LBlock-s 算法的不可能差分分析

贾 平[1,2],徐 洪[1,2],来学嘉[3]
(1. 信息工程大学,河南郑州 450001;2. 数学工程与先进计算国家重点实验室,河南郑州 450001;
3. 上海交通大学计算机科学与工程系,上海 200240)

摘 要: LBlock-s 算法是 CAESAR 竞赛候选认证加密算法 LAC 中的主体算法,算法结构与 LBlock 算法基本一致,只是密钥扩展算法采用了扩散效果更好的增强版设计. 利用新密钥扩展算法中仍然存在的子密钥间的迭代关系,通过选择合适的 14 轮不可能差分特征,我们给出了对 21 轮 LBlock-s 算法的不可能差分分析. 攻击需要猜测的子密钥比特数为 72 比特,需要的数据量为 2^{63} 个选择明文,时间复杂度约为 $2^{67.61}$ 次 21 轮加密. 利用部分匹配技术,我们也给出了直到 23 轮 LBlock-s 算法低于密钥穷举量的不可能差分分析结果. 这些研究可以为 LAC 算法的整体分析提供参考依据.

关键词: LBlock 算法; LBlock-s 算法; 密钥扩展算法; 不可能差分分析
中图分类号: TP 309.7; TN 918.1 **文献标识码:** A **文章编号:** 0372-2112 (2017) 04-0966-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.04.028

Impossible Differential Cryptanalysis of Reduced-Round LBlock-s

JIA Ping[1,2],XU Hong[1,2],LAI Xue-jia[3]
(1. *Information Engineering University*, *Zhengzhou*, *Henan* 450001, *China*;
2. *State Key Laboratory of Mathematical Engineering and Advanced Computing*, *Zhengzhou*, *Henan* 450001, *China*;
3. *Department of Computer Science & Engineering*, *Shanghai Jiao Tong University*, *Shanghai* 200240, *China*)

Abstract: LBlock-s is the kernel block cipher of the authentication encryption algorithm LAC submitted to CAESAR competition. The general structure of LBlock-s is almost the same as that of LBlock, but LBlock-s adopts an improved key schedule algorithm with better diffusion property. Using the shifting relation of subkeys derived by the key schedule algorithm, an impossible differential cryptanalysis on 21-round LBlock-s was presented based on a 14-round impossible differential. The time and data complexities are $2^{67.61}$ 21-round encryptions and 2^{63} chosen plaintexts respectively, and the number of subkey bits needed to be guessed is 72. Using partial-matching method, an impossible differential cryptanalysis on LBlock-s up to 23-round was also presented with time complexity less than exhaustion of all key bits. This work is useful for the security analysis of LAC algorithm.

Key words: LBlock; LBlock-s; key schedule algorithm; impossible differential cryptanalysis

1 引言

LBlock-s 算法是 CAESAR 竞赛[1] 候选认证加密算法 LAC[2] 中的主体算法. 算法整体与 LBlock 算法[3] 基本一致,采用 Feistel-SP 结构,基于 4 比特块设计,分组长度为 64 比特,密钥长度为 80 比特,迭代轮数为 32 轮,但是密钥扩展算法采用了 Wang Yanfeng 等针对 Biclique 攻击提出的扩散效果更好的增强版设计[4].

不可能差分分析[5,6] 是分组密码中非常有效的密码分析方法之一,它利用出现概率为 0 的差分特征排除错误密钥以达到降低密钥搜索量的目的. 2003 年,Kim 等[7] 给出了自动搜索不可能差分链的矩阵方法. 利用该方法,Wu[8] 等给出了 LBlock 算法形如 $(0,\alpha) \not\to_{14r} (\beta, 0)$ 的 14 轮不可能差分特征(其中 α,β 恰有一个非零块),并给出了对 20 轮 LBlock 算法的不可能差分分析. 利用密钥扩展算法的特点,选择新的不可能差分特征,

收稿日期: 2015-10-23; 修回日期: 2016-03-01; 责任编辑: 孙瑶
基金项目: 国家自然科学基金(No. 61100200, No. 61170235, No. 61309017, No. 61472251, No. 61502524, No. 61521003, No. U1536101); 国家 863 高技术研究发展计划(No. 2015AA01A708)

Liu、Karakoc 等[8,9] 随后分别给出了 LBlock 算法的 21 轮、22 轮不可能差分分析. 2014 年亚密会上, Boura 等[10] 研究了利用密钥扩展算法的特点优化不可能差分分析的一般模型,并给出了直到 23 轮 LBlock 算法的不可能差分分析. Minier 等[11~16] 研究了相关密钥条件下减轮 LBlock 算法的差分和不可能差分分析.

由于 LBlock-s 算法与 LBlock 算法整体结构一致,它也具有形如 $(0,\alpha) \not\to_{14r} (\beta,0)$ 的 14 轮不可能差分特征. 然而由于 LBlock-s 算法采用了新的密钥扩展算法,子密钥扩散速度更快,原有的基于密钥扩展算法的不可能差分分析结论不再成立. 通过对新密钥扩展算法的深入分析我们发现,采用新的密钥扩展算法后相邻轮部分子密钥间仍然存在一些可以利用的迭代关系,由此可以对 21 轮 LBlock-s 算法进行不可能差分分析,攻击需要猜测的子密钥比特数为 72 比特,需要的数据量为 2^{63} 个选择明文,时间复杂度约为 $2^{67.61}$ 次 21 轮加密. 再利用 Boura 等的改进方案可以给出直到 23 轮 LBlock-s 算法的不可能差分分析结论.

2　LBlock-s 算法简介

先给出文中要用到的一些符号.

P, C: 64 比特明文和密文;

ΔP, ΔC: 明文和密文差分;

K_r: 第 r 轮的子密钥;

X_r: 第 r 轮左边 32 比特输入, X_0 为第 1 轮右边 32 比特输入;

$X \| Y$: X 和 Y 的级联;

X_i^j: X_i 的第 j 个 4 比特块($0 \leqslant j \leqslant 7$,从左到右依次为 7,6,…,0);

$X <<< i$: X 循环左移 i 个比特;

$[i]_2$: 整数 i 的二进制形式.

下面简要介绍 LBlock-s 算法及其密钥扩展算法.

LBlock-s 是 CAESAR 竞赛候选认证加密算法 LAC 中的主体算法,其结构与 LBlock 算法基本一致,均采用 Feistel-SP 结构,分组长度为 64 比特,密钥长度为 80 比特,迭代轮数为 32 轮. 以 $P = X_1 \| X_0$ 表示 64 比特明文,具体加密过程如下:

(1) 对 $i = 1,2,\cdots,32$,令

$$X_i = F(X_{i-1}, K_{i-1}) \oplus (X_{i-2} <<< 8).$$

(2) 交换左右两块,输出 $C = X_{33} \| X_{32}$,作为 64 比特密文.

其中轮函数 F 定义为: $F(X_i, K_i) = P(S(X_i \oplus K_i))$ (参见图 1),LBlock-s 算法使用了相同的 4 比特 S 盒,而 LBlock 算法使用了 8 个不同的 4 比特 S 盒,具体 S 盒的构造参见文献[2,3].

解密算法是加密算法的逆过程,已知密文 $C = X_{33} \| X_{32}$,输出 64 比特明文 $P = X_1 \| X_0$.

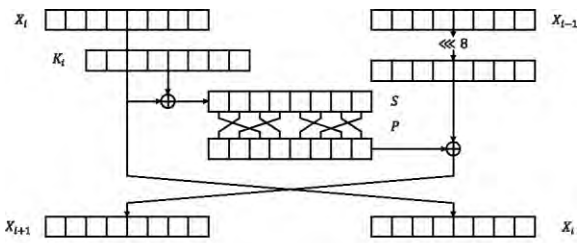


图1　LBlock-s算法中的轮函数

LBlock-s 的密钥扩展算法如下:

将 80 比特密钥 $K = (k_{79}k_{78}\cdots k_1k_0)$ 置入寄存器,次序为 k_{79} 在最左边, k_0 在最右边. 取最左边 32 比特作为子密钥 K_1. 对 $i = 1,2,\cdots,31$,按照以下步骤执行 31 次:

(1) $K <<< 24$

(2) $[k_{55}k_{54}k_{53}k_{52}] = S[k_{79}k_{78}k_{77}k_{76}] \oplus [k_{55}k_{54}k_{53}k_{52}]$

$[k_{31}k_{30}k_{29}k_{28}] = S[k_{75}k_{74}k_{73}k_{72}] \oplus [k_{31}k_{30}k_{29}k_{28}]$

$[k_{67}k_{66}k_{65}k_{64}] = [k_{71}k_{70}k_{69}k_{68}] \oplus [k_{67}k_{66}k_{65}k_{64}]$

$[k_{51}k_{50}k_{49}k_{48}] = [k_{11}k_{10}k_9k_8] \oplus [k_{51}k_{50}k_{49}k_{48}]$

(3) $[k_{54}k_{53}k_{52}k_{51}k_{50}] = [k_{54}k_{53}k_{52}k_{51}k_{50}] \oplus [i]_2$

(4) 取寄存器最左端 32 比特密钥作为子密钥 K_{i+1}.

对比 LBlock 原始密钥扩展算法(如算法 1 所示)可以发现,原密钥扩展算法中寄存器每次更新 13 个比特,而新密钥扩展算法中寄存器每次更新 16 个比特,并且更新时与更多的密钥比特相关使得密钥比特的整体扩散效果更好,这也是 Wang 等[4] 在分析 LBlock 算法抵抗 Biclique 攻击能力时提出的改进方案.

算法 1　LBlock 的密钥扩展算法

将 80 比特密钥 $K = (k_{79}k_{78}\cdots k_1k_0)$ 置入寄存器,次序为 k_{79} 在最左边, k_0 在最右边. 取最左边 32 比特作为子密钥 K_1. 对 $i = 1,2,\cdots,31$,按照以下步骤执行 31 次:

(1) $K <<< 29$

(2) $[k_{79}k_{78}k_{77}k_{76}] = S_8[k_{79}k_{78}k_{77}k_{76}]$

$[k_{75}k_{74}k_{73}k_{72}] = S_9[k_{75}k_{74}k_{73}k_{72}]$

(3) $[k_{50}k_{49}k_{48}k_{47}k_{46}] = [k_{50}k_{49}k_{48}k_{47}k_{46}] \oplus [i]_2$

(4) 取寄存器最左端 32 比特密钥作为 K_{i+1}.

由于 LBlock-s 算法的轮函数以 4 比特块为基本单位,其密钥扩展算法中采用的移位数 24 也是 4 的倍数,不会打乱块内的顺序. 另一方面,密钥寄存器的更新也基于多个 4 比特块的整体运算,因此若将密钥寄存器以 4 比特块为基本单位进行重组,则可以大大简化密钥寄存器各块间的状态更新关系.

不妨记密钥寄存器的初值为 $\kappa_1 = (\kappa_1^{19}, \kappa_1^{18}, \cdots, \kappa_1^0) = K$,其中 $\kappa_1^j = (k_{4j+3}k_{4j+2}k_{4j+1}k_{4j})$ $(0 \leqslant j \leqslant 19)$ 为 4 比特数,并记 i 轮迭代后密钥寄存器的状态值为 $\kappa_{i+1} =$

$(\kappa_{i+1}^{19},\kappa_{i+1}^{18},\cdots\cdots,\kappa_{i+1}^{0})$，其中 $0\leqslant i\leqslant 31$. 再设轮常数为 $i=(i_4i_3i_2i_1i_0)$，并记 $i_H=0\,|\,i_4\,|\,i_3\,|\,i_2$，$i_L=i_1\,|\,i_0\,|\,00$，则密钥寄存器的状态更新满足关系式:

$$\kappa_{i+1}^{16}=\kappa_i^{10}\oplus\kappa_i^{11};$$
$$\kappa_{i+1}^{13}=S(\kappa_i^{13})\oplus\kappa_i^{7}\oplus i_H;$$
$$\kappa_{i+1}^{12}=\kappa_i^{6}\oplus\kappa_i^{16}\oplus i_L;$$
$$\kappa_{i+1}^{7}=S(\kappa_i^{12})\oplus\kappa_i^{1};$$
$$\kappa_{i+1}^{j}=\kappa_i^{(j-6)\bmod 20},\ j\neq 16,13,12,7.$$

由于每次选择密钥寄存器最左边的 32 比特作为子密钥，故第 i 个子密钥可以表示为

$$K_i=(\kappa_i^{19},\kappa_i^{18},\cdots,\kappa_i^{12}),$$

其中 $K_i^j=\kappa_i^{j+12}(j=7,6,\cdots,0)$.

利用寄存器状态块间的移位关系 $\kappa_{i+1}^{j}=\kappa_i^{(j-6)\bmod 20}$ 和子密钥的表达式 $K_i^j=\kappa_i^{j+12}$，我们容易得到相邻几拍部分子密钥间满足下面的推移关系:

$$K_i^0=\kappa_i^{12}=\kappa_{i+1}^{18}=K_{i+1}^{6};$$
$$K_i^1=\kappa_i^{13}=\kappa_{i+1}^{19}=K_{i+1}^{7};$$
$$K_i^4=\kappa_i^{16}=\kappa_{i+1}^{2}=\kappa_{i+2}^{8}=\kappa_{i+3}^{14}=K_{i+3}^{2};$$
$$K_i^5=\kappa_i^{17}=\kappa_{i+1}^{3}=\kappa_{i+2}^{9}=\kappa_{i+3}^{15}=K_{i+3}^{3};$$
$$K_i^7=\kappa_i^{19}=\kappa_{i+1}^{5}=\kappa_{i+2}^{11}=\kappa_{i+3}^{17}=K_{i+3}^{5}.$$

利用这些依赖关系可以大大降低下文不可能差分分析密钥恢复过程中的计算量.

3 LBlock-s 的不可能差分分析

本节介绍 21 轮 LBlock-s 算法不可能差分分析的结果，并分析对更高轮 LBlock-s 算法进行不可能差分分析的可能性和复杂度.

3.1 不可能差分分析原理

不可能差分分析利用出现概率为 0 的差分特征排除错误密钥. 假设已经找到一条 r 轮不可能差分特征 $\Delta\alpha\not\rightarrow\Delta\beta$，不可能差分分析时通常在此差分特征前后分别加上若干轮，不妨将加密过程分成三部分: $E=E_2\circ E_1\circ E_0$. 其中 E_0 为不可能差分前加的若干轮加密过程，E_2 为不可能差分后加的若干轮加密过程，E_1 为不可能差分特征对应的加密过程. 具体攻击过程如下:

选择一组明文对 (P,P') 和对应的密文对 (C,C')，猜测它们分别经过 E_0 加密和 E_2^{-1} 解密时用到的密钥，并计算 E_1 的输入输出差分，若该输入输出差分与不可能差分特征 $\Delta\alpha\not\rightarrow\Delta\beta$ 匹配，则从候选密钥集中排除此猜测密钥. 选择新的明密文对，重复上述过程，直到候选密钥集中只剩下唯一的候选密钥为止.

若 E_0，E_2 加解密过程中用到的密钥是独立的，还可以分别猜测 E_0，E_2 加解密过程中用到的密钥以进一步降低复杂度. 例如，选择好明文对 (P,P') 和对应的密文对 (C,C') 后，可以先猜测明文对 (P,P') 经过 E_0 加密时用到的密钥，并计算 E_0 的输出差分，若该输出差分等于 $\Delta\alpha$，则将 E_0 加密过程中猜测的密钥记入表 A 中. 类似的，再猜测密文对 (C,C') 通过 E_2^{-1} 解密时用到的密钥，并计算 E_2^{-1} 的输出差分，若该输出差分等于 $\Delta\beta$，则将 E_2 解密过程中猜测的密钥记入表 B 中. 最后从候选密钥集中排除 $A\times B$ 中的密钥即可.

不妨设 E_0，E_2 加解密过程中猜测的密钥比特数分别为 k，l，再设利用一个明文对筛选后随机候选密钥被保留的概率为 $1-2^{-c}$，则用 N 个明文对筛选后随机候选密钥被保留的概率为 $P=(1-2^{-c})^N\approx e^{-N/2^c}$，攻击需要的明文对数 $N\geqslant 2^c$，攻击的时间复杂度约为 $T=\max(2^k\times N,2^l\times N)$，采用部分匹配技术可以进一步降低实际攻击的计算复杂度.

此外，实际攻击时选择概率 P 的不同阈值，可以在数据量和计算复杂度间进行折中平衡，例如，若要求 $P=(1-2^{-c})^N<2^{-(k+l)}$，即所有错误密钥都可以被排除，则明文对数 N 满足

$$(1-2^{-c})^N\times 2^{k+l}\leqslant 1\Rightarrow N\geqslant(k+l)\times\ln 2\times 2^c$$

若要求 $P=(1-2^{-c})^N\leqslant 2^{-1}$，即至少排除一半的错误密钥，则明文对数 N 满足 $N\geqslant\ln 2\times 2^c$.

3.2 21 轮 LBlock-s 算法的不可能差分分析

利用 Kim 等自动搜索不可能差分链的矩阵方法，Wu 等[3] 给出了 LBlock 算法的形如 $(0,\alpha)\not\rightarrow_{14r}(\beta,0)$ 的 14 轮不可能差分特征，其中 α，β 恰有一个非零块. 该差分特征只依赖于算法的整体结构，而与 S 盒的具体构造无关，因此 LBlock-s 算法也存在上述形式的 14 轮不可能差分特征. 下面考虑对 LBlock-s 算法的不可能差分分析.

为了充分利用子密钥之间的关系减少计算复杂度，我们选择的 14 轮不可能差分特征为

$$(00000000,00\alpha_500000)\not\rightarrow_{14r}(000\beta_40000,00000000)$$

基于此在前面添加 4 轮在后面添加 3 轮，我们给出了对 21 轮 LBlock-s 算法的不可能差分分析(参见图 2). 攻击中需要猜测的子密钥比特总数为 72 比特，其中后 3 轮需要猜测 28 比特，前 4 轮需要猜测 $56-12=44$ 比特. 攻击中用到的子密钥间的依赖关系为:

$$K_2^6=K_1^0;\ K_3^7=K_2^1;\ K_4^5=K_1^7.$$

图 2 列出了该 14 轮不可能差分特征向前后扩展时中间变量 X_i 差分值的扩散情况(其中绿色部分标注的是非零差分的位置)以及部分加密和解密过程中用到的子密钥(彩色加数字标注的是该子密钥在第几轮变换中被用到). 具体攻击过程如下:

(1) 选取 N 个明文对 (P,P') 及对应的密文对 (C,C')，使其满足:

$$\Delta P=(\Delta X_1,\Delta X_0)=(**00000*,*0*0*0**);$$
$$\Delta C=(\Delta X_{22},\Delta X_{21})=(*0*0000*,**000000).$$

(2) 猜测使$(\Delta X_5,\Delta X_4)=(00000000,00\alpha_500000)$的子密钥:

(a) 猜测 12 比特子密钥 K_1^0,K_1^6,K_1^7,使$(\Delta X_2,\Delta X_1)=(0000*\ 0*\ 0,**00000*\ )$,即要求 $\Delta S(K_1^0\oplus X_1^0)=\Delta X_0^0,\Delta S(K_1^6\oplus X_1^6)=\Delta X_0^5,\Delta S(K_1^7\oplus X_1^7)=\Delta X_0^3$;

(b) 猜测 16 比特子密钥 K_2^1,K_2^3,K_2^2,K_2^3,使$(\Delta X_3,\Delta X_2)=(00000*\ 00,0000*\ 0*\ 0)$,即要求 $\Delta S(K_2^3\oplus X_2^3)=\Delta X_1^7,\Delta S(K_2^1\oplus X_2^1)=\Delta X_1^6$,其中 $S(K_1^2\oplus X_1^2)\oplus X_0^1=X_2^3,S(K_1^3\oplus X_1^3)\oplus X_0^7=X_2^1$;

(c) 猜测 12 比特子密钥 K_3^2,K_2^0,K_2^1,使$(\Delta X_4,\Delta X_3)=(00*\ 00000,00000*\ 00)$,即要求 $\Delta S(K_3^2\oplus X_3^2)=\Delta X_2^1$,其中 $S(K_1^1\oplus X_1^1)\oplus X_0^6=X_2^0,S(K_2^0\oplus X_2^0)\oplus X_1^0=X_3^2$;

(d) 猜测 4 比特子密钥 K_1^4,使$(\Delta X_5,\Delta X_4)=(00000000,00\alpha_500000)$,其中 $K_4^5=K_1^7,K_3^7=K_2^1,K_2^6=K_1^0$,即要求 $\Delta S(K_4^5\oplus X_4^5)=\Delta X_3^2$,其中 $S(K_1^4\oplus X_1^4)\oplus X_0^4=X_2^6,S(K_2^6\oplus X_2^6)\oplus X_1^5=X_3^7,S(K_3^7\oplus X_3^7)\oplus X_2^3=X_4^5$;

(e) 将满足条件的子密钥存储到表 A 中.

(3) 猜测使$(\Delta X_{19},\Delta X_{18})=(000\beta_40000,00000000)$的子密钥:

(a) 猜测 8 比特子密钥 K_{21}^6,K_{21}^7,使$(\Delta X_{21},\Delta X_{20})=(**000000,0*\ 000000)$,即要求 $\Delta S(K_{21}^6\oplus X_{21}^6)=\Delta X_{22}^7,\Delta S(K_{21}^7\oplus X_{21}^7)=\Delta X_{22}^5$;

(b) 猜测 8 比特子密钥 K_{21}^1,K_{20}^6,使$(\Delta X_{20},\Delta X_{19})=(0*\ 000000,000*\ 0000)$,即要求 $\Delta S(K_{20}^6\oplus X_{20}^6)=\Delta X_{21}^7$,其中 $S(K_{21}^1\oplus X_{21}^1)\oplus X_{22}^0=X_{20}^6$;

(c) 猜测 12 比特子密钥 $K_{21}^4,K_{20}^4,K_{19}^4$,使$(\Delta X_{19},\Delta X_{18})=(000\beta_40000,00000000)$,即要求 $\Delta S(K_{19}^4\oplus X_{19}^4)=\Delta X_{20}^6$,其中 $S(K_{21}^4\oplus X_{21}^4)\oplus X_{22}^6=X_{20}^4,S(K_{20}^4\oplus X_{20}^4)\oplus X_{21}^6=X_{19}^4$;

(d) 将满足条件的子密钥存储到表 B 中.

(4) 在候选密钥中去除中 $A\times B$ 有关的子密钥.

在上述攻击过程中,需要用到 84 比特子密钥,然而结合前面提到的子密钥之间的关系,只需猜测 $84-3\times 4=72$ 比特. 用于筛选数据的密钥共 $11\times 4=44$ 比特. 为便于比较,试验中我们仍然选取明密文对数 $N=2^{50}$,筛选结束后被保留下来的错误密钥数约为$(1-2^{-44})^{2^{50}}\times 2^{72}=2^{-20}$,故此攻击方法可以很高的概率恢复出正确密钥. 攻击的主要步骤集中在第(2)步和第(3)步,其具体复杂度分析参见表 1.

下面结合图 2 对表 1 中的数据做简要说明. 攻击的主要步骤为第(2)步和第(3)步,其中第(2)步需要猜

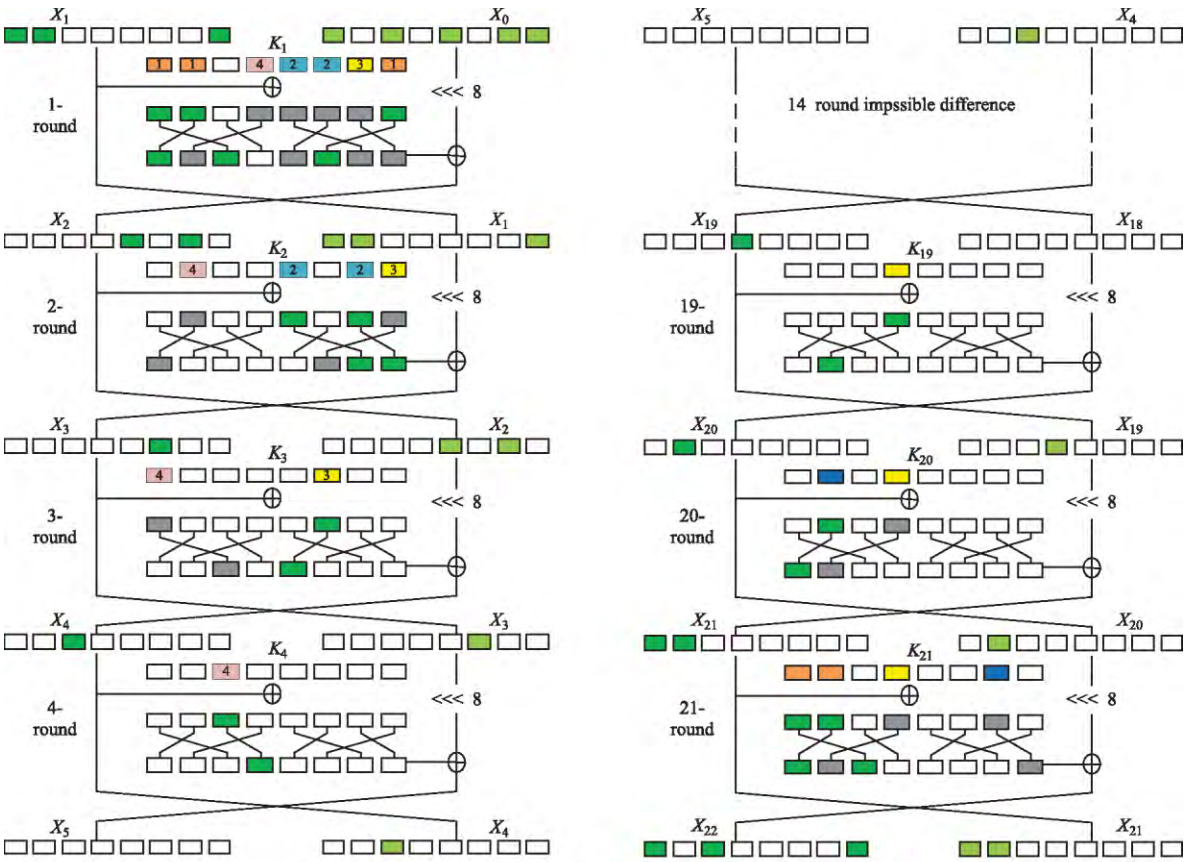


图2　21轮LBlock-s的不可能差分分析 (前面加4轮，后面加3轮)

测前 4 轮用到的部分子密钥，保留那些使得第 4 轮的输出差分等于 $(00000000, 00\alpha_500000)$ 的子密钥，并将结果保存在表 A 中. 类似的，第(3)步需要猜测后 3 轮用到的部分子密钥，保留那些使得第 19 轮的输入差分等于 $(000\beta_40000, 00000000)$ 的子密钥，并将结果保存表 B 中. 由于我们选用的是不可能差分特征，故 $A \times B$ 中的子密钥都为错误密钥，应该从候选密钥中排除. 通过上述筛除方法对多组不同选择明文对进行处理可以尽可能多地排除错误密钥.

表 1 21 轮 LBlock-s 不可能差分攻击复杂度分析

步骤	猜测的子密钥	猜测子密钥数/比特	匹配数/比特	过滤后数据量/对	计算量/1 轮加密
(2).(a)	K_1^0, K_1^6, K_1^7	12	12	$2^{50-12}=2^{38}$	$2^{50}\times 2^{12}\times 2=2^{63}$
(2).(b)	$K_1^2, K_2^3; K_1^3, K_2^1$	16	8	$2^{38-8}=2^{30}$	$2^{38}\times 2^{28}\times 2=2^{67}$
(2).(c)	K_1^1, K_2^0, K_3^2	12	4	$2^{30-4}=2^{26}$	$2^{30}\times 2^{40}\times 2=2^{71}$
(2).(d)	$K_1^4, K_2^6(K_1^0), K_3^7(K_2^1), K_4^5(K_1^7)$	4	4	$2^{26-4}=2^{22}$	$2^{26}\times 2^{44}\times 2=2^{71}$
(3).(a)	K_{21}^6, K_{21}^7	8	8	$2^{50-8}=2^{42}$	$2^{50}\times 2^{8}\times 2=2^{59}$
(3).(b)	K_{21}^1, K_{20}^6	8	4	$2^{42-4}=2^{38}$	$2^{42}\times 2^{16}\times 2=2^{59}$
(3).(c)	$K_{21}^4, K_{20}^4, K_{19}^4$	12	4	$2^{38-4}=2^{34}$	$2^{38}\times 2^{28}\times 2=2^{67}$

我们以第(2)步为例对其复杂度进行分析. 已知的选择明文对具有差分 $\Delta P=(\Delta X_1, \Delta X_0)=(**00000*, *0*0*0**)$，由图 2 中前 4 轮的差分扩散过程知，第 1 轮加密后的输出差分应该满足 $\Delta X_2=0000*0*0$，因此需要猜测 X_1 非零差分位置对应的子密钥 K_1^0, K_1^6, K_1^7 共 12 比特，并计算 ΔX_2. 为使得 ΔX_2 只在第 3、1 两个位置具有非零差分，ΔX_1 经轮函数加密后的输出差分必须与 ΔX_0 另外 3 个位置的差分值匹配(图中用绿色标注)，即要求 $\Delta S(K_1^0 \oplus X_1^0)=\Delta X_0^0, \Delta S(K_1^6 \oplus X_1^6)=\Delta X_0^5, \Delta S(K_1^7 \oplus X_1^7)=\Delta X_0^3$. 因此第(2)(a)步需要猜测 K_1^0, K_1^6, K_1^7 共 12 比特的子密钥，差分值需要在 3 个位置匹配相应的比特数也为 12，2^{50} 选择明文对经 1 轮过滤后剩下的选择明文数降为 $2^{50-12}=2^{38}$，其差分满足 $(\Delta X_2, \Delta X_1)=(0000*0*0, **00000*)$. 由于数据量为 2^{50} 对明文，密钥穷举量为 12 比特，故计算量约为 $2^{50}\times 2^{12}\times 2=2^{63}$ 次 1 轮加密.

类似的，第(2)(b)步需要先猜测子密钥 K_2^1, K_2^3，并计算出 ΔX_3，使得 $\Delta X_3=00000*00$，即要求满足 $\Delta S(K_2^3 \oplus X_2^3)=\Delta X_1^7, \Delta S(K_2^1 \oplus X_2^1)=\Delta X_1^6$. 然而为了计算出 X_2^3, X_2^1，我们还需进一步猜测子密钥 K_1^2, K_1^3 的值(第 1 轮变换中用蓝色加数字 2 标注)，其中 $S(K_1^2 \oplus X_1^2) \oplus X_0^1=X_2^3, S(K_1^3 \oplus X_1^3) \oplus X_0^7=X_2^1$. 因此第(2)(b)步需要猜测 $K_1^2, K_2^3, K_1^3, K_2^1$ 共 16 比特的子密钥，差分值需要在 2 个位置匹配相应的比特数为 8，经第 2 轮过滤后剩下的选择明文数降为 $2^{38-8}=2^{30}$，其差分满足 $(\Delta X_3, \Delta X_2)=(00000*00, 0000*0*0)$ 由于上一步剩余的数据量为 2^{38} 对明文，密钥穷举量共 16 + 12 = 28 比特，故计算量约为 $2^{38}\times 2^{28}\times 2=2^{67}$ 次 1 轮加密.

同理可以分析第(2)(c)和(2)(d)步的复杂度，需要说明的是，第(2)(d)步直接计算 ΔX_5 仅需要用到子密钥 K_4^5，差分值满足 1 个匹配关系 $\Delta S(K_4^5 \oplus X_4^5)=\Delta X_3^2$，然而为了得到 X_4^5 的值，还需要进一步知道 K_3^7, K_2^6, K_1^4 的值. 另一方面，由子密钥间的依赖关系有 $K_4^5=K_1^7, K_3^7=K_2^1, K_2^6=K_1^0$. 它们在前 3 轮已被穷举，故第(2)(d)步实际需要穷举的子密钥只有一个，即 K_4^5，共 4 比特，其余数据同上可得.

从表 1 可以看出，攻击的时间复杂度主要集中在在第(2)(c)和(2)(d)步，约相当于 $2\times 2^{71}/21=2^{67.61}$ 次 21 轮加密. 由于攻击中总共需要猜测 72 比特子密钥，故存储复杂度为 2^{72}.

上述密钥恢复过程中前 4 轮可以得到 44 比特子密钥 $K_1^{7,6,4,3,2,1,0}, K_2^{3,1,0}, K_3^2$，后 3 轮可以得到 28 比特子密钥 $K_{21}^{7,6,4,1,0}, K_{20}^{6,4}, K_{19}^4$. 利用第 2 节给出的子密钥间的迭代关系，由前 44 比特子密钥可以进一步恢复出初始密钥的 44 比特 $K_{79-72,67-48,39-36,31-24,11-8}$，剩余 36 比特可以遍历，计算复杂度远小于 $2^{67.61}$ 次 21 轮加密，因此攻击的总复杂度仍然为 $2^{67.61}$ 次 21 轮加密. 这些轮子密钥与初始密钥的具体对应关系如表 2.

表 2 前 4 轮子密钥与初始密钥的对应关系

恢复的子密钥	对应的原始密钥比特	恢复的子密钥	对应的原始密钥比特
$K_1^7(\kappa_1^{19})$	[79,78,77,76]	$K_1^0(\kappa_1^{12})$	[51,50,49,48]
$K_1^6(\kappa_1^{18})$	[75,74,73,72]	$K_2^3(\kappa_2^{15}=\kappa_1^9)$	[39,38,37,36]
$K_1^4(\kappa_1^{16})$	[67,66,65,64]	$K_2^1(\kappa_2^{13}=\kappa_1^7 \oplus S(\kappa_1^{13}))$	[31,30,29,28](κ_1^7)
$K_1^3(\kappa_1^{15})$	[63,62,61,60]	$K_2^0(\kappa_2^{12}=\kappa_1^6 \oplus \kappa_1^{16} \oplus (0100)_2)$	[27,26,25,24](κ_1^6)
$K_1^2(\kappa_1^{14})$	[59,58,57,56]	$K_3^2(\kappa_3^{14}=\kappa_2^8=\kappa_1^2)$	[11,10,9,8]
$K_1^1(\kappa_1^{13})$	[55,54,53,52]		

表 2 中 $\kappa_i=(\kappa_i^{19},\kappa_i^{18},\cdots,\kappa_i^0)$ 为 i 轮时密钥寄存器的 4 比特块表示，$\kappa_1=(\kappa_1^{19},\kappa_1^{18},\cdots,\kappa_1^0)$ 为初始密钥.

攻击中需要的明文对数为 $N=2^{50}$，由于用到了 44 比特数据筛选，需要的原始明文对数为 $2^{50}\times 2^{44}=2^{94}$. 注意到要求明文差分满足 $\Delta P=(**00000*,*0*0*0**)$，因此可以选择形如 $(**u_1u_2u_3u_4u_5*,*v_1*v_2*v_3**)$ 的明文结构，其中 u_i,v_i 取固定值，* 处取值任意，每个这样的结构可以产生 $2^{32}\times 2^{31}=2^{63}$ 个具有给定输入差分的明文对，取 $2^{94}/2^{63}=2^{31}$ 个这样的结构，即可满足要求的明文对数量. 因此数据复杂度为 $D=2^{31}\times 2^{32}=2^{63}$ 个选择明文.

3.3　更多轮数 LBlock-s 的不可能差分分析

当考虑更多轮 LBlock-s 的不可能差分分析时，我们发现无论是前面还是后面再多添加 1 轮，部分加密和部分解密中需要猜测的子密钥与全部原始密钥比特都有关，即需要猜测的密钥比特总数将达到 80 比特，然而采用部分匹配技术，利用 Boura[10] 等提出的不可能差分分析改进方案的一般模型，我们仍然可以给出直到 23 轮 LBlock-s 的低于密钥穷举量的不可能差分分析结果. 具体结论见表 3.

表 3　22 和 23 轮 LBlock-s 不可能差分分析结果

轮数	N	r_{in}/r_{out}	$\lvert\Delta_{in}\rvert/\lvert\Delta_{out}\rvert$	c_{in}/c_{out}	$\lvert k_{in}\cup k_{out}\rvert$	时间复杂度	数据量	存储量
22	2^{56}	4/4	32/32	28/28	80	$2^{79.10}$	2^{57}	2^{56}
	2^{57}	**4/4**	**32/32**	**28/28**	**80**	**$2^{78.86}$**	**2^{58}**	**2^{57}**
	2^{58}	4/4	32/32	28/28	80	$2^{79.63}$	2^{59}	2^{58}
23	**2^{72}**	**5/4** **4/5**	**48/32** **32/48**	**44/28** **28/44**	**80**	**$2^{78.60}$**	**2^{57}**	**2^{72}**
	2^{73}	5/4 4/5	48/32 32/48	44/28 28/44	80	$2^{79.08}$	2^{58}	2^{73}

表 3 中 N 为具有指定输入差分和输出差分的明文对数，$\lvert\Delta_{in}\rvert$ 和 $\lvert\Delta_{out}\rvert$ 分别为输入差分 Δ_{in} 和输出差分 Δ_{out} 中非 0 比特数，c_{in} 和 c_{out} 分别表示部分加密和部分解密过程中需要满足过滤条件的比特，$\lvert k_{in}\cup k_{out}\rvert$ 表示部分加密和部分解密过程中需要猜测的实际密钥比特数.

由文献[10]的结论知，不可能差分分析改进方案的的存储复杂度为 $\min\{N,2^{\lvert k_{in}\cup k_{out}\rvert}\}$，数据复杂度为

$$C_N=\max\{\min_{\Delta\in\{\Delta_{in},\Delta_{out}\}}\{\sqrt{N2^{n+1-\lvert\Delta\rvert}}\},N2^{n+1-\lvert\Delta_{in}\rvert-\lvert\Delta_{out}\rvert}\}<2^n \tag{1}$$

计算复杂度为

$$T_{comp}=\left(C_N+\left(N+N\cdot\frac{2^{\lvert k_{in}\cup k_{out}\rvert}}{2^{(c_{in}+c_{out})}}\right)C_E'+2^{\lvert K\rvert}P\right)C_E \tag{2}$$

其中 n 为明文块大小，C_E' 为部分加密和部分解密过程的计算量占整体计算量的比例，C_E 为整体计算量大小，P 为经 N 个明文对筛选后随机候选密钥被保留下来的概率

$$P=(1-2^{-(c_{in}+c_{out})})^N\approx e^{-N/2^{(c_{in}+c_{out})}} \tag{3}$$

下面以 22 轮为例给出各行数据的计算过程，本文中 $n=64$，$\lvert K\rvert=80$. C_E' 可以近似用参与计算的 S 盒个数来估计，由于部分加密和部分解密过程中参与计算的 S 盒总数为 28，故约相当于 $C_E'=28/(8\times 22)\approx 2^{-2.65}$ 次 22 轮加密. 此外，由于部分加密和部分解密过程中需要满足的过滤条件比特数为 56，因此实时攻击需要的明文对数 N 至少为 2^{56}.

当 $N=2^{56}$ 时，由式(1)、式(2)和式(3)可知，需要的原始明文个数为

$$C_N=\max\{\sqrt{N2^{64+1-32}},N2^{64+1-32-32}\}=2N=2^{57}$$

概率 $P=(1-2^{-(c_{in}+c_{out})})^N\approx e^{-1}\approx 2^{-1.44}$，计算复杂度为

$$T_{comp}=(2^{57}+(2^{80-56}\times 2^{56}+2^{56})\times 2^{-2.65}+2^{80}\times 2^{-1.44})C_E\approx 2^{79.08}C_E$$

当 $N=2^{57}$ 时，由式(1)、式(2)和式(3)知，需要的原始明文个数为

$$C_N=\max\{\sqrt{N2^{64+1-48}},N2^{64+1-32-32}\}=2N=2^{58}$$

概率 $P=(1-2^{-(c_{in}+c_{out})})^N\approx e^{-2}\approx 2^{-2.88}$，计算复杂度为

$$T_{comp}=(2^{58}+(2^{80-56}\times 2^{57}+2^{57})\times 2^{-2.65}+2^{80}\times 2^{-2.88})C_E\approx 2^{78.86}C_E$$

当 $N=2^{58}$ 时，由式(1)、式(2)和式(3)知，需要的原始明文个数为

$$C_N=\max\{\sqrt{N2^{64+1-48}},N2^{64+1-32-32}\}=2N=2^{59}$$

概率 $P=(1-2^{-(c_{in}+c_{out})})^N\approx e^{-4}\approx 2^{-5.76}$，计算复杂度为

$$T_{comp}=(2^{59}+(2^{80-56}\times 2^{58}+2^{58})\times 2^{-2.65}+2^{80}\times 2^{-5.76})C_E\approx 2^{79.63}C_E>2^{78.86}C_E$$

因此当 $N=2^{57}$ 时攻击效果最好，此时计算复杂度约相当于 $2^{78.86}$ 次 22 轮加密，数据复杂度约需要 2^{58} 个选择明文，存储复杂度为 2^{57} 个明文对.

3.4　与 LBlock 算法不可能差分分析结果的比较

通过对 LBlock 和 LBlock-s 密钥扩展算法的分析我

们发现,采用 LBlock-s 的密钥扩展算法,其密钥差分扩散更快,这也使得相关密钥条件下利用不可能差分分析方法得不到与 LBlock 相同的分析效果. 在单密钥条件下,表 4 列出了 LBlock 和 LBlock-s 算法不可能差分分析结果的对比.

表 4 LBlock-s 与 LBlock 的不可能差分分析结果对比

	轮数	猜测密钥比特数	时间复杂度	数据量	空间复杂度	参考文献
LBlock	21	72	$2^{73.7}$	$2^{62.5}$ CP	$2^{52.5}$	[8]
	22	76	$2^{79.28}$	2^{58} CP	2^{76}	[9]
	22	71	$2^{71.53}$	2^{60} CP	2^{59}	[10]
	23	73	$2^{75.36}$	2^{59} CP	2^{74}	[10]
LBlock-s	21	72	$2^{67.71}$	2^{63} CP	2^{72}	3.2 节
	22	80	$2^{78.86}$	2^{58} CP	2^{57}	3.3 节
	23	80	$2^{78.60}$	2^{57} CP	2^{57}	3.3 节

从表 4 我们可以看到,利用不可能差分分析方法至多可以对 23 轮 LBlock 和 LBlock-s 算法实施低于密钥穷举的分析,其中 21 轮密码分析的效果大致相当,但对 22 轮和 23 轮算法的分析,LBlock-s 算法需要猜测所有 80 比特密钥,攻击效果远不如 LBlock 算法.

4 小结

本文研究了减轮 LBlock-s 算法的不可能差分分析. 通过选择合适的不可能差分特征,利用密钥扩展算法部分子密钥间仍然存在的依赖关系,我们给出了对 21 轮 LBlock-s 算法的不可能差分分析,攻击需要的时间复杂度约为 $2^{67.61}$ 次 21 轮加密,需要的数据量为 2^{63} 个选择明文,攻击中需要猜测的子密钥比特数为 72 比特. 利用 Boura 等优化不可能差分攻击的一般模型,我们给出了直到 23 轮 LBlock-s 算法的不可能差分分析,利用部分匹配技术,攻击的计算复杂度均低于密钥穷举量. 利用本文得到的研究成果,我们下一步还将进一步考虑 LAC 整体算法的安全性分析.

参考文献

[1] Cryptographic Competitions. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness [DB/OL]. http://competitions.cr.yp.to/caesar.html, 2016-03-01.

[2] ZHANG Lei, WU Wenling, WANG Yanfeng, et al. LAC: A Lightweight Authenticated Encryption Cipher. Submission to CAESAR [R/OL]. http://competitions.cr.yp.to/round1/lacv1.pdf, 2014-03-15.

[3] WU Wenling, ZHANG Lei. LBlock: a lightweight block cipher [A]. Applied Cryptography and Network Security (LNCS 6715) [C]. Berlin Heidelberg: Springer, 2011. 327-344.

[4] WANG Yanfeng, WU Wenling, YU Xiaoli, et al. Security on Lblock against biclique cryptanalysis [A]. Information Security Applications (LNCS 7690) [C]. Berlin Heidelberg: Springer, 2012. 1-14.

[5] BIHAM E, BIRYUKOV A, SHAMIR A. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials [A]. Advances in Cryptology-EUROCRYPT'99 (LNCS 1592) [C]. Berlin Heidelberg: Springer, 1999. 12-23.

[6] KNUDSEN L R. DEAL-a 128-bit Block Cipher [R]. Norway: Department of Informatics, University of Bergen, 1998.

[7] KIM J, HONG S, et al. Impossible differential cryptanalysis for block structures [A]. Progress in Cryptology-INDOCRYPT 2003 (LNCS 2904) [C]. Berlin Heidelberg: Springer, 2003. 82-96.

[8] LIU Ya, GU Dawu, LIU Zhiqiang, LI Wei. Impossible differential attacks on reduced-round Lblock [A]. Information Security Practice and Experience (LNCS 7232) [C]. Berlin Heidelberg: Springer, 2012. 97-108.

[9] KARAKOC F, DEMIRCI H, HARMANC A E. Impossible differential cryptanalysis of reduced-round Lblock [A]. Information Security Theory and Practice Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems (LNCS 7322) [C]. Berlin Heidelberg: Springer, 2012. 179-188.

[10] BOURA C, NAYA-PLASENCIA M, SUDER V. Scrutinizing and improving impossible differential attacks: applications to CLEFIA, camellia, Lblock and simon [A]. Advances in Cryptology-ASIACRYPT 2014 (PART I, LNCS 8873) [C]. Berlin Heidelberg: Springer, 2014. 179-199.

[11] MINIER M, NAYA-PLASENCIA M. A related key impossible differential attack against 22 rounds of the lightweight block cipher Lblock [J]. Information Processing Letters, 2012, 112(16): 624-629.

[12] LIU Shusheng, GONG Zheng, WANG Libin. Improved related-key differential attacks on reduced-round Lblock [A]. Information and Communications Security (LNCS 7618) [C]. Berlin Heidelberg: Springer, 2012. 58-69.

[13] SUN Siwei, HU Lei, WANG Peng, et al. Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, Lblock, DES (L) and other bit-oriented block ciphers [A]. Advances in Cryptology-ASIACRYPT 2014 (PART I, LNCS 8873) [C]. Berlin Heidelberg: Springer, 2014. 158-178.

[14] 黄永洪,郭建胜,罗伟. LBlock 算法的相关密钥-不可能差分攻击 [J]. 电子学报, 2015, 43(10): 1948-1953.
HUANG Yong-hong, GUO Jian-sheng, LUO Wei. Relat-

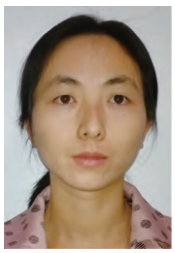
ed-key impossible differential attacks on Lblock [J]. Acta Electronica Sinica, 2015, 43(10): 1948 – 1953. (in Chinese)

[15] 彭昌勇,朱创营,黄莉,等. 扩展的代数侧信道攻击及其应用 [J]. 电子学报,2013,41(5):859 – 864.
PENG Chang-yong, ZHU Chuang-ying, HUANG Li, et al. Extended algebraic-side channel attack and its application [J]. Acta Electronica Sinica, 2013, 41(5): 859 – 864. (in Chinese)

[16] 彭昌勇,朱创营,黄莉,等. 对分组密码的形式化函数分析及其应用 [J]. 电子学报,2013,41(11):2314 – 2316.
PENG Chang-yong, ZHU Chuang-ying, HUANG Li, et al. Formal function cryptanalysis of block cipher and its application [J]. Acta Electronica Sinica, 2013, 41(11): 2314 – 2316. (in Chinese)

作者简介

贾　平　男,1987 年生于甘肃张掖,硕士研究生. 研究方向为分组密码设计与分析.
Email: xxgcjp@163.com



徐　洪　女,1979 年生于湖北随州,博士,硕士生导师. 研究方向为对称密码的设计与分析.
E-mail: xuhong0504@163.com

来学嘉　男,1954 年 6 月生,上海交通大学教授,博士生导师. 研究方向为密码学与信息安全.
E-mail: lai-xj@cs.sjtu.edu.cn