LBlock 算法的改进中间相遇攻击

郑雅菲[1),2)] 吴文玲[1)]

[1)](中国科学院软件研究所可信计算与信息保障实验室 北京 100190)

[2)](中国科学院大学研究生院 北京 100049)

摘 要 LBlock 算法是 2011 年在 ACNS 会议上提出的轻量级分组密码算法,目前已存在 17 轮、19 轮 LBlock 算法的中间相遇攻击.文中评估 LBlock 算法在预建表中间相遇攻击下的安全性.预建表中间相遇攻击提出并发展于 AES 算法(高级加密标准)的安全性分析,是近些年密码分析中的一个研究热点.预建表中间相遇攻击属于典型的区分器类攻击,包含离线和在线两个阶段.文中通过综合离线阶段区分器的建立过程和在线阶段密钥的恢复过程,利用程序搜索 LBlock 算法有效区分器与对应初始密钥的最优攻击参数.结果表明,LBlock 算法存在 11 轮区分器,21 轮 LBlock 算法不抵抗预建表中间相遇攻击,攻击的数据复杂度仅为 $2^{34.1}$ 选择明文,计算复杂度为 $2^{75.8}$ 次 21 轮加密,存储复杂度为 $2^{74.8}$ 个 64 比特块.与 LBlock 算法已有中间相遇攻击相比,文中将攻击轮数由 19 轮扩展至 21 轮,刷新了 LBlock 算法在中间相遇攻击下的安全性评估结果.与不可能差分、积分分析等其他分析结果相比,文中攻击具有显著的低数据复杂度,在实际攻击环境下具有重要意义.此外,为了提高 LBlock 密钥扩展算法的扩散速度,汪艳凤等人提出了一种新的密钥扩展算法.文中评估了采用新的密钥扩展算法的 LBlock 在预建表中间相遇攻击下的安全性,并成功得到了复杂度优于穷举搜索的 20 轮攻击,结果显示新的密钥扩展算法以 1 轮的优势增强了 LBlock 算法抵抗此类攻击的能力.

关键词 分组密码;LBlock;中间相遇攻击;区分器;数据复杂度;密钥扩展算法

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2017.01080

Improved Meet-in-the-Middle Attack of LBlock Cipher

ZHENG Ya-Fei[1),2)] WU Wen-Ling[1)]

[1)](*Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing* 100190)

[2)](*Graduate University of Chinese Academy of Sciences, Beijing* 100049)

Abstract LBlock is a lightweight block cipher proposed in the ACNS (Applied Cryptography and Network Security) conference in 2011. There already exist meet-in-the-middle (MITM) attacks on 17-round and 19-round LBlock. In this paper, the security of LBlock under MITM attack with precomputed table is evaluated. MITM attack with precomputed table, which is introduced and improved during its application to block cipher AES (Advanced Encryption Standard), has been a hot topic in cryptanalysis in recent years. MITM attack with precomputed table is a typical distinguisher-based attack and consists of two phases of offline and online. Combining the offline distinguisher building phase and the online key recovering phase, we search for the best parameters for the distinguisher and the corresponding online key guess by programming. The result shows that 11-round distinguishers of LBlock can be constructed and 21-round LBlock is not secure under MITM attack with precomputation table. The data complexity is only $2^{34.1}$ chosen plaintexts, the time complexity is $2^{75.8}$ 21-round encryptions, and the memory complexity is $2^{74.8}$ 64 bit blocks. Compared with existing MITM attacks of LBlock, the round number is extended from 19 to 21, which has refreshed the security evaluation of LBlock under MITM attack. Compared with other

收稿日期:2016-04-13;在线出版日期:2016-09-29.本课题得到国家"九七三"重点基础研究发展规划项目基金(2013CB338002)、国家自然科学基金(61272476,61672509,61232009)资助.**郑雅菲**,女,1988 年生,博士,主要研究方向为分组密码.E-mail:zhengyafei@tca.iscas.ac.cn.**吴文玲**,女,1966 年生,博士,研究员,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为对称密码学.

cryptanalysis, such as impossible differential attack and integral attack, the data complexity of our attack is significantly low, which is important for the practical attack of LBlock. What's more, in order to speed up the diffusion of the key schedule of LBlock, Wang et al. proposed a new key schedule to against the biclique attack. We also evaluate the security of LBlock with new key schedule under MITM attack, and a 20-round attack with complexity lower than brute force attack is presented, which means that the new key schedule improves the security of LBlock under MITM attack by 1 round.

Keywords　block cipher; LBlock; meet-in-the-middle attack; distinguisher; data complexity; key schedule

1　引　言

近些年来，现实需求使得分组密码的设计需要具备更多的功能与特性，如传感器、智能卡等资源受限设备的广泛使用要求设计者提供以硬件实现代价低为主要参考标准的轻量级分组密码算法. LBlock 算法[1]是由 Wu 等人 2011 年提出的一个轻量级分组密码算法，其分组长度为 64 比特，密钥长度为 80 比特，采用 32 轮变体 Feistel 结构，具有较高的软硬件实现效率.

LBlock 算法自发布以来，受到了国际密码学界的广泛关注，目前已存在大量针对缩减轮数 LBlock 算法的安全性分析，如中间相遇攻击[2-3]、不可能差分分析[4-5]、积分分析[6]、多维零相关线性分析[7]，以及双系攻击[8]等. 其中双系攻击依赖对密钥空间的划分，可以看做优化的穷搜攻击，属于改进中间相遇攻击的一种. 为了提高 LBlock 密钥扩展算法的扩散速度，增强 LBlock 抵抗双系攻击的能力，汪艳凤等人提出了一种新的密钥扩展算法，为叙述方便，下文称采用新的密钥扩展算法的 LBlock 算法为 LBlock* 算法，新的密钥扩展算法同时考虑软件实现速度与相关密钥下算法安全性，可有效增强 LBlock 算法抵抗双系攻击的能力，其具体描述将在 5.1 节中给出.

在 LBlock 算法已有的分析结果中，除中间相遇攻击外，其他几种攻击均需要 2^{60} 左右的数据量，接近全密码本. 随着计算机技术的迅猛发展，CPU 的计算能力在逐步加速，可用的存储空间正逐步加大. 高计算复杂度与高存储的弊端可借助并行计算等高性能算法得以缓解. 但是，实际攻击环境中，获取大量所需的明密文一直是攻击实现的最大障碍. 因此，低数据量下的攻击对于算法的安全性有着重要意义. 中间相遇攻击作为一种低数据复杂度分析方法，也越来越受到关注.

中间相遇攻击由 Diffie 和 Hellman[9]于 1977 年针对多重 DES(Data Encryption Standard)提出，用来衡量单密钥集合下算法的安全性. 基础中间相遇攻击的思想是寻找尽可能长的独立密钥集合，在独立密钥集合下分别对已知明密文对进行加解密操作，再通过选定中间变量进行匹配来筛选正确密钥. 攻击过程可分为两个阶段：中间相遇阶段通过部分加解密匹配过滤一部分错误密钥，减小密钥空间；密钥检测阶段穷搜剩余密钥，利用新的明密文对确定唯一正确密钥. 基础的中间相遇攻击需要对密钥空间进行独立子集合划分，通常要求加密结构和密钥扩展算法简单且扩散速度慢，这种严格要求从一定程度上限制了中间相遇攻击的应用范围.

通过放宽对密钥集合划分或筛选条件的限制，密码学者提出了多种改进的中间相遇攻击. 改进的中间相遇攻击大体可分为两类，基于密钥划分的中间相遇攻击和基于区分器的中间相遇攻击. 在基于区分器的中间相遇攻击中，一种引起广泛关注的方法是利用 δ-集构造区分器的预建表中间相遇攻击. 2008 年 Demirci 等人[10]通过预先建立哈希表的方式构造 AES 的 4 轮区分器，并成功得到了 7 轮 AES-192 和 8 轮 AES-256 的中间相遇攻击；2010 年，Dunkelman 等人[11]提出三种重要技术，即多重集技术、差分枚举技术和密钥桥技术来进一步优化预建表中间相遇攻击，其中差分枚举技术利用截断差分路径的概念，将 4 轮 AES 区分器涉及的未知参数个数由 25 个字节降至 16 个字节，大大减少了攻击所需的存储与预计算复杂度；2013 年 Derbez 等人[12]利用有效建表技术将 4 轮 AES 区分器涉及的未知参数个数由 16 个字节降至 10 个字节；2014 年 Li 等人[13]利用密钥筛技术，在区分器构造过程中找到了

两个密钥字节的匹配,可进一步减少存储.预建表中间相遇攻击提出并发展于 AES 算法的安全性分析,得到了目前 AES 算法单密钥集下的最优分析结果.

预建表中间相遇攻击广泛适用于 SPN 结构的分组密码,对很多密码算法给出了更有效的分析结果.后续实践证明预建表中间相遇攻击亦适用于 Feistel 结构密码[14-17],值得注意的是,在一般 SPN 结构下可有效减少区分器中未知参数个数的差分枚举技术并不普遍适用于 Feistel 结构密码算法,如 LBlock 算法.在实际分析过程中需要根据算法本身结构特点选择是否使用此技术.

本文将预建表中间相遇攻击应用于 LBlock 算法,重新评估 LBlock 算法在中间相遇攻击下的安全性.通过综合离线阶段区分器的建立过程和在线阶段密钥的恢复过程,以最长攻击轮数为目标,利用程序搜索区分器和初始密钥的最优参数,并以一组实际参数为例,给出了数据复杂度为 $2^{34.1}$ 选择明文,计算复杂度为 $2^{75.8}$ 次 21 轮加密,存储为 $2^{74.8}$ 个 64 比特块的 21 轮预建表中间相遇攻击.与 LBlock 算法现有中间相遇攻击相比,本文成功将攻击轮数由 19 轮扩展至 21 轮,刷新了 LBlock 算法在中间相遇攻击下的安全性评估结果;与积分分析、多维零相关线性分析、不可能差分分析等结果相比,本文攻击具有显著的低数据复杂度,在实际攻击环境中具有重要意义.

本文还评估了 LBlock* 算法在预建表中间相遇攻击下的安全性.结果证明 20 轮 LBlock* 算法存在复杂度优于穷举搜索的预建表中间相遇攻击,这说明汪艳凤等人提出的新的密钥扩展算法不仅可以改善 LBlock 算法在双系攻击下的安全性,也以 1 轮的优势有效提高了 LBlock 算法抵抗预建表中间相遇攻击的能力.

本文第 2 节简单介绍 LBlock 算法;第 3 节介绍预建表中间相遇攻击的主要思想;第 4 节给出完整 21 轮 LBlock 算法的预建表中间相遇攻击,包括攻击采用的搜索策略与最优参数、区分器建立过程、密钥恢复过程以及复杂度分析;第 5 节简要介绍新的密钥扩展算法与 20 轮 LBlock* 算法的预建表中间相遇攻击;第 6 节结束语总结全文.

2 符号说明与 LBlock 算法简介

2.1 符号说明

文中涉及的符号解释如表 1 所示.

表 1 符号解释

符号	解释
P^i	第 i 个明文
X_{2i},X_{2i+1}	第 i 轮加密两个 32 比特输入值
$\Delta X_{2i},\Delta X_{2i+1}$	第 i 轮加密两个 32 比特输入差分
$\Delta X_i[j]$	ΔX_i 的第 j 个半字节
X_{2i}^j,X_{2i+1}^j	P^j 经 i 轮加密后的状态值
K	80 比特主密钥
K_i	第 i 轮轮密钥
$X_i^j[k]$	X_i^j 的第 k 个半字节
$X_i^j[k-l]$	X_i^j 的第 k 个到第 l 个半字节
$K_i[j]$	K_i 的第 j 个半字节
$K_i[j-k]$	K_i 的第 j 个到第 k 个半字节
k_i	初始密钥的一个比特
$\lll i$	左循环移 i 比特
$\oplus$	按比特异或
$[i]_2$	i 的二进制表示
$X\|Y$	连接比特串 X 与 Y

需要注意的是,本文所有计数从 0 开始.

2.2 LBlock 算法简介

LBlock 是 2011 年提出的轻量级分组密码算法,采用 32 轮的变体 Feistel 加密结构,分组长度为 64 比特,密钥长度为 80 比特.记第 i 轮加密输入为 X_{2i},X_{2i+1},输出为 X_{2i+2},X_{2i+3},即明文 $P=X_0\|X_1$,密文 $C=X_{64}\|X_{65}$,加密流程如图 1 所示.

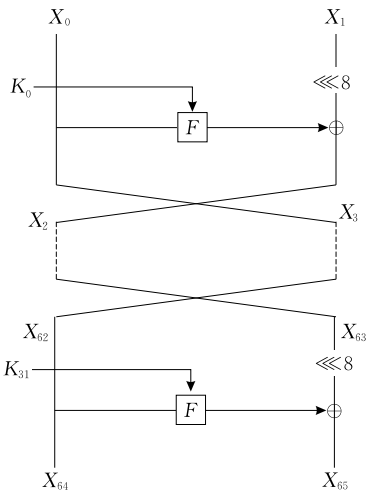

图 1 LBlock 算法加密流程

第 i 轮($i=0,1,\cdots,30$)加密为

$$X_{2i+2}=F(X_{2i},K_i)\oplus(X_{2i+1}\lll 8),$$

$$X_{2i+3}=X_{2i}.$$

最后一轮加密不包含左右分支交换操作:

$$X_{64}=X_{62},$$

$$X_{65}=F(X_{62},K_{31})\oplus(X_{63}\lll 8).$$

解密过程为加密过程的逆,此处不再介绍.

轮函数 F 包含三层操作,即轮密钥加、混淆层和扩散层 P.

轮密钥加操作为当前状态与轮密钥简单的异或和.

混淆层为 8 个 4 比特 S 盒的并置，本文攻击只要求 S 盒满足差分均衡性质，其详细描述可参见 LBlock 算法设计文档[1].

P 为 8 个半字节上的向量置换，其具体变换如图 2 中所示.

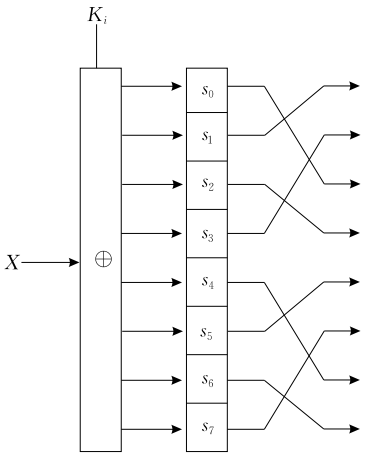

图 2　F 轮函数

为减小硬件实现代价，LBlock 算法采用较简单的密钥扩展算法.

首先将 80 比特主密钥 $K=\{k_{79}k_{78}\cdots k_0\}$ 存入寄存器，第 i 轮加密选择当前寄存器左 32 比特作为轮密钥 $K_i(i=0,1,\cdots,31)$，每取一次密钥寄存器按如下方式更新：

$a.[k_{79}k_{78}\cdots k_0]=[k_{50}k_{49}\cdots k_0k_{79}\cdots k_{51}]$

$b.[k_{79}k_{78}k_{77}k_{76}]=s_9[k_{79}k_{78}k_{77}k_{76}]$

$[k_{75}k_{74}k_{73}k_{72}]=s_8[k_{75}k_{74}k_{73}k_{72}]$

$c.[k_{50}k_{49}k_{48}k_{47}k_{46}]=[k_{50}k_{49}k_{48}k_{47}k_{46}]\oplus[i]_2$

其中 s_9，s_8 均为 4 比特 S 盒.

LBlock 密钥扩展算法包含循环移位、4 比特 S 盒、常数异或三种简单操作，便于硬件实现，但是扩散速度较慢，这个特点导致在 LBlock 密钥相关的攻击中，攻击者可以通过排除密钥冗余的方式减少密钥猜测量，进而增加攻击轮数或改善攻击复杂度.

3　预建表中间相遇攻击简介

预建表中间相遇攻击提出并发展于 AES 的安全性分析，攻击包括两个阶段：离线阶段和在线阶段.

离线阶段通过预先建立哈希表的方式构造区分器. 其一般情形为，选择加密满足特殊性质的明文结构，如 δ-集，将输出某处序列值用中间参数表示，遍历这些中间参数值并建表存储其与输出序列值的对应关系.

如图 3 所示，在线阶段将预建表看作中间 n_2 轮的一个区分器，在其前端与末端分别增加 n_1、n_3 轮.

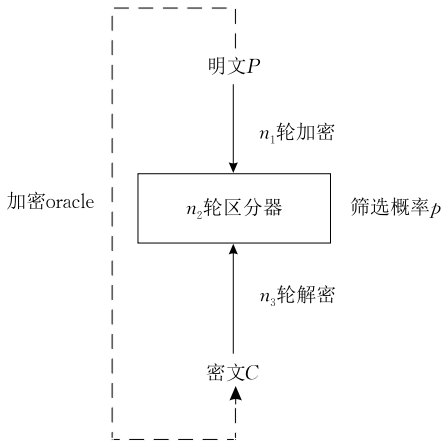

图 3　预建表中间相遇攻击框架

恢复密钥的过程为：

(1) 猜测区分器前端 n_1 轮部分加密涉及的轮密钥，构造 δ-集，获得相应明密文；

(2) 猜测区分器末端 n_3 轮部分解密涉及的轮密钥，解密密文至指定序列值；

(3) 判断解密所得序列值是否与离线阶段建立的哈希表中存储值存在匹配. 若存在，则当前密钥作为正确密钥候选值；若不存在，删除错误密钥；

(4) 在剩余密钥的基础上，通过穷搜的方式恢复完整主密钥.

如果离线阶段区分器的建立利用了截断差分路径，则在线阶段需要对大量明密文对进行筛选来得到一个满足差分路径的正确对，再基于该正确对构造 δ-集. 筛选正确对的过程需要大量的数据和计算量，鉴于 LBlock 算法的结构特点，本文不选用截断差分路径的方法.

预建表中间相遇攻击的计算复杂度由离线阶段和在线阶段两部分组成，通常需要大量的存储来完成建表，这往往是预建表中间相遇攻击的难点问题.

如何选择区分器中 δ-集的活跃位置和输出序列的位置来减少未知参数个数，进而减少离线阶段的存储与计算量，以及如何利用密钥扩展算法排除密钥冗余，减少在线阶段密钥猜测量是攻击成功的关键.

4　21 轮 LBlock 中间相遇攻击

本节介绍 LBlock 算法的预建表中间相遇攻击在程序实现中采用的搜索策略，并给出最长攻击轮数和最小复杂度. 最后以一组具体参数为例，给出 21 轮 LBlock 算法预建表中间相遇攻击的完整过程，攻击的计算复杂度单位采用 21 轮 LBlock 加密，

存储单位采用 64 比特块.

4.1 搜索策略与最优参数

首先简单介绍攻击采用的搜索策略.

(1)离线阶段搜索区分器

构造区分器的关键在于密钥未知的情况下,如何选择 δ-集经过 r 轮加密后合适的输出有序序列,使其所有可能情况区别于随机函数加密的情况.而有序序列取值可由中间轮的某些未知参数决定,这些未知参数的确定即 4.2 节中性质 2 的证明过程.如果在区分器末端选择 m 个半字节构造有序序列,则未知参数个数必须小于 $15m$;同时为了保证攻击的复杂度不超过穷搜 2^{80},参数个数必须小于 $80/4=20$ 个.综合这两点,本文选择两个半字节来构造有序序列,$m=2$.

结合 LBlock 算法结构,以半字节为单位,遍历区分器所有可能的输入输出情况,即输入为一个 δ-集,输出为两个半字节处有序序列,搜索涉及未知参数个数不大于 19 的所有有效区分器.

参数个数不大于 19 满足区分器的筛选概率 $0<p<1$,并且可以保证攻击的计算与存储复杂度不超过穷搜攻击.

(2)在线阶段确定猜测密钥

对(1)中得到的每个区分器,分别在其起始端与末端增加轮数,将区分器起始端延伸至明文、区分器末端延伸至密文,并确定部分加解密过程中需要猜测的轮密钥.为使攻击有效,猜测轮密钥比特数不应超过 76 比特,这是为了保证在线阶段复杂度不超过穷搜复杂度 2^{80}.

为了保证数据复杂度不达到全密码本,明文端不能全活跃.

(3)确定初始密钥,排除密钥冗余

对所有可能的初始密钥,遍历所有可满足(2)中猜测轮密钥已知的初始密钥活跃情况.

初始密钥是主密钥等价的轮密钥,LBlock 密钥扩展算法可逆,可从任意轮密钥出发经过密钥扩展算法和密钥扩展逆算法得到各轮轮密钥.

选取适当的初始密钥可使得在线阶段需要猜测的密钥量最少,进而减少密钥恢复的复杂度.

具体方式为,令初始密钥中部分密钥比特非活跃,且非活跃密钥比特数不应小于 4,若由初始密钥经密钥扩展算法及其逆算法仍可推得部分加解密过程中需猜测的所有轮密钥,则攻击有效.

(4)输出最优参数

选择攻击轮数最长,区分器中未知参数最少,且初始密钥中非活跃密钥比特数最大的一组参数作为结果输出.

上述搜索策略综合离线阶段区分器的建立过程与在线阶段密钥的恢复过程,可保证输出的结果对应最优的攻击.

本文实验结果可手动验证.

结果显示 LBlock 算法存在 8 个 11 轮区分器:

$(0;0,1)$, $(1;3,6)$, $(2;0,1)$, $(3;2,7)$,

$(4;4,5)$, $(5;2,7)$, $(6;4,5)$, $(7;3,6)$,

其中 $(i;j,k)$ 表示区分器输入活跃半字节为 $X_{2n+1}[i]$,输出活跃半字节为 $X_{2n+23}[j]$ 和 $X_{2n+23}[k]$ 的 11 轮区分器,n 为轮数.

这 8 个 11 轮区分器对应的在线阶段密钥猜测量并不相同,这导致了不同的攻击轮数与复杂度.

表 2 给出了搜索程序输出的 8 组最优参数,满足仅在这 8 组参数下攻击的轮数最长,且复杂度最小.其中前端轮数与末端轮数表示区分器两端增加的加解密轮数.非活跃比特表示初始密钥中的未知比特.

表 2 LBlock 预建表中间相遇攻击最优参数

区分器	区分器轮数	参数个数	前端轮数	末端轮数	初始密钥	非活跃密钥比特
(3;2,7)	11	19	4	6	K_{13}	$\{k_{79},\cdots,k_{74},k_2,k_1,k_0\}$
(3;2,7)	11	19	4	6	K_{14}	$\{k_{31},\cdots,k_{23}\}$
(3;2,7)	11	19	4	6	K_{15}	$\{k_{60},\cdots,k_{52}\}$
(3;2,7)	11	19	4	6	K_{16}	$\{k_9,\cdots,k_1\}$
(3;2,7)	11	19	4	6	K_{17}	$\{k_{38},\cdots,k_{30}\}$
(3;2,7)	11	19	4	6	K_{18}	$\{k_{67},\cdots,k_{59}\}$
(3;2,7)	11	19	4	6	K_{19}	$\{k_{16},\cdots,k_8\}$
(3;2,7)	11	19	4	6	K_{20}	$\{k_{45},\cdots,k_{37}\}$

4.2 建立 11 轮区分器

本节介绍 11 轮区分器(3;2,7)的构造过程.

LBlock 加密算法采用 8 个不同的 4 比特 S 盒,预建表中间相遇攻击不涉及这些 S 盒的具体内容,但是要求下述性质成立.

性质 1(S 盒差分均衡性). 已知 S 盒的输入、输出差分 Δ_{in}、Δ_{out},则 $S(x)\oplus S(x\oplus\Delta_{in})=\Delta_{out}$ 平均意义下有一个解.

证明. LBlock 算法混淆层中的任意一个 4 比特 S 盒,对任意输入、输出差分 Δ_{in}、Δ_{out},其差分分布表中满足 $S(x)\oplus S(x\oplus\Delta_{in})=\Delta_{out}$ 的解的个数必为 0、2、4 之一.

对随机给定输入、输出差分对 Δ_{in}、Δ_{out}，$S(x)\oplus S(x\oplus\Delta_{in})=\Delta_{out}$ 解的期望个数为

$$E=\frac{16-a}{16}\left(\frac{b}{16-a}\times 2+\frac{c}{16-a}\times 4\right),$$

其中 a、b、c 分别表示解的数目 0、2、4 出现的次数. 由于 a、b、c 满足 $0a+2b+4c=16$，有 $E=1$. 性质 1 得证. 证毕.

为了构造区分器，给出 LBlock 算法 δ-集的概念.

定义 1. 一个 δ-集为 LBlock 算法的 16 个状态集合，这些状态满足在一个半字节处互不相同(活跃半字节)，在其他半字节处为常值(非活跃半字节).

性质 2. 一个 δ-集 $\{P^0,P^1,\cdots,P^{15}\}$，满足 $X_1[3]$ 活跃，而 $X_0[0\sim7]$ 与 $X_1[0,1,2,4,5,6,7]$ 非活跃，则经过 11 轮 LBlock 加密，输出差分序列：

$$X_{23}^1[2]\oplus X_{23}^0[2],X_{23}^2[2]\oplus X_{23}^0[2],\cdots,X_{23}^{15}[2]\oplus X_{23}^0[2],$$
$$X_{23}^1[7]\oplus X_{23}^0[7],X_{23}^2[7]\oplus X_{23}^0[7],\cdots,X_{23}^{15}[7]\oplus X_{23}^0[7]$$

可完全由以下 19 个参数决定：

$$Y_1^0[1],Y_2^0[0],Y_3^0[2,7],Y_4^0[3,5,6],Y_5^0[0,1,4],$$
$$Y_6^0[0,1,2],Y_7^0[3,5],Y_8^0[1,4],Y_9^0[0,6],$$

其中 $Y_r^0[i]=X_{2r}^0[i]\oplus K_r[i]$.

即 $X_{23}[2]$，$X_{23}[7]$ 处 30 个 4 比特差分序列仅有 $2^{4\times19}=2^{76}$ 种取值，而非 $2^{15\times2\times4}=2^{120}$ 种取值. 区分器的筛选概率为 $2^{76}/2^{120}=2^{-44}$.

证明. 如图 4 所示，其中白色填充部分表示该半字节状态差分为零，斜线部分表示该半字节受区

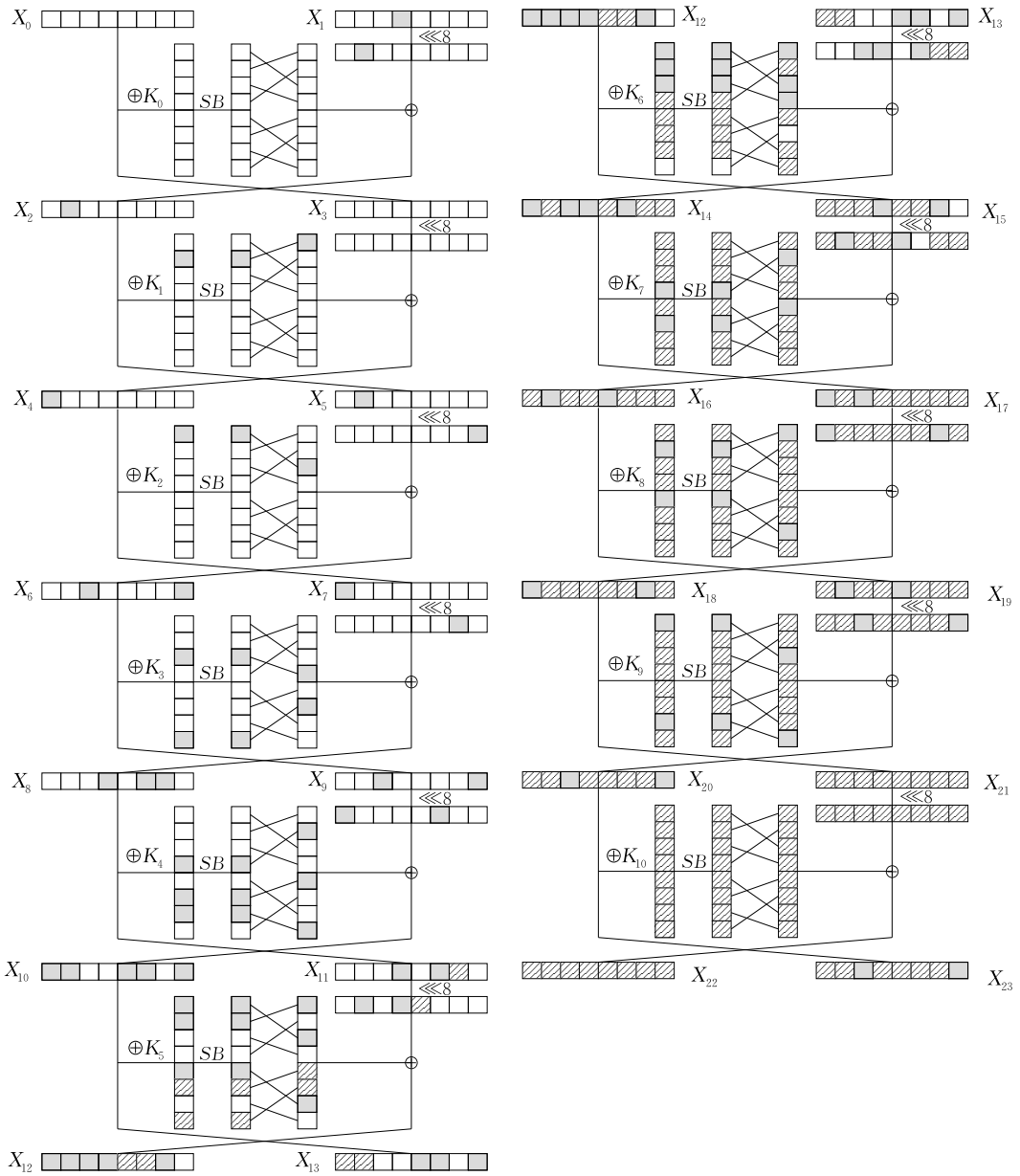

图 4　LBlock 的 11 轮区分器(3;2,7)

分器输入差分影响，黑色填充部分表示该半字节影响 $X_{23}[2]$，$X_{23}[7]$处差分，同时受区分器输入差分影响.

由于该区分器不涉及密钥，可任意选取 11 轮区分器的起始轮，不防设区分器从首轮开始.

明文差分 $P^1\oplus P^0$，$P^2\oplus P^0$，…，$P^{15}\oplus P^0$作为区分器输入差分 ΔX_0，ΔX_1，满足 ΔX_0为零差分，ΔX_1仅在 $\Delta X_1[3]$处为非零差分.

由于密钥加操作不影响差分，零差分经混淆层 S 盒的输出仍为零差分，易得一轮加密后的输出差分 ΔX_2，ΔX_3，即 $\Delta X_2[1]$处为已知非零差分，其他位置均为零差分.

猜测 $Y_1^0[1]=X_2^0[1]\oplus k_1[1]$，则第 2 轮混淆层 S 盒的输入差分已知，且其非零差分处值已知，由性质 1 中 S 盒差分均衡性可知混淆层的输出差分，又因为线性置换层 P 不改变差分值，故第 2 轮加密输出差分 ΔX_4，ΔX_5已知.

第 3 轮加密混淆层输入差分已知，猜测非零差分处的值 $Y_2^0[0]$，由性质 1，S 盒输出差分已知，则第 3 轮输出差分 ΔX_6，ΔX_7已知.

类似的，猜测第 4 轮混淆层输入非零差分处的值 $Y_3^0[2,7]$，第 4 轮输出差分 ΔX_8，ΔX_9已知；猜测第 5 轮混淆层输入非零差分处的值 $Y_4^0[3,5,6]$，第 5 轮输出差分 ΔX_{10}，$\Delta X_{11}[0\sim5,7]$已知；猜测第 6 轮混淆层输入非零差分处的值 $Y_5^0[0,1,4]$，第 6 轮输出差分 $\Delta X_{12}[0,1,2,3,6,7]$，$\Delta X_{13}[2\sim7]$已知；猜测第 7 轮混淆层输入非零差分处的值 $Y_6^0[0,1,2]$，第 7 轮输出差分 $\Delta X_{14}[0,2,3,5]$，$\Delta X_{15}[3,6,7]$已知；猜测第 8 轮混淆层输入非零差分处的值$Y_7^0[3,5]$，第 8 轮输出差分 $\Delta X_{16}[1,4]$，$\Delta X_{17}[0,2]$已知；猜测第 9 轮混淆层输入非零差分处的值 $Y_8^0[1,4]$，第 9 轮输出差分 $\Delta X_{18}[0,6]$，$\Delta X_{19}[1,4]$已知.

猜测第 10 轮混淆层输入非零差分处的值 $Y_9^0[0,6]$，第 10 轮输出差分 $\Delta X_{20}[2,7]$已知，从而 11 轮输出差分 $\Delta X_{23}[2,7]$已知，即 $X_{23}[2]$，$X_{23}[7]$处 30 个 4 比特差分序列仅由猜测的 19 个半字节 $Y_1^0[1]$，$Y_2^0[0]$，$Y_3^0[2,7]$，$Y_4^0[3,5,6]$，$Y_5^0[0,1,4]$，$Y_6^0[0,1,2]$，$Y_7^0[3,5]$，$Y_8^0[1,4]$，$Y_9^0[0,6]$决定. 性质 2 得证. 证毕.

4.3 密钥恢复

基于性质 2 的 11 轮区分器，我们在前端加 4 轮，后端加 6 轮，给出 21 轮 LBlock 算法的中间相遇攻击，如图 5 所示.

攻击过程由离线区分器建立和在线密钥恢复两个阶段组成.

(1) 离线阶段：

计算 120 比特差分序列的所有 2^{76}种可能值，并存储于哈希表中.

(2) 在线阶段：

① 选择明文 $P^0=X_0^0\|X_1^0$，猜测密钥 $K_0[1,7]$，$K_1[0]$，$K_2[2]$，部分加密 P^0得区分器输入端 $X_9^0[3]$；

② 在区分器输入端构造 δ-集，令 $X_9[3]$遍历所有 16 种可能值，具体为令 $X_9^0[3]$异或 15 种非零差分. $X_8[0\sim7]$与 $X_9[0,1,2,4,5,6,7]$为未知常值；

③ 猜测状态 $Y_0^0[3,5,6]$，$Y_1^0[3,5]$，$Y_2^0[3]$，$Y_3^0[3]$，部分解密 δ-集，即可得到明文输入的活跃差分，与 $P^0=X_0^0\|X_1^0$异或即可得符合 11 轮区分器特性的 16 个明文集合；

④ 询问加密 oracle 得明文集相应密文；

⑤ 猜测密钥 $K_{15}[1,7]$，$K_{16}[6,7]$，$K_{17}[2,4,5,7]$，$K_{18}[0,1,2,7]$，$K_{19}[0\sim7]$，$K_{20}[0\sim7]$，部分解密密文得 $X_{31}[2]$，$X_{31}[7]$处差分序列；

⑥ 检查解密所得差分序列是否存在于离线阶段建立的哈希表中. 若存在，则猜测密钥作为正确密钥候选值，否则删除错误密钥.

在剩余密钥基础上穷搜恢复完整主密钥.

易知构造 δ-集时猜测的状态 $Y_0^0[3,5,6]$，$Y_1^0[3,5]$，$Y_2^0[3]$，$Y_3^0[3]$可在密钥 $K_0[0,1,2,3,5,6,7]$，$K_1[0,2,3,5]$，$K_2[2,3]$，$K_3[3]$下经过部分加密 P^0得到，故在线阶段共猜测$(14+28)\times4=168$比特密钥，若不考虑密钥冗余，攻击的计算量超过穷搜.

利用 LBlock 密钥扩展算法扩散较慢的特点，选择 K_{13}作为初始密钥来排除密钥冗余，记 $K_{13}=\{k_{79},k_{78},\cdots,k_0\}$，在线阶段加解密过程所需 168 比特密钥可完全由 K_{13}的 71 比特决定，即 K_{13}的$\{k_{79},k_{78},k_{77},k_{76},k_{75},k_{74},k_2,k_1,k_0\}$这 9 比特密钥不影响部分加解密过程.

如图 6 所示，选择 K_{13}为初始密钥并令$\{k_{79},k_{78},k_{77},k_{76},k_{75},k_{74},k_2,k_1,k_0\}$这 9 个比特未知，在图 6 中用灰色表示. 根据密钥扩展算法，K_{13}依次经过以下操作得到 K_{14}：左循环移 29 比特，高位两个半字节 S 盒，$k_{50}k_{49}k_{48}k_{47}k_{46}$这 5 个比特上常数加. 易知 K_{14}的$\{k_{31},k_{30},k_{29},k_{28},k_{27},k_{26},k_{25},k_{24},k_{23}\}$这 9 个比特是未知的. 逐轮扩展可得到 K_{15}，K_{16}，…，K_{20}的状态.

接下来由 K_{13}经过密钥扩展逆算法推导 K_{12}，K_{11}，…，K_0. K_{13}依次经过以下操作得到 K_{12}：常数加，高位两个半字节 S 盒，右循环移 29 比特. 易知 K_{12}的$\{k_{53},k_{52},k_{51},k_{50},k_{49},k_{48},k_{47},k_{46},k_{45},k_{44},k_{43}\}$

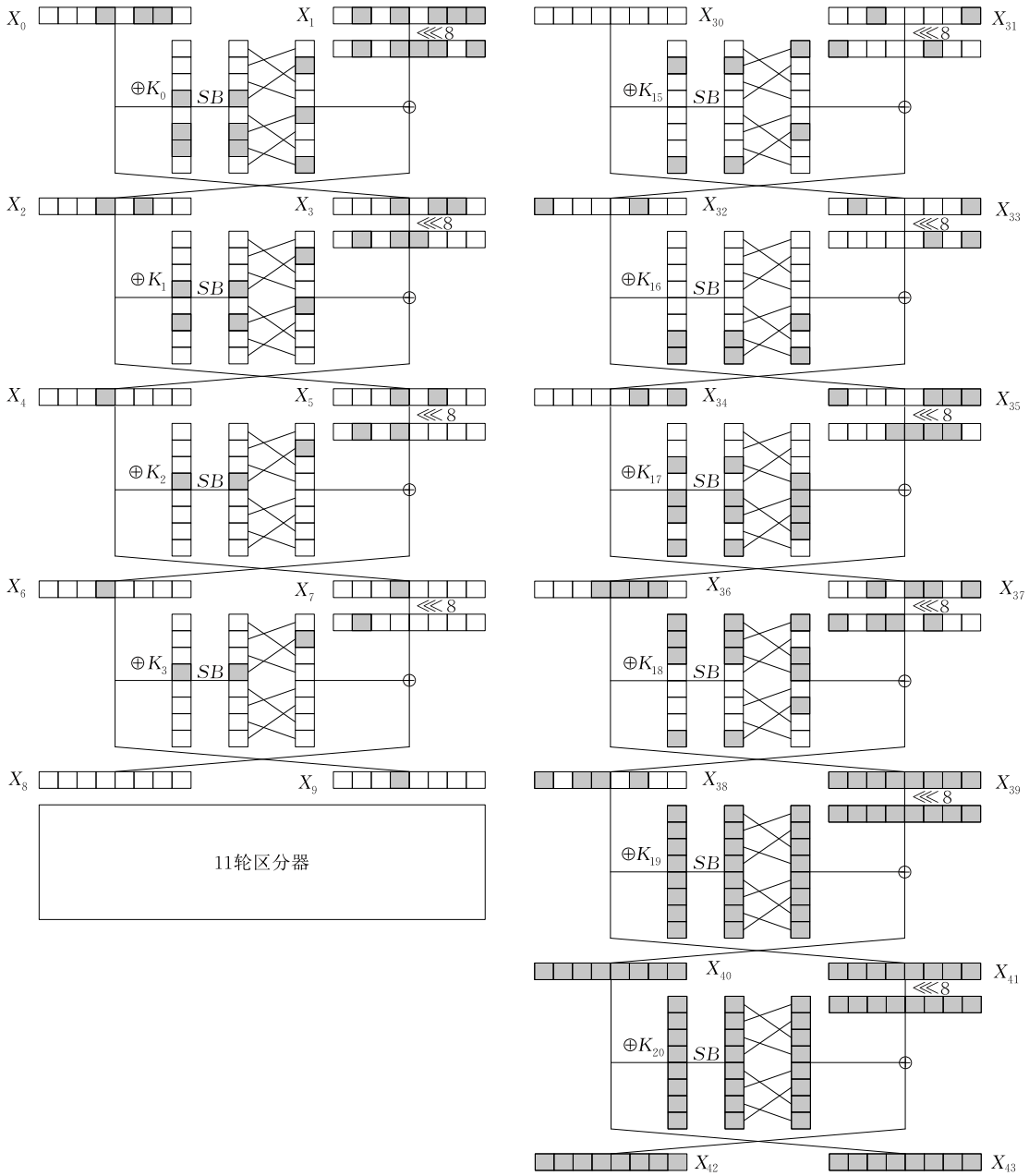

图 5　21 轮 LBlock 算法的中间相遇攻击

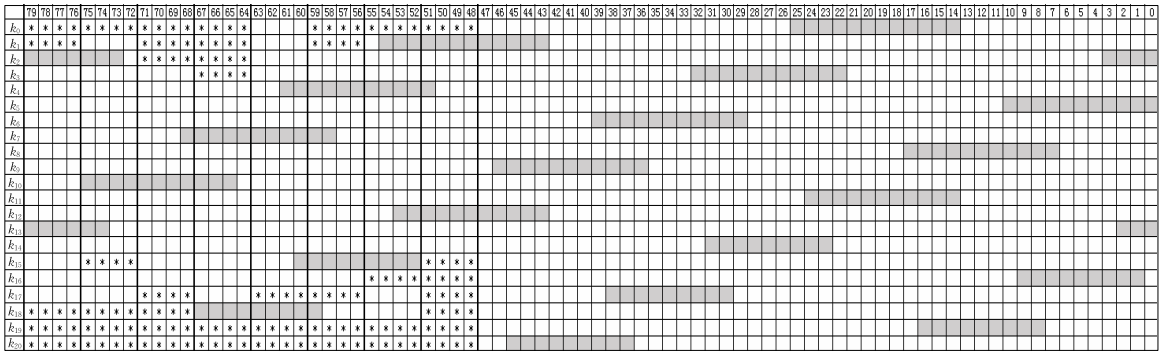

图 6　初始密钥 K_{13} 下 21 轮 LBlock 算法的各轮密钥

这 11 个比特是未知的. 逐轮扩展可以得到 K_{11}，…，K_0 的状态.

将部分加解密过程中涉及的密钥比特在图 6 中用“*”表示，可以发现这些密钥比特不包含任一灰

色未知比特，即选择初始密钥为 K_{13}，且令未知密钥比特为$\{k_{79},k_{78},k_{77},k_{76},k_{75},k_{74},k_2,k_1,k_0\}$时，各轮密钥状态可如下表示：

第 0 轮：1 1 1 1 1 1 1 1
第 1 轮：1 1 1 1 1 1 0 0
第 2 轮：0 0 1 1 1 1 1 1
第 3 轮：1 1 1 1 1 1 1 1
第 4 轮：1 1 1 1 0 0 0 0
第 5 轮：1 1 1 1 1 1 1 1
第 6 轮：1 1 1 1 1 1 1 1
第 7 轮：1 1 0 0 0 0 1 1
第 8 轮：1 1 1 1 1 1 1 1
第 9 轮：1 1 1 1 1 1 1 1
第 10 轮：1 0 0 0 1 1 1 1
第 11 轮：1 1 1 1 1 1 1 1
第 12 轮：1 1 1 1 1 1 0 0
第 13 轮：0 0 1 1 1 1 1 1
第 14 轮：1 1 1 1 1 1 1 1
第 15 轮：1 1 1 1 0 0 0 1
第 16 轮：1 1 1 1 1 1 1 1
第 17 轮：1 1 1 1 1 1 1 1
第 18 轮：1 1 1 0 0 0 1 1
第 19 轮：1 1 1 1 1 1 1 1
第 20 轮：1 1 1 1 1 1 1 1

其中 0 表示不可由当前初始密钥推得的半字节密钥，1 表示可由当前初始密钥推得的半字节密钥.

在线阶段涉及的 42 个密钥字节：

$K_0[0,1,2,3,5,6,7]$，$K_1[0,2,3,5]$，$K_2[2,3]$，$K_3[3]$，$K_{15}[1,7]$，$K_{16}[6,7]$，$K_{17}[2,4,5,7]$，$K_{18}[0,1,2,7]$，$K_{19}[0\sim7]$，$K_{20}[0\sim7]$

可由初始密钥通过密钥扩展算法和逆算法得到.

4.4 复杂度分析

攻击所需数据量为 2^{32} 选择明文. 离线阶段计算复杂度为 $2^4\times2^{76}$ 次部分加密，约为 $2^{76.9}$ 次 21 轮 LBlock 加密. 存储为 2^{76} 个 120 比特差分序列，约为 $2^{76.9}$ 个 64 比特块.

在线阶段计算复杂度为 $2^{71}\times2^4$ 次部分加解密，约为 $2^{72.7}$ 次 21 轮加密.

一次筛选后剩余密钥量为 $2^{71}\times2^{76}/2^{120}=2^{27}$. 在剩余密钥下穷搜 K_{13} 中$\{k_{79},k_{78},k_{77},k_{76},k_{75},k_{74},k_2,k_1,k_0\}$的值，即可以 2^{36} 的计算量恢复 K_{13} 的全部信息，根据 K_{13} 与主密钥的等价性可通过密钥扩展算法恢复主密钥.

为了平衡离线阶段与在线阶段的复杂度，考虑时间存储折中(TMTO)，即离线阶段建表时不存储全部 2^{76} 种差分序列，而是按一定比例存储，不妨设仅存储 2^{76-m} 种差分序列，则离线阶段计算复杂度为 $2^{76.9-m}$ 次 21 轮加密，存储为 $2^{76.9-m}$ 个 64 比特块. 在线阶段需重复 2^m 次来保证攻击的成功率，计算复杂度为 $2^{72.7+m}$ 次 21 轮加密，数据复杂度变为 2^{32+m} 选择明文.

选择 $m=2.1$，则攻击总的计算复杂度为 $2^{76.9-m}+2^{72.7+m}=2^{75.8}$ 次 21 轮加密，存储为 $2^{74.8}$ 个 64 比特块，数据复杂度为 $2^{34.1}$ 选择明文.

5 LBlock* 预建表中间相遇攻击

文献[8]中，作者为了改善 LBlock 算法密钥扩展算法扩散较慢的缺点，增强 LBlock 算法抵抗双系攻击的能力，提出了一种新的密钥扩展算法.

本章节评估 LBlock* 算法在预建表中间相遇攻击下的安全性，判断新的密钥扩展算法是否增强了 LBlock 算法抵抗中间相遇攻击的能力.

5.1 新的密钥扩展算法

新的密钥扩展算法不改变轮密钥的选取方式，仍将 80 比特主密钥 $K=\{k_{79}k_{78}\cdots k_0\}$ 存入寄存器，第 i 轮加密选择当前寄存器左 32 比特作为轮密钥 $K_i(i=0,1,\cdots,31)$，每取一次密钥，寄存器按如下方式更新：

$a.\ [k_{79}k_{78}\cdots k_0]=[k_{55}k_{54}\cdots k_0k_{79}\cdots k_{56}]$

$b.\ [k_{55}k_{54}k_{53}k_{52}]=s[k_{79}k_{78}k_{77}k_{76}]\oplus[k_{55}k_{54}k_{53}k_{52}]$

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

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

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

$c.\ [k_{54}k_{53}k_{52}k_{51}k_{50}]=[k_{54}k_{53}k_{52}k_{51}k_{50}]\oplus[i]_2$

其中 s 为 4 比特 S 盒.

与之前密钥扩展算法相比，新的密钥扩展算法的特点是采用 4 比特倍数的循环移位. 在排除密钥冗余时，可以忽略最后的常数加，将新的密钥扩展算法看作半字节级变换.

5.2 20 轮攻击最优参数

此章节仍采用 4.1 节中介绍的搜索策略. 程序输出的最优参数如表 3 所示.

表 3 20 轮 LBlock* 中间相遇攻击最优参数选取

区分器	参数个数	前端轮数	末端轮数	初始密钥	非活跃密钥比特
(3;2,7)	19	3	6	K_{14}	$\{k_{71},k_{70},k_{69},k_{68}\}$
(3;2,7)	19	3	6	K_{15}	$\{k_{15},k_{14},k_{13},k_{12}\}$
(3;2,7)	19	3	6	K_{16}	$\{k_{39},k_{38},k_{37},k_{36}\}$
(3;2,7)	19	3	6	K_{17}	$\{k_{63},k_{62},k_{61},k_{60}\}$
(3;2,7)	19	3	6	K_{18}	$\{k_7,k_6,k_5,k_4\}$
(3;2,7)	19	3	6	K_{19}	$\{k_{31},k_{30},k_{29},k_{28}\}$

在 11 轮区分器前端加 3 轮，后端加 6 轮，通过猜测相关轮密钥进行部分加解密，匹配区分器，可以得到 20 轮 LBlock* 算法的预建表中间相遇攻击.

5.3　20 轮 LBlock* 的预建表中间相遇攻击

选择 11 轮区分器(3;2,7)，初始密钥 K_{14}，初始密钥非活跃比特为$\{k_{71},k_{70},k_{69},k_{68}\}$，非活跃密钥比特数为 4.

离线阶段 11 轮区分器的建立过程可参考 4.2 节. 20 轮 LBlock* 算法中间相遇攻击的在线阶段如图 7 所示.

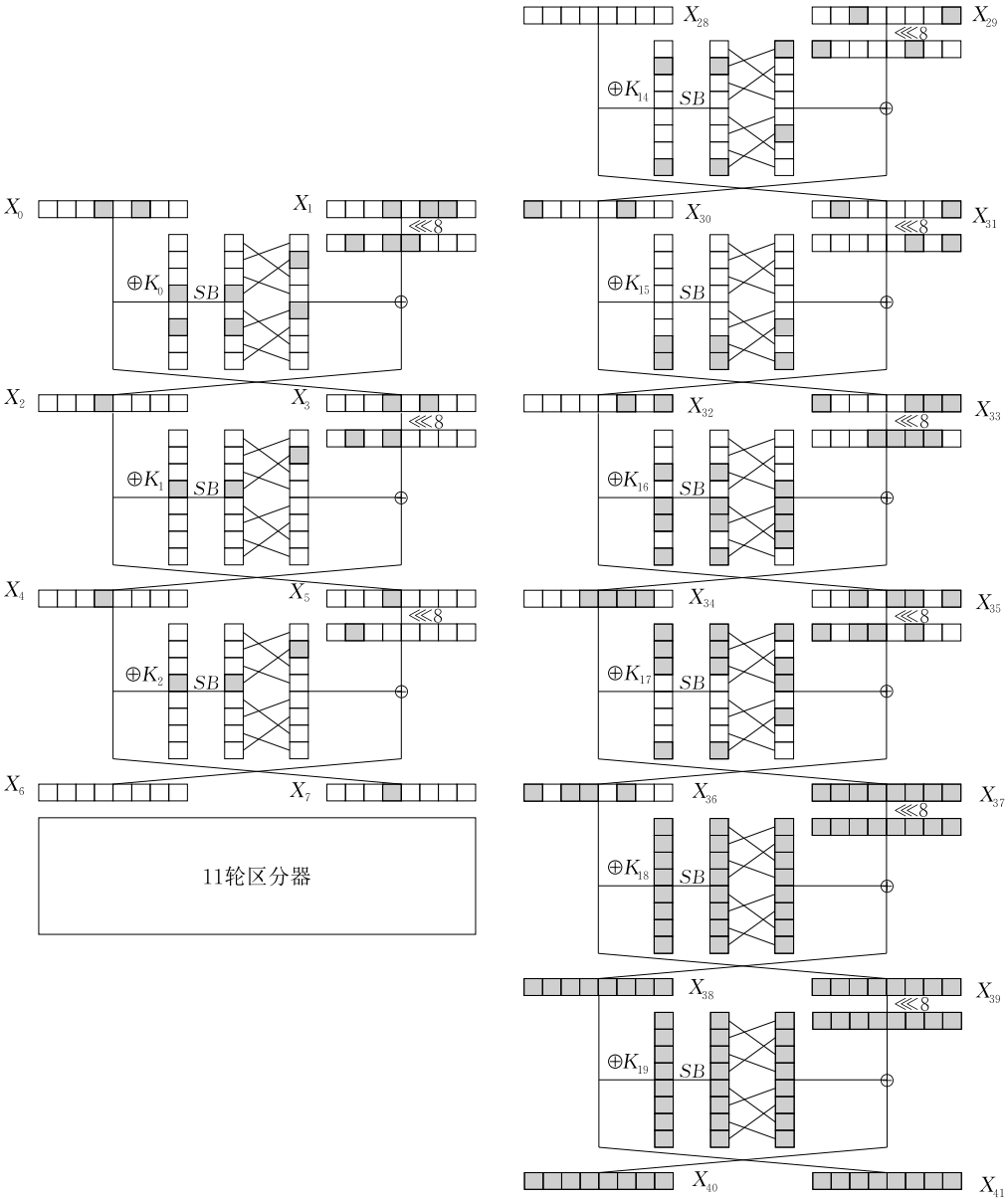

图 7　20 轮 LBlock* 算法的中间相遇攻击

当前初始密钥下各轮轮密钥可表示如下：

第 0 轮：1 1 1 1 0 1 0 0
第 1 轮：0 0 1 1 0 1 0 0
第 2 轮：0 0 1 1 1 0 0 0
第 3 轮：0 0 1 1 1 1 1 1
第 4 轮：1 1 0 0 1 1 1 1
第 5 轮：1 1 0 1 1 1 1 1
第 6 轮：1 1 0 1 1 1 1 1
第 7 轮：1 1 1 1 0 0 1 1
第 8 轮：1 1 1 1 0 1 1 1
第 9 轮：1 1 1 1 0 1 1 1
第 10 轮：1 1 1 1 1 1 0 0
第 11 轮：0 0 1 1 1 1 1 1
第 12 轮：1 1 1 1 1 1 1 1
第 13 轮：1 1 1 1 1 1 1 1
第 14 轮：1 1 0 1 1 1 1 1

第 15 轮：1 1 1 1 1 1 1 1
第 16 轮：1 1 1 1 1 1 1 1
第 17 轮：1 1 1 1 0 1 1 1
第 18 轮：1 1 1 1 1 1 1 1
第 19 轮：1 1 1 1 1 1 1 1

即 20 轮 LBlock* 算法预建表中间相遇攻击在线阶段涉及的 35 个密钥字节：

$K_0[0,2,3,5]$,$K_1[2,3]$,$K_2[3]$,
$K_{14}[1,7]$,$K_{15}[6,7]$,$K_{16}[2,4,5,7]$,
$K_{17}[0,1,2,7]$,$K_{18}[0\sim7]$,$K_{19}[0\sim7]$

均可由当前初始密钥通过密钥扩展算法得到.

攻击的具体步骤与 21 轮 LBlock 算法中间相遇攻击类似，此处不再赘述.

5.4 复杂度分析

离线阶段计算复杂度为 $2^4\times2^{76}$ 次部分加密，约为 $2^{76.9}$ 次 20 轮 LBlock 加密. 存储为 2^{76} 个 120 比特差分序列，约为 $2^{76.9}$ 个 64 比特块.

在线阶段计算复杂度为 $2^{76}\times2^4$ 次部分加解密，约为 $2^{77.7}$ 次 20 轮加密.

一次筛选后剩余密钥量为 $2^{76}\times2^{76}/2^{120}=2^{32}$.

在剩余密钥下穷搜 K_{14} 中$\{k_{71},k_{70},k_{69},k_{68}\}$所有可能值，即可以 2^{36} 的计算量恢复 K_{11} 的全部信息，由 K_{11} 与主密钥的等价性即可通过密钥扩展算法恢复主密钥.

综上所述，20 轮 LBlock* 算法预建表中间相遇攻击的计算复杂度为 $2^{76.9}+2^{77.7}+2^{36}\approx2^{78.3}$ 次 20 轮加密，存储为 $2^{76.9}$ 个 64 比特块，数据复杂度仅为 2^{20} 选择明文.

6 结束语

本文重新评估 LBlock 算法在中间相遇攻击下的安全性，在区分器穷搜的基础上充分利用密钥扩展算法排除冗余，成功将攻击轮数从现有的 19 轮增加至 21 轮，刷新了 LBlock 算法在中间相遇攻击下的安全性评估；与积分、多维零相关线性、不可能差分等其他接近全密码本 2^{80} 的数据复杂度相比，本文攻击具有显著的低数据复杂度，仅为 $2^{34.1}$，这对实际攻击环境下 LBlock 算法的安全性评估具有重要意义. 具体的分析结果对比如表 4 所示.

本文亦评估了新的密钥扩展算法下 LBlock 算法抵抗此类攻击的安全性，并得到了 20 轮优于穷举搜索的攻击结果，即新的密钥扩展算法有效增强了 LBlock 算法抵抗预建表中间相遇攻击的能力.

表 4 LBlock 算法分析结果对比

攻击类型	轮数	数据复杂度	计算复杂度	存储复杂度	文献
中间相遇攻击	17	2^{32}	$2^{55.5}$	—	[2]
中间相遇攻击	19	2^{38}	2^{65}	2^{64}	[3]
中间相遇攻击	21	$2^{34.1}$	$2^{75.8}$	$2^{74.8}$	本文
不可能差分分析	21	$2^{62.5}$	$2^{73.7}$	$2^{55.5}$	[4]
不可能差分分析	23	$2^{59.6}$	$2^{74.1}$	$2^{74.6}$	[5]
积分分析	23	2^{63}	2^{76}	2^{61}	[6]
多维零相关线性攻击	23	$2^{62.1}$	2^{76}	2^{60}	[7]
双系分析	32	2^{52}	$2^{78.4}$	2^4	[8]

致 谢 各位专家学者与编辑老师给本文提出了宝贵意见，在此表示感谢！

参 考 文 献

[1] Wu Wenling, Zhang Lei. LBlock: A lightweight block cipher// Proceedings of the Applied Cryptography and Network Security. Nerja, Spain, 2011: 327-344

[2] Zhang Lei, Shang Ya-Li, Sun Bo. A meet-in-the-middle attack on LBlock. Journal of University of Chinese Academy of Sciences, 2014, 31(4): 564-569(in Chinese)
(张磊，尚亚黎，孙勃. LBlock 的中间相遇攻击. 中国科学院大学学报，2014，31(4)：564-569)

[3] Wei Y, Su C, Ma C. A meet-in-the-middle attack on the LBlock cipher//Proceedings of the IEEE Conference Anthology. Shanghai, China, 2013: 1-3

[4] Liu Y, Gu D, Liu Z, Li W. Impossible differential attacks on reduced-round LBlock//Proceedings of the International Conference on Information Security Practice & Experience 2012. Guangzhou, China, 2012: 97-108

[5] Boura C, Naya-Plasencia M, Suder V. Scrutinizing and improving impossible differential attacks: Applications to CLEFIA, Camellia, LBlock and Simon//Proceedings of the ASIACRYPT 2014. Taiwan, China, 2014: 179-199

[6] Zhang H, Wu W. Structural evaluation for generalized feistel structures and applications to LBlock and TWINE// Proceedings of the INDOCRYPT 2015. Bangalore, India, 2015: 218-237

[7] Wang Y, Wu W. Improved multidimensional zero-correlation linear cryptanalysis and applications to LBlock and TWINE// Proceedings of the Information Security and Privacy. New South Wales, Australia, 2014: 1-16

[8] Wang Y, Wu W, Yu X, et al. Security on LBlock against biclique cryptanalysis//Proceedings of the Information Security Applications. Jeju Island, Korea, 2012: 1-14

[9] Diffie M, Hellman M. Exhaustive cryptanalysis of the NBS data encryption standard. Computer, 1977, 10(6): 74-84

[10] Demirci H, Selçuk A A. A meet-in-the-middle attack on 8-round AES//Proceedings of the Fast Software Encryption. Lausanne, Switzerland, 2008: 116-126

[11] Dunkelman O, Keller N, Shamir A. Improved single-key attacks on 8-round AES-192 and AES-256//Proceedings of the ASIACRYPT 2010. Singapore, 2010: 158-176

[12] Derbez P, Fouque P A, Jean J. Improved key recovery attacks on reduced-round AES in the single-key setting//Proceedings of the EUROCRYPT 2013. Athens, Greece, 2013: 371-387

[13] Li L, Jia K, Wang X. Improved single-key attacks on 9-round AES-192/256//Proceedings of the Fast Software Encryption. London, UK, 2014: 127-146

[14] Lin L, Wu W. Improved meet-in-the-middle distinguisher on feistelschemes//Proceedings of the Selected Areas in Cryptography 2015. Sackville, Canada, 2015: 122-142

[15] Dong X, Li L, Jia K, et al. Improved attacks on reduced-round Camellia-128/192/256//Proceedings of the CT-RSA 2015. San Francisco, USA, 2015: 59-83

[16] Biryukov A, Derbez P, Perrin L. Differential analysis and meet-in-the-middle attack against round-reduced TWINE//Proceedings of the Fast Software Encryption. Istanbul, Turkey, 2015: 3-27

[17] Kong Hai-Long, Wang Wei, Zhang Guo-Yan. Automatic search algorithm of meet in the middle attack on TWINE-128. Journal of Cryptologic Research, 2015, 2(6): 559-569(in Chinese)
(孔海龙，王薇，张国艳. TWINE-128 的中间相遇攻击的自动检测算法. 密码学报，2015，2(6)：559-569)

ZHENG Ya-Fei, born in 1988, Ph.D. Her research interests focus on block cipher.

WU Wen-Ling, born in 1966, Ph.D., professor, Ph.D. supervisor. Her research interests focus on symmetric cryptography.

Background

Meet-in-th'middle (MITM) attack was proposed by Diffie and Hellman in 1977 for the security analysis of TDEA (Triple Data Encryption Algorithm). During the past three decades, MITM attack has received worldwide attention and has been improved by a variety of techniques. General MITM attack can be divided into two classes, the MITM attack based on key partition and MITM attack based on distinguisher. The former includes the 3-subset MITM attack, biclique attack, etc. The latter includes truncated differential attack with MITM technique, MITM attack with precomputed table, and so on. MITM attack with precomputed table was proposed in the analysis of AES in the single key mode, and has been a hot topic of the cryptanalysis of block cipher in recent years. It has been significantly improved by multiset technique, differential enumeration technique, efficient tabulation, key bridging technique, and so on. At the same time, the security of many block ciphers, such as Camellia, CLEFIA, PRINCE, ARIA, against this attack has been evaluated, and most of the attacks are the currently best single key attacks.

LBlock cipher is a lightweight block cipher proposed in 2011 by Wu's team. Since the publicity of this cipher, its security has been researched under almost all traditional attacks, including differential attack, linear attack, integral attack, impossible differential attack. In 2013 and 2014, Wei Yong-Zhuang et al and Zhang Lei et al applied MITM attack on round reduced LBlock respectively. Their results give a preliminary evaluation of the security of LBlock cipher against MITM attack. In this paper, we focus on the security of LBlock under MITM attack with precomputed table. We select the best 11-round distinguisher and initial key comprehensively through programming, and a 21-round attack is proposed with low data requirement. Furthermore, we give another two groups of parameters, under which the same 21-round attacks can be achieved with the same time, memory, and data complexity. In conclusion, our attack extends the round number of MITM attack on LBlock cipher from 19 to 21, with a significantly low data complexity.

LBlock, like most other lightweight block ciphers, choose rather simple key schedule for the purpose of efficient hardware implementation, and this has led to lots of key-related attacks. In the biclique attack of LBlock, Wang Yan-Feng et al proposed a new key schedule for LBlock to speed up its diffusion. In this paper, the security of LBlock under the new key schedule is evaluated. The result shows that the new key schedule has a stronger diffusion property, and has successfully improved the security of LBlock under MITM-like attacks. This result may shine some light on the designation of key schedule of lightweight block cipher.

The research is supported by the National Basic Research Program (973 Program) of China (No. 2013CB338002), the National Natural Science Foundation of China (Nos. 61272476, 61672509, 61232009).