较如表 1 所示。

表 1 CLEFIA-128 算法不可能差分密码分析的结果比较

各类分析	攻击轮数	恢复密钥长度/bit	攻击选择明文量	时间复杂度
文献[1-2]分析	10	32	$2^{101.7}$	2^{120}
文献[3]分析	12	96	$2^{119.1}$	$2^{98.1}$
文献[5]分析	12	80	$2^{118.9}$	2^{82}
文献[6]分析	12	88	2^{111}	2^{90}
本文分析	13	114	2^{120}	$2^{125.5}$

5 结束语

本文利用 9 轮不可能差分和轮密钥的关系，分析未使用白化密钥的 13 轮 CLEFIA-128 算法。在文献[1-6]中，使用的方法都是通过轮函数 S 盒差分分布表来计算出密钥，并把它作为错误密钥。本文则是通过猜测轮密钥，筛选出满足轮函数中 S 盒输入输出差分对的数据对，对于满足条件的数据对所猜测的密钥为错误密钥。本文分析考虑了轮密钥间的相互的关系降低了猜测密钥量，并使用了 Early Abort 技术，降低了计算复杂度。但笔者同时发现，近几年对于 Camellia 算法的不可能差分密码分析中大多采用类似本文的方法。下一步将对具体 2 种方法的优劣进行分析。

参考文献

[1] Sony Corporation. The 128-bit Blockcipher CLEFIA, Security and Performance Evaluations, Revision 1.0[EB/OL]. (2007-06-01). http://www.sony.net/Products/cryptography/clefia/.

[2] Shirai T, Shibutani K, Akishita T, et al. The 128-bit Blockcipher CLEFIA(Extended Abstract)[C]//Proc. of FSE'07. Dubrovnik, Croatia: [s. n.], 2007: 181-195.

[3] Wang Wei, Wang Xiaoyun. Improved Impossible Differential Cryptanalysis of CLEFIA[EB/OL]. (2008-05-05). http://eprint.iacr.org/2007/466.

[4] Tsunoo Y, Tsujihara E, Shigeri M, et al. Impossible Differential Cryptanalysis of CLEFIA[C]//Proc. of FSE'08. Atlanta, USA: [s. n.], 2008: 398-411.

[5] Zhang Wenying, Han Jing. Impossible Differential Analysis of Reduced Round CLEFIA[C]//Proc. of Inscrypt'08. Beijing, China: [s. n.]: 181-191.

[6] Wu Wenling, Zhang Lei, Zhang Wentao. Improved Impossible Differential Cryptanalysis of Reduced-round Camellia[C]//Proc. of SAC'08. [S. l.]: ACM Press, 2008: 442-456.

编辑 金胡考

(上接第 140 页)

圆曲线密码系统具有更高的安全性。本文提出的新方案基于椭圆曲线密码算法，椭圆曲线密码体制确保了其安全性。而文献[8]所提的盲代理盲签名方案基于大素数分解和离散对数，因此，本文方案具有更高的安全性。

在基于椭圆曲线密码体制的各种签名算法中，影响计算速度的关键是点加运算、倍点运算及模逆运算，假设 M 表示一次椭圆曲线上倍点运算，I 表示一次求模逆运算，S 表示一次椭圆曲线上加法运算。将本文提出的方案与文献[9]提出的椭圆曲线上的盲代理盲签名方案进行效率比较。代理授权阶段，2 种方案计算步骤一致，在此仅比较代理签名阶段和验证阶段的计算量，如表 1 所示。

表 1 方案效率比较

方案	签名阶段	验证阶段	总计算量
文献[9]方案	5M+ I	5M+4S+I	10M+4S+2I
本文方案	3M+S+ 2I	4M+3S	7M+4S+2I

由表 1 可以看出，本文方案与文献[9]方案相比，减少了 3 次倍点运算，计算时间复杂度更低。本文方案不需要可信的第三方参与，因而更容易实现。

5 结束语

本文结合盲代理签名和代理盲签名的优点，设计了基于椭圆曲线不需可信方的盲代理盲签名方案，并对其安全性进行了分析，该方案既防止了签名权限的滥用，又保护了代理签名者和用户的隐私。与文献[8-9]方案相比，本文方案具有更快的运算速度和更高的安全性，且不需要可信第三方的参与，在电子商务等方面具有较高的实用价值。

参考文献

[1] Mambo M, Usuda K, Okamoto E. Proxy Signature: Delegation of the Power to Sign Messages[J]. IEICE Transactions on Fundamentals, 1996, E79-A(9): 1338-1353.

[2] Lin W D, Jan J K. A Security Personal Learning Tools Using a Proxy Blind Signature Scheme[C]//Proceedings of ICCLC'00. Washington D. C., USA: IEEE Computer Society, 2000: 273-277.

[3] Lee B, Kim H, Kim K. Strong Proxy Signature and Its Applications[C]//Proceedings of ICS'00. Tainan, China: [s. n.], 2000: 54-59.

[4] Shum K, Victor K. A Strong Proxy Signature Scheme with Proxy Signer Privacy Protection[C]//Proceedings of WETICE'10. Larissa, Greece: [s. n.], 2002: 55-56.

[5] 谷利泽，李献中，杨义先. 不需要可信任方的匿名代理签名方案[J]. 北京邮电大学学报, 2005, 28(1): 48-50.

[6] 赵泽茂. 基于椭圆曲线的代理盲签名方案[J]. 河海大学学报, 2006, 34(3): 329-332.

[7] 靳 虹, 王相海. 基于椭圆曲线的不需要可信方的匿名代理签名方案[J]. 计算机科学, 2009, 36(11): 120-122.

[8] 李 娆, 何明星. 可收回代理权的公平盲代理盲签名方案[J]. 计算机工程, 2006, 32(13): 156-158.

[9] 张建中, 马伟芳. 椭圆曲线上的盲代理盲签名方案[J]. 计算机工程, 2010, 36(11): 126-127, 130.

编辑 金胡考