

可重构 S 盒替换单元研究与设计

常忠祥, 陈 卓

(国防科技大学 信息通信学院, 湖北 武汉 430010)

摘 要: 针对微型终端资源受限导致加密算法单一的问题, 对分组密码中的关键部件 S 盒进行可重构设计, 提出以 8—1 S 盒为基本可重构单元, 将 S 盒转换位布尔函数表达式, 采用变量分组、递进计算的方式, 大幅减少表达式中的与项个数, 提升中间结果的利用率. 在此基础上, 设计了一种可重构 S 盒单元, 并在 TSMC 45nm CMOS 工艺下进行综合, 工作频率可达 1.67GHz, 与现有研究成果相比, 本设计不仅能够很好的满足当前微处理器的速度需求, 且资源占用仅为同类设计的 2/3.

关键词: 分组密码; S 盒; 可重构; 变量分组; 递进计算;

中图分类号: TP302.2

文献标识码: A

文章编号: 1000—7180(2018)12—0125—04

A Reconfigurable S—box Unit Design for Micro Terminal

CHANG Zhong-ixang, CHEN Zuo

(College of Information and Communication, National University of Defense Technology, Wuhan 430010, China)

Abstract: Based on the limited resources of tiny terminals lead to the problem of single encryption algorithm, the S—box as the key components of the block cipher is reconfigurable designed. This paper put forward an 8—1 S box as the basic reconfigurable unit, converts S box of Boolean function expressions, with the method of grouping variable, and recursive calculation, dramatically reduced expression of and — item number, increase the utilization of intermediate results. On this basis, we design a reconfigurable unit S box, and under the TSMC 45 nm CMOS technology synthetically. The working frequency can reach 1.67 Ghz, compared with the existing research results, this design can well satisfy the current demand of the speed of the microprocessor. At the same time, the resource usage is only two—thirds of similar design.

Key words: Block cipher; S box; Reconfigurable; Variables Grouping; Progressive calculation;

1 引言

随着电子支付、即时通讯和电子货币等应用领域的快速发展, 手持设备、可穿戴设备等小型、微型智能移动系统终端呈爆炸式增长^[1]. 但由于小型、微型智能移动系统终端资源受限的问题, 通常采用固定算法加密的方法, 导致数据存在极大的潜在威胁^[2]. 亟需一种能够支持多个密码算法重构实现的方式. 而 S 盒作为密码算法的关键部件, 其实现方式和效率直接决定了分组类密码算法是否能够在资源受限终端重构实现.

S 盒通常有基于 LUT^[3]、基于逻辑电路^[4]和布

尔函数^[5]等方式, 其中基于 LUT 方式以 ROM 为基础, 虽然设计简单, 但需要消耗大量的存储资源, 很难满足资源受限的情况; 基于逻辑电路是根据 S 中存储数据的计算过程, 用运算部件逻辑电路实时计算, 电路规模和路径延迟都是难以接受的, 虽然能够通过特定的手段缩减逻辑电路的规模, 但其计算过程会更加复杂, 路径延迟和更长; 基于布尔函数的实现方式, 是将 S 盒替代等价于布尔函数运算, 随着 S 盒输入数据的增多, 则布尔函数的变量也增多, 其异或项个数成指数级增长, 硬件资源是难以接受的, 需要采用递进的计算方法, 完成结果的计算, 其递进的计算方式决定了资源消耗和处理效率, 如果能够找

到合理的递进计算方式,是一种较好的选择.

本文通过分析不同类型 S 盒的特点,分析不同类型 S 盒之间的转换关系,提出以 8-1 的 S 盒为可重构 S 单元的实现基础,将其等价为 8 变量的布尔函数,通过变量分组,将该函数转换为多组 4 变量的布尔函数和四变量的与项运算,极大的提升 8 变量布尔函数的并行计算能力,降低硬件资源的消耗,同时,充分利用递进计算的中间结果,完成 4-4、6-4S 盒的计算,并以此为基础,设计了灵活、高效的可重构 S 盒单元.

2 S 盒基本可重构单元的选取

对常见的 DES, AES, E2, MARS, SERPENT, Twofish 等几十种包含 S 盒替代操作的分组密码算法进行研究和分析,分组密码算法中 S 盒替代操作存在以下特点:

(1) S 盒种类主要以 8-8 为主,如 AES, Twofish, E2, SAFER+ 等多个算法;8-32, 8-4, 6-4 等次之,如 DES, MARS, CAST-256, GOST 等少量算法;极个别算法也包含 13-8, 11-8 等一些特殊的替换类型,如 LOKI97.

(2) 算法均需要经过多轮运算,且每轮中包含多个 S 盒并行运算. 如 DES 分组长度 64 bit, 每轮 16 个 6-4 的 S 盒,需运算 16 轮,共需 256 个 S 盒.

(3) 大部分算法中每一轮操作使用的 S 盒完全相同,也有极个别算法不同,如 Serpent 算法,第 1~8 轮使用 S 盒各不相同.

通过以上分析可以发现,8-8 的 S 盒是当前算法中最常用的运算类型. 同时,由 S 盒的替代原理可知,不同类型间的 S 盒是可以相互转换的实现的. 其中 6-4 和 4-4 是 8-8 的子集,直接可以利用 8-8 的部分资源来实现,会造成部分资源的闲置. 而 8-8 是 8-32、9-8、11-8、13-8 的子集,必须通过多个 8-8 组合实现.

$$S = \{S_1, S_2, S_3, S_4\} \quad (1)$$

$$S_1(x_7x_6x_5x_4x_3x_2x_1x_0) = a_7a_6a_5a_4a_3a_2a_1a_0 \quad (2)$$

$$S_2(x_7x_6x_5x_4x_3x_2x_1x_0) = b_7b_6b_5b_4b_3b_2b_1b_0 \quad (3)$$

$$S_3(x_7x_6x_5x_4x_3x_2x_1x_0) = c_7c_6c_5c_4c_3c_2c_1c_0 \quad (4)$$

$$S_4(x_7x_6x_5x_4x_3x_2x_1x_0) = d_7d_6d_5d_4d_3d_2d_1d_0 \quad (5)$$

由式(1)可知,8-32 S 盒等价于将四个 8-8 的 S 盒的结果并置输出,由式(2)(3)(4)(5)可知,每一

个 8-8 的 S 盒操作完全相同, S 盒的内容相互独立,可以相同,也可以不同.

$$S = \{x_m \cdots x_9 x_8 S_{2^{m-8}}, \cdots, \overline{x_m} \cdots \overline{x_9} \overline{x_8} S_1\} \quad (6)$$

$$S_1(x_7x_6x_5x_4x_3x_2x_1x_0) = a_7a_6a_5a_4a_3a_2a_1a_0$$

...

$$S_{2^{m-8}}(x_7x_6x_5x_4x_3x_2x_1x_0) = v_7v_6v_5v_4v_3v_2v_1v_0 \quad (7)$$

由式 6 可知, $m-8(m>8)$ 的 S 盒等价于利用 m 位输入中的任何 8 bit 数据计算出 2^{m-8} 个结果,然后通过 $x_m \cdots x_9 x_8$ 的与项确定 2^{m-8} 个结果中唯一的输出结果. 即随着 m 的增大,所需的 8-8 运算成指数级增长.

同理,也可以采用 4-4 位基本运算部件,通过叠加完成 $m-4, m-8, m-32(m>4)$ 的 S 盒. 若 4-4 实现 8-8 的 S 盒, 4-4 的结果空间为 $2^4 \times 4 = 64$ bit, 而 8-8 的结果空间为 $2^8 \times 8 = 2048$ bit, 需要 32 个 4-4 才能扩展完成, 与 8-8 位基本运算部件相比, 虽然提升了灵活性和资源利用率, 但 4-4 由于加密强度不够, 仅在极个别算法中才有. 而 8-8 在密码算法中的出现频率最高, 且能够较快的组合完成目前常见的 8-32、11-8、13-8 类型的 S 盒, 因此本文选择以 8-8 为的实现为基础, 研究 S 盒可重构设计技术.

3 S 盒可重构设计原理

S 盒替代运算在本质上等价于非线性布尔函数的运算, S 盒的输入就是布尔函数的自变量, 输出就是函数值. 则 8-8 S 盒可以拆分为 8 个 8-1 的 S 盒, 则 8-1S 盒可表示为:

$$f(x) = \sum_{i=1}^{2^8-1} c_i x_7^{i_7} x_6^{i_6} x_5^{i_5} x_4^{i_4} x_3^{i_3} x_2^{i_2} x_1^{i_1} x_0^{i_0} \quad (8)$$

式中, $c_i \in \{0, 1\}$, $i_7 \in \{0, 1\}, \cdots, i_0 \in \{0, 1\}$, $x_7^0, \cdots, x_0^0$ 分别表示 $x_7, \cdots, x_0$ 取反; $x_7^1, \cdots, x_0^1$ 分别表示 $x_7, \cdots, x_0$ 自身.

式 8 也可以看作两个 4 变量的布尔函数乘积

$$\begin{aligned} f(x) &= \sum_{i=0}^{2^8-1} c_i x_7^{i_7} x_6^{i_6} x_5^{i_5} x_4^{i_4} x_3^{i_3} x_2^{i_2} x_1^{i_1} x_0^{i_0} \\ &= \left(\sum_{i=0}^{2^4-1} c_i x_7^{i_7} x_6^{i_6} x_5^{i_5} x_4^{i_4} \right) \left(\sum_{i=0}^{2^4-1} d_i x_{i_{33}} x_{i_{22}} x_{i_{11}} x_{i_{00}} \right) \end{aligned} \quad (9)$$

对于密码算法中的 S 盒来说, 通常方式是将输入变量分为行地址和列地址, 即一个行地址可以对应多个列地址, 一个列地址对应多个行地址. 如式 9 中, 若 (x_7, x_6, x_5, x_4) 对应行地址, (x_3, x_2, x_1, x_0)

对应列地址.但由于 S 盒输出结果相对固定,且不同的行地址,对应的列地址也不相同,很难提取出式 9 的形式,但对中间计算过程分析可以发现,单一行地址可以对应多个列地址,如 AES 算法中,每一个行地址都要对应 16 个列地址.则(9)可以变换为.

$$f(x) = \sum_{i=0}^{2^8-1} c_i x_7^i x_6^i x_5^i x_4^i x_3^i x_2^i x_1^i x_0^i$$

$$= a_0 x_7^i x_6^i x_5^i x_4^i \left(\sum_{i=0}^{2^4-1} d_i x_3^i x_2^i x_1^i x_0^i \right) \oplus \dots$$

$$\oplus a_{15} x_7^i x_6^i x_5^i x_4^i \left(\sum_{i=0}^{2^4-1} d_i x_3^i x_2^i x_1^i x_0^i \right) \quad (10)$$

由式(10)可知, $f(x)$ 可以拆分为 16 个四变量 (x_3, x_2, x_1, x_0) 布尔函数与对应的四变量与项进行与运算,结果进行异或运算,可以得出 8-1 的 S 盒.实现不同 8-1 的 S 盒,仅需要改变 a_i 和 d_i 的取值即可.而目前算法中常见的 S 盒 a_i 是必然存在的,因此,仅需要改变 d_i 的值即可,则完成一个 8-1 的 S 盒需要 $16 \times 16 \text{ bit}$ 的控制信息.同时,在式(10)中可以看出, $(x_7, x_6, x_5, x_4), (x_3, x_2, x_1, x_0)$ 两组变量结果相互独立,可以并行计算,能够在很大程度上减少关键路径的长度.而实现 8-8 的 S 盒时,仅需将 8 个 8-1 S 盒输出结果并置即可.

完成 4-1 的 S 盒操作时,其等价的布尔函数表达式如式(11)所示:

$$f(x) = \sum_{i=0}^{2^4-1} c_i x_3^i x_2^i x_1^i x_0^i \quad (11)$$

与式(10)相比,式(11)个仅相当于其括号中计算内容,因此,当括号中的布尔函数计算完成后,直接输出即为 4-1 的结果,同时,8-1 一共包含 16 个该种类性的运算,则 8-1 的计算过程中可以输出 16 个 4-1 的计算结果,等价于支持 4 个 4-4 的 S 盒运算.

完成 6-1 的 S 盒时,其等价的布尔函数表达式如式(12)所示:

$$f(x) = \sum_{i=0}^{2^8-1} c_i x_7^i x_6^i x_5^i x_4^i x_3^i x_2^i x_1^i x_0^i$$

$$= x_7^i x_6^i (a_0 x_{15} x_4^i \left(\sum_{i=0}^{2^4-1} d_i x_{13} x_2^i x_1^i x_0^i \right) \oplus \dots$$

$$\oplus a_3 x_{15} x_4^i \left(\sum_{i=0}^{2^4-1} d_i x_{13} x_2^i x_1^i x_0^i \right)) \oplus \dots \oplus$$

$$x_7^i x_6^i (a_{12} x_{15} x_4^i \left(\sum_{i=0}^{2^4-1} d_i x_{13} x_2^i x_1^i x_0^i \right) \oplus \dots$$

$$\oplus a_3 x_{15} x_4^i \left(\sum_{i=0}^{2^4-1} d_i x_{13} x_2^i x_1^i x_0^i \right)) \quad (12)$$

由式(12)可知,6-1 的 S 盒仅有 $(x_5, x_4, x_3, x_2, x_1, x_0)$ 参与,则 8-1 S 盒计算的中间结果包含 4 个 6-1 的输出结果,且每一个 6-1 S 盒的输出结果分别与 (x_7, x_6) 共同参与后续运算,可以将 (x_7, x_6) 未参与前的结果直接输出,完成 6-4 S 盒运算.

由上分析可知,利用变量分组、递进计算的方法,能够灵活的对 8-1 S 盒的进行可重构设计,提升并行计算能力,提高计算效率,大幅缩减电路的规模,减少硬件消耗,是一种较好的选择.

4 可重构 S 盒设计

由第二章分析可知,以式(12)为依据,设计 8-1 S 盒的可重构架构,能够灵活的完成 4-4、6-4 的 S 盒运算,则可重构 S 盒单元设计如图 1 所示.

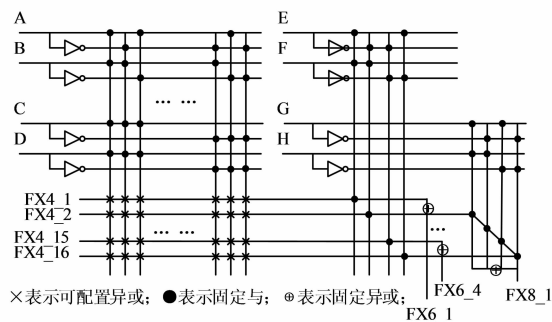


图1 可重构 8-1 S 盒架构

如图 1 所示,包含三个与阵列,输入分别为 $(ABCD), (EF)$ 和 (GH) ,分别完成 $(ABCD), (EF)$ 和 (GH) 内部所有最小与项的组合. 1 个可异或阵列,负责完成输入为 $ABCD$ 的四变量的布尔函数运算,其直接输出结果为 4-1 的结果,共 16 bit 数据,支持 4 个 4-4 的运算. 然后与 (EF) 形成的最小与项完成与运算,依次将其分为四组,每组内数据进行异或,输出 6-1 S 盒的结果,共 4 bit 数据. 最,支持 1 个 6-4 的运算,最后 6-1 S 盒的结果与 (EF) 形成的最小与项完成与运算,并将所有结果进行异或,支持 1 个 8-1 的运算. 在计算过程中,仅可配置异或阵列需要配置信息,共需 $16 \times 16 = 256 \text{ bit}$ 配置信息.

同理,要实现 8-8 的 S 盒,变量固定时,其可组成的最小项个数也是固定的,则图 1 中的与阵列可以重复利用,仅需要将除与阵列意外的所有计算阵列复制,输出结果进行并置即可. 但同样,电路所需的配置信息也增加了八倍,共 256 byte.

由图 1 和图 2 可以知道,实现不同类型的 S 盒,其差别仅在于架构中异或阵列的大小,因此,用户可

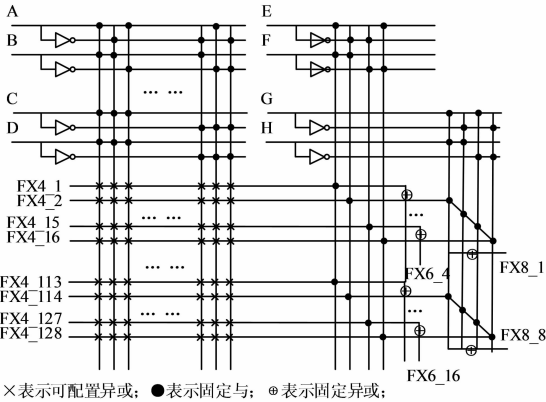


图2 可重构8—8S盒架构

以根据实际应用微型终端上能够支持的资源大小，来确定异或阵列的大小，异或阵列的大小和速度成正比，和配置信息量成正比，和灵活性成反比。

5 功能与性能评估

为了客观的评估本文设计的可重构S盒硬件架构，分别从功能和性能两方面进行评估。本文选取单次运算支持32个4—4S盒，8个6—4S盒，1个8—8S盒为对象，在TSMC 45nm CMOS工艺下进行综合和评估。

(1)功能评估

与传统的LUT实现相比，在支持相同功能前提下，LUT实现最简单，但需要占用内存的资源，尤其是当算法中S盒中类较多时，需要占用大量的内存资源存储S盒内容，这在资源受限的终端上是难以接受的。本设计虽然增加了少量的配置资源，但不会占用内存的空间，即使需要不同的S盒，仅需对配置寄存器操作即可，虽然性能有所降低，但资源约束较小。

与逻辑电路实现相比，逻辑电路是由S盒内容的生成方式决定的，针对性很强，如AES中S盒的计算方式只适用于AES，且计算环节非常复杂。而本设计是不关心S内容的具体生成方式，只关心输入和输出的对应关系，灵活性较大，且能够原来复杂的计算过程等价到布尔函数的运算中去，降低计算难度，提升计算速度。

(2)性能评估

本文分别选取文献[7]中基于ARM的LUT实现方式和文献[8]中与本文相似的实现方式在工作频率、资源占用、AES中S盒实现指令条数、配置信息等方面进行对比。

表1 性能对比

功能单元	工作频率(GHz)	资源(nand)	运算次数AES
文献[7]	1.3	—	20
文献[8]	1.45	18k	13
本设计	1.67	12k	13

在相同工艺、支持相同功能的前提下，本设计的工作频率可达1.67GHz，明显高于文献[7]中ARM的1.3GHz和文献[8]的1.45GHz，能够满足处理器对频率的需求。而资源方面，ARM占用的是内存资源，因此无法与本设计进行客观对比，文献[8]与本设计类同，但文献[8]直接采用的是8变量形成的最小项，实现6—4、4—4时，需要将其转换为8变量的形式，虽然也能实现，不利用应用，重构能力差。同时，在支持相同功能的前提下，虽然关键路径上都是1个非门、2个与门、8个异或门，但文献[8]中与阵列是本设计的两倍，综合过程中会增加大量的缓冲器，来保证信号的强度，导致延迟比本设计大，资源占用是本设计的1.5倍。

6 结束语

本文首先分析了分组密码算法中S盒的类型以及相互之间的对应转换关系，并利用布尔函数表达式来表示S盒输入与输出的对应关系，通过将变量进行分组递进的方式完成S盒的计算，与相似的设计方式相比，与阵列能够减少一半，资源占用减少1/3，工作频率可达1.67GHz，完成能够满足当前处理器的需求。下一步，是将目前所有分组密码算法中的S盒的表达式进行统计，寻求各种不同S盒的布尔函数表达式中与项之间的规律，进一步降低异或阵列的规模，降低配置信息的量，从而降低硬件资源的消耗。

参考文献：

[1] 金晨辉,郑浩然,张少武,等. 密码学[M]. 高等教育出版社, 2009.

[2] 戴乐育. 可重构分组密码协处理器二维指令系统研究与设计[D]. 郑州:解放军信息工程大学, 2015.

[3] 张月华,张新贺,刘鸿雁. AES算法优化及其在ARM上的实现[J]. 计算机应用, 2011, 31(6): 1539-1542.

[4] 敖天勇,贺章擎. 基于仿射变换的秘钥相关S盒构造方法[J]. 华中科技大学学报(自然科学版), 2015, 43(8): 39-42.

都是很不错的,其中以 D 块选取 16×16 的大小时,图像效果最好.图 1 给出了 R 块选取 4×4 , D 块选取 16×16 时的测试图像.

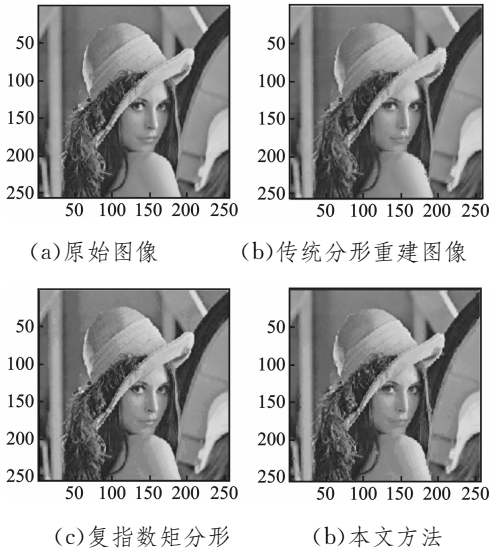


图 1 三种方法测试效果

6 结束语

本文提出了基于复指数矩和模糊聚类相结合的快速分形编码方法.首先根据 D 块的复指数矩不变量利用模糊聚类对 D 块进行分类,再根据 R 块的复指数矩不变量寻找最优匹配的 D 块.实验表明,在解码图像质量不变的同时,大大提高了分形编码的速度.在综合考虑编码时间、重建图像质量及算法复杂度情况下,本文所提出的分形编码算法将图像的复指数矩与其他分类方法相结合优化分形编码,是今后进一步研究的课题.

参考文献:

[1] A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations [J].

[5] 赵静远. 分组密码算法 SAFER,CAST-256 和 PRIDE 的分析[D]. 济南:山东大学, 2015.

[6] 贾平, 徐洪, 戚文峰. 轻量 S 盒密码性质研究[J]. 密码学报, 2015, 2(6): 497-504. [7] Y. Lee. A 45nm 1. 3Ghz 16. 7 Double-Precision GFLOPS/W RISC-V Processor with Vector Accelerators[C]. The 40th European Solid-State Circuits Conference, September 22 ~24, 2014:199-202.

[8] 郭怡惠,李森森,戴紫彬,张立朝. RISC 微处理器 S 盒替换指令扩展[J]. 微电子学与计算机, 2014 , 31 (5): 53-57.

IEEE Trans. Image Process. 1992, L(1):18-30.

[2] 吴一全, 郭建军, 孙子翼. 基于 Krawtchouk 矩不变量和核模糊聚类的快速分形编码[J]. 光电子. 激光, 2010,21(9):1412-1416.

[3] M. K. Hu. Visual Pattern Recognition by Moment Invariants[J] IRE Transactions on Information Theory 1962, 8(2):179-187.

[4] M. R. Teague. Image Analysis via the General Theory of Moments [J]. J Opt Soc Am, 1980, 70 (8): 920-930.

[5] Y. Sheng and L. Shen. Orthogonal Fourier-Mellin moments for invariant pattern recognition[J]. J Opt Soc Am A, 1994, 11(6): 1748-1757.

[6] H. Ren,W. Wu,W. Bo,Y. Sheng,Z. Ping, Multidistortion-invariant image recognition with radial harmonic Fourier moments[J]. J Opt Soc Am A Opt Image Sci Vis. , 2003, 20(4):631-637.

[7] Z. Ping,H Ren, J. Zou,Y. Sheng,W. Bo. Generic orthogonal moments: Jacobi-Fourier moments for invariant image description [J]. Journal of Optoelectronics Laser , 2007, 40(4):1245-1254.

[8] 姜永静. 指数矩及其在模式识别中的应用[D]. 北京: 北京邮电大学, 2011.

[9] 张莉,周伟达,焦李成. 核聚类算法[J]. 计算机学报, 2002,25(6):587-591.

[10] 普运伟,金炜东,朱明等. 核模糊 C 均值算法的聚类有效性研究[J]. 计算机科学 ,2007 ,34(2):207-210.

作者简介:

朱智林 男,(1963-),博士,教授.研究方向为数字图像处理.

王永玉(通讯作者) 女,(1973-),博士,副教授.研究方向为图像处理. E-mail:yongyuwang@sina.com.

平子良 男,(1945-),硕士,教授.研究方向为数字图像处理.

(上接第 128 页)

[5] 赵静远. 分组密码算法 SAFER,CAST-256 和 PRIDE 的分析[D]. 济南:山东大学, 2015.

[6] 贾平, 徐洪, 戚文峰. 轻量 S 盒密码性质研究[J]. 密码学报, 2015, 2(6): 497-504. [7] Y. Lee. A 45nm 1. 3Ghz 16. 7 Double-Precision GFLOPS/W RISC-V Processor with Vector Accelerators[C]. The 40th European Solid-State Circuits Conference, September 22 ~24, 2014:199-202.

[8] 郭怡惠,李森森,戴紫彬,张立朝. RISC 微处理器 S 盒替换指令扩展[J]. 微电子学与计算机, 2014 , 31 (5): 53-57.

[9] 覃晓草,李树国. 一种 AES 算法中 S 盒和逆 S 盒替换的表达式方法[J]. 微电子学与计算机. 2015, 32(1): 112-115.

[10] 吴筱,郭培源,何多多. DES 和 SM4 算法的可重构研究 与实现 [J]. 计算机应用研究. 2014, 31 (3): 853-856.

作者简介:

常忠祥 男,(1985-),讲师,博士.研究方向专用处理器设计. E-mail:changzhongxiang0@126.com.