·安全技术· 文章编号：1000-3428(2019)10-0130-04 文献标志码：A 中图分类号：TP336

基于 LBlock 算法的密码 SoC 安全存储总线设计

张翌维，林 霖，赵 建，李发君，梁立新

（深圳技术大学 大数据与互联网学院，广东 深圳 518118）

摘 要： 密码片上系统（SoC）的数据访存通路是侵入式探针分析的重要目标，为抵御侵入式分析，利用 LBlock 算法设计一种 SoC 存储加密总线。将 LBlock 算法硬件结构每 4 轮展开为 1 个时钟周期，使 32 轮加解密时序压缩到 8 个时钟周期，同时将数据存储器一般采用的 32 位总线缓冲至 64 位，以配合 LBlock 算法的分组操作。FPGA 验证结果表明，该设计方案使得芯片内嵌数据存储器（如 RAM、Flash 等）的总线即使被探针攻击获取也无法解读，应用 64 位数据块进行 8 个时钟周期加密的访存吞吐率达到 533 kb/s，且避免了 32 位分组加密穷举攻击，实现代价低。

关键词： 密码片上系统；存储总线；总线加密；LBlock 算法；侵入式分析

开放科学（资源服务）标志码（OSID）：

中文引用格式： 张翌维，林霖，赵建，等. 基于 LBlock 算法的密码 SoC 安全存储总线设计[J]. 计算机工程，2019，45(10)：130-133.

英文引用格式： ZHANG Yiwei，LIN Lin，ZHAO Jian，et al. Design of secure memory bus for crypto SoC based on LBlock algorithm[J]. Computer Engineering，2019，45(10)：130-133.

Design of Secure Memory Bus for Crypto SoC Based on LBlock Algorithm

ZHANG Yiwei，LIN Lin，ZHAO Jian，LI Fajun，LIANG Lixin

（College of Big Data and Internet，Shenzhen Technology University，Shenzhen，Guangdong 518118，China）

【Abstract】 The data access path of the crypto System on Chip（SoC）is an important target of intrusive probe analysis. To resist intrusive analysis，an SoC memory encryption bus is designed by using the LBlock algorithm. Take every 4 rounds of LBlock algorithm hardware structure expansion as 1 clock cycle，so that the 32 bit encryption and decryption timing is compressed to 8 clock cycles，and the 32 bit bus generally used by the data memory is buffered to 64 bit to work with the packet operation of the LBlock algorithm. FPGA verification results show that the design scheme makes the bus of data memory embedded in a chip（RAM，Flash，etc.）can not be read even if it is acquired by the probe attack. The throughput rate of data access reaches 533 KB/s after 8 clock cycles of encryption using 64 bit data blocks，and exhaustive attacks against 32 bit block encryption can be avoided. The implementation cost is reduced.

【Key words】 crypto System on Chip（SoC）；memory bus；bus encryption；LBlock algorithm；invasive analysis

DOI： 10.19678/j.issn.1000-3428.0051952

0 概述

随着物联网产业市场的扩大，大量新兴的物联网应用走进人们的生活中，密码片上系统（System on Chip，SoC）的安全性问题越发凸显。密码 SoC 作为物联网安全的硬件基础设施，其应对非侵入、半侵入和侵入式分析的抵御能力，将直接影响物联网系统的物理层安全。在侵入式方向，密码 SoC 的安全性需要具备完善的抵御措施，应对安全传感器旁路、随机数发生器切割失效、数据总线探针等多种攻击手段[1-3]。相比于非侵入式和半侵入式分析，侵入式物理分析所需设备更加昂贵，例如原子力显微镜（AFM）、聚焦离子束（FIB）、研磨抛光设备等高端设备。此外，部分侵入式分析平台对安全分析人员的技术水平提出了更加综合和专业化的要求[1]，分析员通常需要具备化学、微电子、材料学以及数学等多

基金项目： 深圳技术大学新引进高端人才财政补助科研启动项目（2018010801008）；深圳技术大学校企产学研合作项目（2019106401005，2019106401006）；深圳技术大学教学改革研究项目（校教改 2018）。

作者简介： 张翌维（1980—），男，副教授、博士，主研方向为密码芯片 VLSI 设计、网络安全；林 霖，副教授；赵 建，助理教授；李发君，教授；梁立新，副教授。

收稿日期： 2018-06-28 **修回日期：** 2018-10-19 **E-mail：** changdavid@163.com

维度的知识[4-6]。

针对密码 SoC 片内存储器总线的探针分析,是侵入式分析(攻击)中最直接获取敏感信息(如密钥、关键数据)的方法。攻击者首先除去 SoC 的封装,然后将芯片敏感信号引出至芯片表面,并通过预留焊盘打通实现与外界的信号连接,从而使用微探针获取感兴趣的信号,监听芯片内存储器访存时收发的敏感信息。

如果存储器总线布线在芯片顶层,攻击者可直接采用微探针探测存储器的数据总线节点,监听并获取芯片所保护的敏感信息;如果采用具备表层主动防护层的安全布线,布置在下层,攻击者通常会使用 FIB 离子束对芯片进行打孔、切割、连接和引出,旁路主动防护层传感器使其失效[3],进一步采用微探针进行监听探测[7-9]。此外,攻击者还可借助电子显微镜、AFM 等高倍成像显微镜研究存储器的介质阵列,对存储内容进行解读。

针对上述侵入式攻击手段,密码 SoC 需要对片上数据存储总线进行加密保护,例如对 RAM、EEPROM、Flash 等所存储的信息进行加密处理,保障存储器的访存端口信号为加密信号,而明文信号节点打散布局布线,隐藏在大规模设计中,从而确保存储器介质内容即便被攻击者观测,也无法对其进行解读。本文提出一种基于 LBlock 算法的密码 SoC 存储加密总线,将 LBlock 算法轮次在时序上进行压缩,以降低时序代价,提高访存效率。

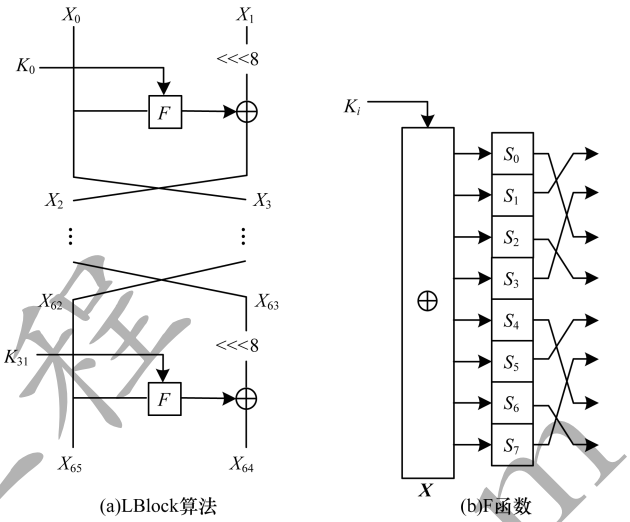
1　方案设计

为抵御侵入式的物理探测针对存储器周边敏感信号线的探测,需要保障存储器直连的周边节点或连线为加密后的信息,而加密前的明文信息节点则乱序、打散布局在芯片的各个金属层,并且不出现在顶层,根据当前主流的 90 nm 以下的工艺密度,从数以亿计的打散信号中准确探测到加密前的明文电路节点并不现实。这样保障了存储介质被显微镜观测分析不可解读,并且无法通过 FIB 和微探针探测存储器周边节点,无法监听及获取明文数据。

加密算法选择受 80 位密钥控制、64 位输入输出的 LBlock 算法,一方面该算法结构简单,关键路径短,利于通过电路展开压缩执行时间,提升访存吞吐率;另一方面,该算法强度经过长期论证[10-13],加密分组为 64 位,降低了 32 位总线直接加密被穷举的风险。

1.1　LBlock 算法结构

LBlock 算法[14]属于轻量级分组密码算法,受到国际密码学界的广泛关注,具有较高的软硬件实现效率。其分组长度为 64 位,密钥长度为 80 位,采用 32 轮变体 Feistel 结构。图 1(a)为算法整体流程,算法将 64 位明文分成左右两部各 32 位,进行32 轮运算后仍得出左右两部各 32 位,共 64 位密文输出。图 1(b)给出图 1(a)中 F 函数的算法描述,32 位输入与子密钥 K_i 异或后,每 4 位一组进入 8 个4 进4 出的 S 盒,输出后需按图 1(b)中的排序进行换位。



(a)LBlock算法　　(b)F函数

图 1　LBlock 算法

密钥扩展将 80 位主密钥取高 32 位作为轮密钥 $K_i(i=0,1,\cdots,31)$,每取一次轮密钥按以下步骤更新 K_i:

1)$[k_{79}k_{78}\cdots k_0]=[k_{50}k_{49}\cdots k_0k_{79}k_{78}\cdots k_{51}]$;

2)$[k_{79}k_{78}k_{77}k_{76}]=S_9[k_{79}k_{78}k_{77}k_{76}]$,$[k_{75}k_{74}k_{73}k_{72}]=S_8[k_{75}k_{74}k_{73}k_{72}]$;

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

其中,S_8、S_9 均为 4 比特 S 盒。LBlock 算法所涉及 S 盒标准参照 LBlock 算法标准,皆为 4 比特类型。

解密仍采用图 1 的操作,仅需将明文替换为密文,子密钥出现次序改为逆序 $K_i(i=31,30,\cdots,0)$。

1.2　算法实现结构

加密电路的数据路径如图 2 所示。当加密轮次为首轮($i=0$)时,图 2 中两个二选一 MUX 电路的"0"端选通,X_0 与 X_1 两个 32 位明文信号输入至电路,分别得到 X_2 与 X_3;此后的轮次二选一 MUX 的"1"端选通,第 i 轮($i=1,2,\cdots,31$)运算得到 X_{2i+2}与 X_{2i+3},第 32 轮($i=31$)输出 X_{65} 与 X_{64}为 64 位密文$[X_{65}\parallel X_{64}]$。

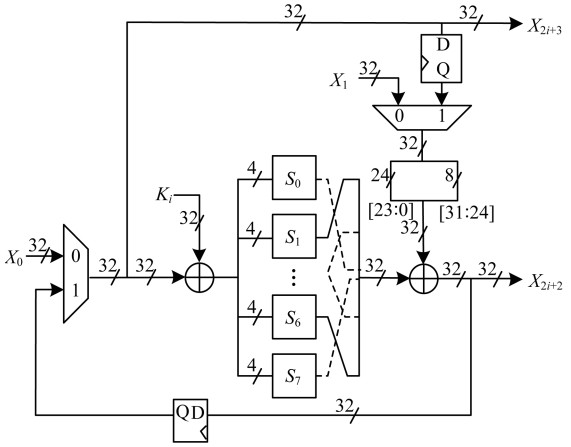


图 2　加密电路的数据路径设计

图 3 为密钥扩展电路的数据路径设计。当加密轮次为首轮($i=0$)时,二选一 MUX 电路的“0”端选通,同时直接取主密钥的高 32 位作为首轮子密钥 K_0;此后轮次二选一 MUX 电路的“1”端选通,第 i 轮($i=1,2,\cdots,31$)密钥扩展得到子密钥 K_i。

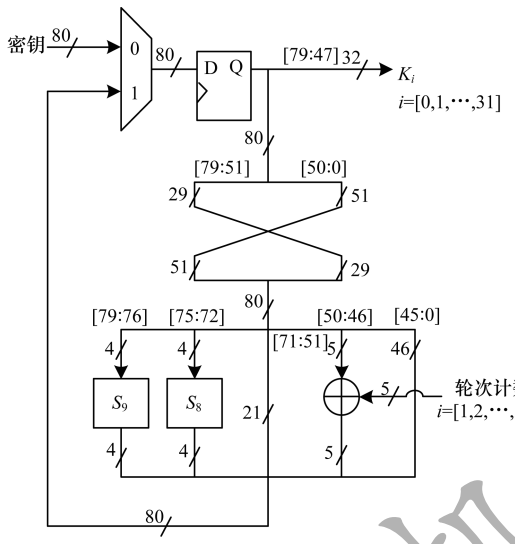


图 3 密钥扩展电路的数据路径设计

除密钥的安全存储外,密码总线中所有电路不应借助片上易失或非易失存储空间存储,因为这些空间本身也需受到该密码总线的加密存储保护。

加密密钥应在非易失性存储体中受保护的空间内存储,在芯片生产或发行时写入,只允许密码总线以硬件连接方式载入,CPU 等其他电路不可访问。

解密电路的数据路径与加密电路一致,仅将输入变为密文。解密时通常需要将密钥扩展的每一轮子密钥寄存然后逆序使用,这并不适用于密码总线,因为需占用的存储空间(通常为 RAM 空间)本身也应受该加密算法保护。本文方案解密时可将 K_{31} 所在的最后一轮 80 位密钥,在生产或发行时存储在非易失存储体中受保护的空间,然后每一轮进行密钥扩展的逆运算,如此可得 $K_{30},K_{29},\cdots,K_0$。逆扩展中 4 比特S 盒应取 S_8、S_9 的逆 S_8^{-1}、S_9^{-1}。如此,密钥扩展在加密与解密为两个不同电路,且时序完全相同。4 比特 S 盒的实现可直接采用真值表的组合逻辑描述,利用综合工具的逻辑优化能力,得到最优关键路径的组合逻辑电路。相比于 DES 算法的 6 比特 S 盒和 AES 算法 8 比特 S 盒,LBlock 算法普遍采用 4 比特 S 盒,其关键路径大幅缩短,且在算法中配置更加灵活,面积更加紧凑。

1.3 密码总线 VLSI 设计

基于 LBlock 算法的 VLSI 结构的优点是数据路径非常短,结构简练且具有较好的密码强度。但在时序上每次加密或解密需要进行 32 轮,也即完成一个分组加解密需要 32 个时钟周期,这对于存储读写总线的实时性和吞吐率会带来较大的影响。所以,可将密码 VLSI 结构进行展开,每个时钟完成 4 个 ~ 8 个轮次,加快密码总线读写性能。图 4 给出单时钟 4 轮次展开电路结构,该结构下一个时钟周期可完成 4 轮加密,共 8 个时钟周期完成加密,其中加解密第一个时钟周期中二选一 MUX 电路“0”端选通,其余时钟周期“1”端选通。

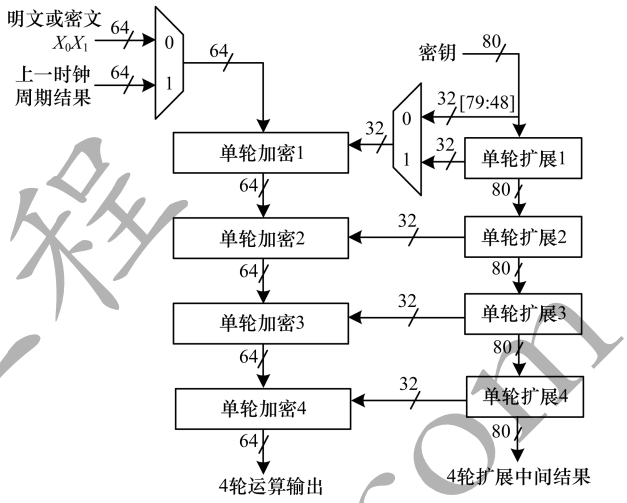


图 4 4 时钟时序展开电路结构

相比于 32 轮标准实现,4 时钟时序展开电路信号输入设计只需少量调整,单轮加密 1 模块、单轮扩展 1 模块参照图 2 和图 3,分别取消二选一 MUX 和寄存器(DQ 触发器),然后将单轮输出分别引向下一轮单轮加密 2 模块、单轮扩展 2 模块,以此类推,时序展开设计非常便捷。如此每个时钟周期得到 4 轮运算结果和密钥扩展结果,8 个时钟完成加解密。

本文设计采用 4 轮叠加的时序展开结构,关键路径延迟会相应增加,这主要考虑到 LBlock 算法为轻量级算法,其单轮次关键路径延迟低于 1 ns。4 时序展开后,由于综合工具的逻辑优化能力,总延迟仅为 3 ns 左右,在 100 MHz 时钟频率下具有足够的时序裕量,满足当前密码 SoC 的主流频率约束。

2 验证结果与性能分析

总线位宽一般为 8 位、16 位或者 32 位,如选择 32 位或以下分组的加密算法,因密码本空间不足,会遭受直接穷举攻击。LBlock 密码算法分组长度为 64 位,在总线挂接设计时,需设计缓冲 Buffer 机制将总线单拍信息(8 位、16 位或 32 位)填充至 64 位再加密,无法凑整时应填充随机数凑整。

在 Manis 测试验证平台上进行设计验证,FPGA 型号为 Spartan-3E xc3s1400A,采用 Verilog 硬件描述语言实现本文所述的 4 时钟时序展开密码总线,总线位宽 32 位,在 50 MHz 时钟频率下,连续地址加密写和解密读数据吞吐率达到 266 Mb/s。FPGA 平台环境如图 5 所示。

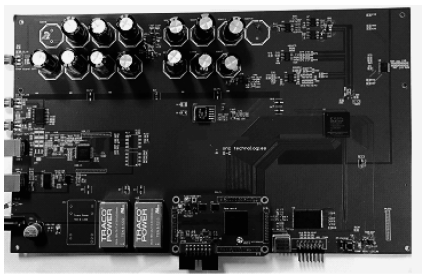


图 5　FPGA 验证平台

图 6 为向地址 0x0000 的存储空间写入 64 位数据 0x0123456789abcdef 的过程,Round 寄存器用于 8 时钟计数,密钥也同为 0x0123456789abcdef。通过对访问缓冲区 BUF_L 和 BUF_H 的控制,得到写入密文为 0x4b7179d8ebee0c26,批量写入可进行依次流水操作。读出时使用时序相同的解密电路对存储密文进行解密。

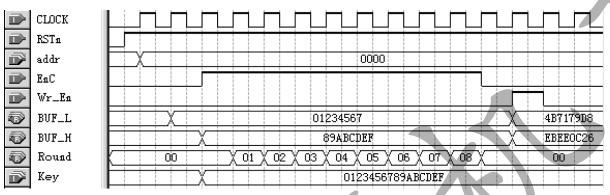


图 6　密码存储总线设计时序

表 1 给出本文设计方案与其他轻量级算法实现性能对比[14-15],标准 32 轮 LBlock 实现在较低门级下具备较好性能,相比之下本文方案采用 4 时序展开技术,在门级增加约一倍的情况下,将连续访存性能再提高近 2.7 倍,时钟频率为 100 KHz 时峰值速率达到 533 kb/s,这说明本文方案吞吐率性能更加匹配 SoC 存储总线的性能需求。此外,本文方案采用 FPGA 实现,门级为预估所得,需指出的是,在当前流行的 40 nm ~ 90 nm 工艺段,2 000 门级以下的代价增加成本低于 ¢1。

表 1　各轻量级算法实现性能对比

算法	分组长度/bit	密钥长度/bit	门级	吞吐率/(kb·s^{-1})	工艺段/μm
DES 算法[14]	64	56	2 300	44.4	0.18
KATAN 算法[14]	64	80	1 054	25.1	0.13
LBlock 标准算法[14]	64	80	1 320	200.0	0.18
PRESENT 算法[15]	64	80	1 570	200.0	0.18
本文设计方案	64	80	2 986(预估)	533.0	FPGA

3　结束语

本文设计一种基于 LBlock 算法的密码 SoC 存储器加密总线。将 LBlock 算法轮次在时域空间进行4 时序展开,降低时序代价。FPGA 验证结果表明,本文方案在保持算法 32 轮密码强度的基础上,大幅提高了访存吞吐率,且实现代价低,更加匹配 SoC 系统对密码总线的性能和安全性需求。

参考文献

[1] TORRANCE R, JAMES D. The state-of-the-art in IC reverse engineering [C]//Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin, Germany: Springer, 2009: 363-381.

[2] SAMYDE D, SKOROBOGATOV S, ANDERSON R, et al. On a new way to read data from memory [C]// Proceedings of the 1st International IEEE Security in Storage Workshop. Washington D. C., USA: IEEE Press, 2002: 65-69.

[3] CIORANESCO J M, DANGER J L, GRABA T, et al. Cryptographically secure shields [C]//Proceedings of 2014 IEEE International Symposium on Hardware-oriented Security and Trust. Washington D. C., USA: IEEE Press, 2014: 25-31.

[4] WEINGART S H. Physical security devices for computer subsystems: a survey of attacks and defenses[C]//Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin, Germany: Springer, 2000: 302-317.

[5] SKOROBOGATOV S. Physical attacks and tamper resistance[M]. Berlin, Germany: Springer, 2012.

[6] 冉彤,白国强. 基于系统级封装(SiP)的信息安全芯片集成设计[J]. 微电子学与计算机, 2012, 29(1): 10-14.

[7] SHAHRJERDI D, RAJENDRAN J, GARG S, et al. Shielding and securing integrated circuits with sensors [C]// Proceedings of 2014 IEEE/ACM International Conference on Computer-Aided Design. Washington D. C., USA: IEEE Press, 2014: 170-174.

[8] BRIAIS S, CIORANESCO J M, DANGER J L, et al. Random active shield[C]//Proceedings of 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography. Washington D. C., USA: IEEE Press, 2012: 103-113.

[9] BRIAIS S, CARON S, CIORANESCO J M, et al. 3D hardware canaries [C]//Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin, Germany: Springer, 2012: 1-22.

[10] 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.

[11] ZHANG Huiling, WU Wenling. Structural evaluation for generalized feistel structures and applications to LBlock and TWINE [C]//Proceedings of the INDOCRYPT'15. Berlin, Germany: Springer, 2015: 218-237.

[12] BOURA C, NAYA-PLASENCIA M, SUDER V. Scrutinizing and improving impossible differential attacks: applications to CLEFIA, Camellia, LBlock and Simon[C]//Proceedings of ASIACRYPT'14. Berlin, Germany: Springer, 2014: 179-199.

[13] 郑雅菲,吴文玲. LBlock 算法的改进中间相遇攻击[J]. 计算机学报, 2017, 40(5): 1080-1091.

[14] WU Wenling, ZHANG Lei. LBlock: a lightweight block cipher [C]//Proceedings of International Conference on Applied Cryptography and Network Security. Berlin, Germany: Springer, 2011: 327-344.

[15] BOGDANOV A, KNUDSEN L R, LEANDER G, et al. PRESENT: an ultra-lightweight block cipher [C]// Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin, Germany: Springer, 2007: 450-466.

编辑　陆燕菲