低面积复杂度 AES 低熵掩码方案的研究

姜久兴[1]，厚娇[1]，黄海[2]，赵玉迎[1]，冯新新[3]

（1. 哈尔滨理工大学理学院，黑龙江 哈尔滨 150080；2. 哈尔滨理工大学软件与微电子学院，黑龙江 哈尔滨 150080；
3. 哈尔滨理工大学计算机科学与技术学院，黑龙江 哈尔滨 150080）

摘 要：在 Nassar 等提出的循环移位 S 盒掩码方案（RSM）的基础上，提出了一种针对高级加密标准（AES）算法低熵掩码方案。该方案的核心思想是利用 S 盒共用思想降低面积复杂度，采用乱序技术提高系统安全性，并通过流水线技术提高系统的吞吐量。对于 AES，所提方案可将其 S 盒的数量从 16 个降低为 4 个（不包括密钥扩展模块）。实验表明，与 RSM 相比，组合逻辑、时序逻辑和存储面积分别降低了 69%、60%和 80%，能够抵御基于偏移量 CPA 攻击，具有更高的安全性。

关键词：循环移位 S 盒掩码方案；低熵掩码方案；S 盒共用；高级加密标准；流水线

中图分类号：TP309.2

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2019100

Research on area-efficient low-entropy masking scheme for AES

JIANG Jiuxing[1], HOU Jiao[1], HUANG Hai[2], ZHAO Yuying[1], FENG Xinxin[3]

1. School of Science, Harbin University of Science and Technology, Harbin 150080, China
2. School of Software and Microelectronics, Harbin University of Science and Technology, Harbin 150080, China
3. School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

Abstract: Based on the rotating S-box masking (RSM) proposed by Nassar et al, a low-entropy masking scheme for the advanced encryption standard (AES) was proposed. Reducing the area complexity by reusing the S-boxes, improving the hardware security by shuffling operation and improving the throughput by pipelining operation were the main idea of the proposed scheme. For the AES, the number of S-boxes could be reduced from 16 to 4 (key expansion module wasn't included). Compared with the RSM, the combinational logic, the dedicated logic and the memory size are reduced to 69%, 60% and 80% respectively. In addition, the theoretical analysis shows that the proposed scheme can resist offset based CPA attack, thus has higher security than the RSM.

Key words: rotating S-box masking, low-entropy masking scheme, S-box reusing, AES, pipeline

1 引言

高级加密标准（AES, advanced encryption standard）算法因安全性、效率、灵活性等方面具有良好的性能，因此被广泛地应用在实践中。侧信道攻击（SCA, side-channel attack）技术[1]的出现对加密芯片的安全性构成了很大的威胁，典型的 SCA 包括时间分析攻击、电磁辐射攻击、功耗分析攻击[2]等，其中差分功耗攻击（DPA, differential power attack）[3]，尤其是高阶 DPA，对芯片的硬件安全的威胁最大，采用掩码方案是抵御 DPA 最有效的方法。AES 加密算法的安全性将随着掩码阶数的增加而提高，但掩码阶数的增加会带来硬件成本成倍的增加。目前，掩码方案主要有查找表掩码[4-5]、加法链掩码[6]和复合域掩码[7]这 3 种，相较于后 2 种方案，查找表掩码方案具有实现速度快、易于实现等特点，但

收稿日期：2018–10–09；修回日期：2019–03–29

通信作者：黄海，ic@hrbust.edu.cn

基金项目：国家自然科学基金资助项目（No.61604050, No.51672062）

Foundation Item: The National Natural Science Foundation of China (No.61604050, No.51672062)

存在存储空间大的不足，难以应用到资源受限的设备中。针对这一问题，Nassar 等[8]提出了一种循环移位 S 盒掩码（RSM, rotating S-box masking）方案，该方案能够有效地降低面积复杂度，是一种安全性和性能的折中方案，能够抵抗一阶和二阶 SCA。然而，对于一些面积受限的设备，例如智能卡、物联网终端设备等，RSM 仍不能满足实际的应用需求[9]。

本文在 RSM 方案的基础上，分析了 RSM 方案掩码值的特点，提出了一种基于 S 盒共用的方案，所提方案能使面积复杂度进一步降低，并针对 AES 进行了掩码方案架构设计与硬件实现，对整体架构进行了流水线的设计。所提方案与 RSM 方案相比，大幅度降低了面积复杂度，从理论上和实验上证明了具有更高的安全性和更好的性能。所提方案在满足性能和安全性的前提下，能够有效地节约硬件资源，降低实现成本，对资源受限的设备和民用小型服务器具有重要的意义。

2 相关工作

AES 加密算法的掩码方案中，基于查找表的掩码方案通过采用 S 盒重计算技术增加攻击难度来抵御 SCA。1999 年，Chari 等[10]首次提出了随机查找表方案，对 S 盒进行了 $T(u)=S(u\oplus r)\oplus s$ 的随机化处理，其中，u 是明文，r 是输入掩码，s 是输出掩码。但是该方案没有对 AES 的所有轮都添加掩码，因此仅能抵御一阶 DPA。文献[11]对文献[10]中所提方案采用中间轮攻击操作，得到了正确的密钥。2001 年，Itoh 等[12]采用固定值掩码方案抵御中间轮攻击，但所有轮操作都采用同一掩码值。该方案着重考虑 AES 加密算法的加密速度，将随机掩码和根据随机掩码计算得到的 S 盒存储在 ROM 中，减少了 RAM 的空间。但是该方案只能抵御一阶 SCA，不能抵抗二阶及以上的 SCA，因为通过对 2 个经过掩码的中间结果进行异或处理，就可以去掉掩码值。

Nassar 等[8]提出了 RSM 方案，采用了低熵掩码设计，利用 16 个不同的 S 盒，解决了固定值掩码方案难以抵抗二阶 SCA 攻击的问题。RSM 掩码原理如式(1)所示。

$$\mathrm{SB}'_j=\mathrm{SB}(M_j\oplus X)\oplus M_{(j+1)\bmod 16},\ 0\leqslant j\leqslant 15 \quad (1)$$

其中，SB 表示 128 位数据的字节替代操作，X 表示需要掩码的 128 位数据，$M_j=\{m_j,m_{(j+1)\bmod 16},\cdots,m_{(j+15)\bmod 16}\}$。$j$ 的选取在第一轮，之后每轮都是固定值（即掩码值循环左移），因此掩码值的选取共有 16 种可能。在每轮加密操作结束后需要进行掩码补偿，即去掉上次的掩码，添加新的掩码。为了提高 AES 加密算法的计算速度，使用重计算的 16 个 S 盒存储在 ROM 中。

RSM 方案的优点在于以牺牲少量面积和性能为代价，达到抵御一阶 SCA 攻击、二阶 SCA 攻击、零值攻击等攻击，有效地提高了 AES 的安全性。2014 年，Yamashita 等[13]公布了一种变种的 RSM 方案 vRSM（variant RSM），该方案去掉了中间轮的重新掩码操作（这些操作对于 RSM 方案是不可或缺的），在不影响安全性的前提下，进一步降低掩码的复杂度，其面积开销为 RSM 方案的 90%，但该方案的输出掩码值是由输入掩码值决定的。2015 年，徐佩[14]提出了改进的 RSM 方案，该方案充分考虑了掩码值的汉明质量，对随机掩码的选取进行了优化设计，同时对 S 盒循环移位的次数进行了随机化设计，能够抵抗基于一阶偏移量的相关性功耗分析（CPA, correlation power analysis），这是 RSM 方案不具备的。为了增加算法的加密速度，文献[14]对 AES 的前两轮和最后两轮采用随机掩码的方式，其他中间轮采用固定值掩码的方式，这样中间轮能够采用相同的掩码 S 盒，减少计算掩码 S 盒数量，提高速度。可以看出，文献[14]提出的方案是 RSM 方案的改进方案，但存在不能抵御中间轮的攻击和面积相对于其他方案面积有所增加的缺点，从安全性方面看，仍不是一个最优的方案。

综上，现有的基于查找表掩码的掩码方案难以兼顾安全性和复杂度。本文在分析 RSM 方案掩码值特点的基础上，提出了一种低熵掩码方案，通过 S 盒共用思想减少掩码 S 盒的数量，从而进一步降低 RSM 方案的面积复杂度。

3 S 盒共用的思想

3.1 掩码值的选取

根据文献[15]可知，RSM 方案的掩码值必须是正交的，这样才能确保算法的安全性。因此，本文掩码值的选择方法采用文献[16]中的方法，具体如下。假设正交矩阵的维度是 Q，列数是 N，行数是 $Q\wedge J$，J 的选取需满足 $N=\dfrac{Q\wedge J-1}{Q-1}$，首先利用置换方法获得正交矩阵[16]，其中，$Q$=2，$N$=15，$J$=4，

生成的正交矩阵 $\boldsymbol{A}$ 有 15 列，如式(2)所示。这 15 列的值分别为 00ff、0f0f、0ff0、3333、33cc、3cc3、3c3c、5555、55aa、5a5a、5aa5、6666、6699、6969、6996，然后从正交矩阵 $\boldsymbol{A}$ 中随机选择 8 列构造满足约束条件的 16 个掩码值。通过分析矩阵 $\boldsymbol{A}$ 能够看出，每一列中的 4 个十六进制数的要么相同，要么取反，这也是本文设计 S 盒共用掩码方案能够实现的基础。

$$\boldsymbol{A}=\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1\\
0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&1\\
0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0\\
0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0\\
0&1&1&0&0&1&1&1&1&0&0&1&1&0&0&1\\
0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&1\\
0&1&1&1&1&0&0&1&1&0&0&0&0&1&1&0\\
1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0\\
1&0&1&0&1&0&1&1&0&1&0&1&0&1&0&1\\
1&0&1&1&0&1&0&0&1&0&1&1&0&1&0&1\\
1&0&1&1&0&1&0&1&0&1&0&0&1&0&1&0\\
1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&1\\
1&1&0&0&1&1&0&1&0&0&1&1&0&0&1&0\\
1&1&0&1&0&0&1&0&1&1&0&1&0&0&1&1\\
1&1&0&1&0&0&1&1&0&0&1&0&1&1&0&0
\end{bmatrix} \tag{2}$$

3.2　S 盒共用掩码方案

S 盒共用掩码方案是一个通用方案，适用于掩码值满足正交向量要求的任意掩码方案。以随机选取{0x00 0x0f 0x36 0x39 0x53 0x5c 0x65 0x6a 0x95 0x9a 0xa3 0xac 0xc6 0xc9 0xf0 0xff}为例，对该方案进行详细说明。能够看出随机选取的 16 位掩码值中，高 4 位和低 4 位不是相同就是互补。根据这一特征，进行如下分组，用 M_t 表示，$t\in[1,4]$。

$$M_1=[00,0\mathrm{f},\mathrm{f}0,\mathrm{ff}] \tag{3}$$

$$M_2=[36,39,\mathrm{c}6,\mathrm{c}9] \tag{4}$$

$$M_3=[53,5\mathrm{c},\mathrm{a}3,\mathrm{ac}] \tag{5}$$

$$M_4=[65,6\mathrm{a},95,9\mathrm{a}] \tag{6}$$

在每组掩码分组中选用的掩码值满足如式(7)所示的条件。

$$S_m(x\oplus M_{tr})=S(x)\oplus M_{t(r+1)\bmod 4} \tag{7}$$

其中，x 是无掩码的 8 位数据，M_{tr} 为第 t 个掩码分组中的第 r 个掩码值，是 S 盒的输入掩码，$M_{t(r+1)\bmod 4}$ 是 S 盒的输出掩码，S 代表 S 盒操作，S_m 代表带掩码的 S 盒操作。

全部 128 位明文的掩码满足

$$\mathrm{SB}'_j=\mathrm{SB}(M_{p,q,h,n}\oplus X)\oplus M_{(p+1),(q+1),(h+1),(n+1)},\quad \forall p、q、h、n\in\{1,4\} \tag{8}$$

$$\begin{aligned}M_{pqhn}=\{&M_{Pp},M_{Qq},M_{Hh},M_{Nn},M_{P(p+1)\bmod 4},M_{Q(q+1)\bmod 4},\\&M_{H(h+1)\bmod 4},M_{N(n+1)\bmod 4},\cdots,M_{P(p+3)\bmod 4},M_{Q(q+3)\bmod 4},\\&M_{H(h+3)\bmod 4},M_{N(n+3)\bmod 4}\}\end{aligned} \tag{9}$$

其中，P、Q、H、$N\in t$，表示在上述掩码分组中挑选任意一组；$p,q,j,n\in[1,4]$，表示从每组中选择第几个值为掩码分组的第一个掩码值。上述的掩码分组中能够随机挑选一组作为共用 S 盒，假设在第 t 组中可选共用的 S 盒输入掩码为 mt_2，输出掩码为 mt_3。例如在 M_2 中，选择“39”为输入掩码，则“c6”为输出掩码，表示为 Sbox_$M_2$2_$M_2$3，第 t 组的共用 S 盒表示为 Sbox_$M_t r$_$M_t(r+1)$mod4。S 盒共用掩码方案原理如图 1 所示。

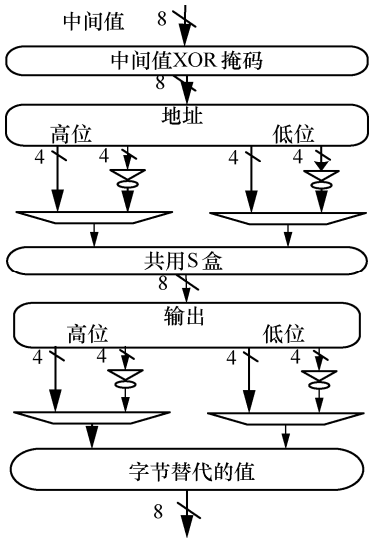


图 1　S 盒共用掩码方案原理

在每组掩码分组中，首先根据选择的输入输出掩码计算得到共用 S 盒，剩余的 3 组掩码值与所选择的输入输出掩码存在固定关系，即掩码值的高低位不是相同，就是互补。按照这个规律，利用 S 盒共用掩码方案可以有效地降低掩码 S 盒的数量，具体过程如下。当 32 位的中间值异或一组掩码值时，能够通过判断该组掩码值的输入掩码和共用 S 盒输入掩码高低位关系来选择地址；如果值相同，那么 S 盒的地址不变，否则，S 盒的地址取反；同样能够通过判断

该组掩码值的输出掩码和该组共用 S 盒的输出掩码的高低位关系来选择输出；如果值相同，那么 S 盒的输出不变，否则，S 盒输出取反。因此，每组就能够利用同一个 S 盒(8 位)来实现字节替代操作。

4 AES 算法的实现

4.1 线性部分的掩码实现

4.1.1 线性操作的掩码实现

1) XOR 掩码——密钥加操作

XOR（exclusive-OR）操作将轮密钥与输入数据进行异或操作，其结果进行字节替代操作，该中间结果和密钥有很强的关系，所以一定要对 XOR 操作进行掩码。密钥加是线性操作，能够利用异或操作实现掩码，但必须注意和掩码值异或的顺序，一定要确保所有中间值都是经过掩码的，不能存在某个中间值没有进行掩码就直接进行某个变换，这样才能够保证 XOR 操作不泄露任何的中间结果。

2) SR 掩码——行移位操作

SR（shift row）操作在字节替代变换之后，由式(7)可以看出，经过字节替代操作的中间值仍受掩码保护，所以不需要再添加新的掩码值。

3) MC 掩码——列混合操作

SR 之后进行 MC（mix column）操作，与 SR 类似，该操作的中间值也是具有掩码的，因此不需要再添加新的掩码值。

4.1.2 掩码补偿

本文方案的掩码补偿发生在每轮加密结束时，通过异或操作去掉原掩码值，增加新掩码来实现掩码更新。为了得到正确的密文，最后一轮仅需去除现在的掩码值，不需要添加新掩码。进行字节替代操作之后，中间值是具有掩码的，根据式(7)~式(9)，把掩码值按照矩阵形式进行排列，其每轮的行变换关系如图 2 所示。

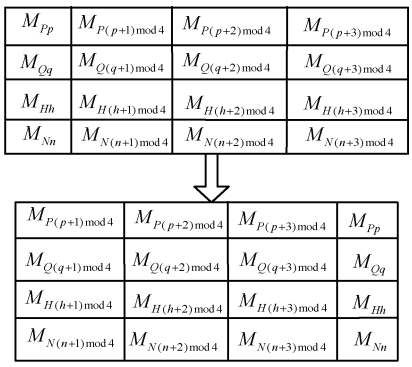


图 2 掩码值每轮的行变换关系

对非最后一轮的输入掩码依次进行行变换（LT, line transformation）操作、SR 掩码、MC 掩码和添加新的掩码操作，而最后一轮仅需对输入掩码进行 LT 操作和 SR 操作。为了保证算法的安全性，在每轮 XOR 操作结束时，需要进行掩码补偿操作，即先添加新掩码然后去掉之前的掩码，具体操作如图 3 所示。

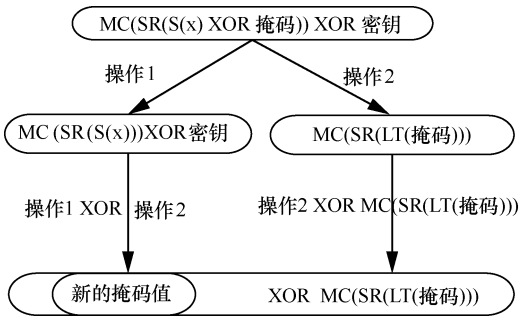


图 3 掩码补偿操作

4.2 AES 字节替代模块的实现

AES 算法中字节替代是唯一的非线性操作，为了满足掩码的基本原则，设计的式(7)能够满足对字节替代操作的掩码，在进行字节替代时要通过重新计获得新的 S 盒。在 4 个掩码值分组中，分别计算 4 个掩码分组中每组内的共用 S 盒，该组内共用 S 盒的选择有 4 种情况，能够随机挑选任意一个作为共用的掩码 S 盒，所以需要 4 个共用 S 盒。

S 盒共用掩码方案在进行字节替代操作时，输入值为 32 位，32 位的数据被分为 4 组，每一组为 8 位。进行字节替代操作时，根据轮操作中间值的掩码值，选择相应的共用 S 盒完成操作。在进行字节替代操作时，这 4 组数据是并行执行的。S 盒共用掩码方案字节替代模块的原理如图 4 所示。首先 32 位明文被分为 4 组，每组 8 位，这 4 组数据分别与 M_P、M_Q、M_H 和 M_N 进行异或操作，然后进行字节替代操作，最后把这 4 组字节替代后的值组合成 32 位的数据。

4.3 AES 密钥扩展模块的实现

AES 的密钥扩展是把初始的 128 位密钥平均分为 4 组，进行密钥扩展操作，共产生 44 组密钥（包括初始密钥），每组 32 位，当分组的组数是 4 的倍数时，需要进行字节替代操作，这也是密钥扩展模块唯一的非线性操作，字节替代操作共需要 4 个 S 盒，其中，S 盒与轮操作中的 S 盒相同，因此共用 S 盒方案也能够用于密钥扩展模块。对于密钥扩展

操作而言，其线性部分和 S 盒都可按照轮操作的掩码方式进行掩码，从而提高安全性。在进行密钥扩展时，首先 128 位密钥与掩码值进行异或操作，然后分为 4 组进行密钥扩展，由于应用 S 盒共用的掩码方案，因此在进行字节替代操作时仅需要 4 个掩码 S 盒。由于密钥扩展产生的密钥添加了掩码，为了得到正确的密文，在进行密钥加操作之后要去掉密钥的掩码值，因为列混合操作之后的中间结果是带有掩码值的数据，所以并不会泄露真实的中间结果。

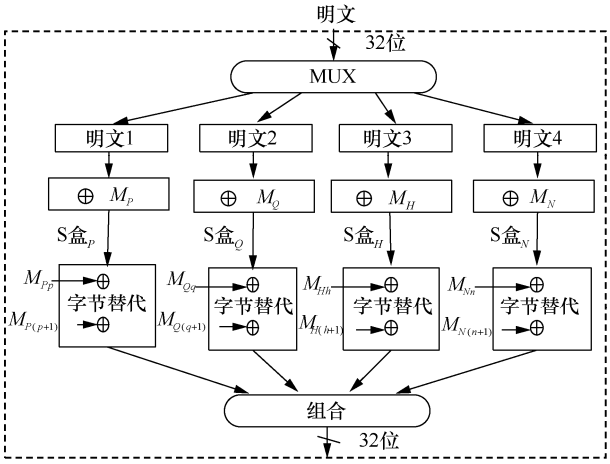


图 4　S 盒共用掩码方案的字节替代模块

4.4　流水线设计

为了提高算法的执行效率，对 S 盒共用掩码方案进行了流水线设计[17-19]。根据文献[19]可知，若交换算法的行移位模块与字节替代模块的操作顺序，不会对密文有影响，从而可实现流水线设计。由于 S 盒共用掩码方案字节替代用到了 4 个 S 盒，最终实现 128 位数据的字节替代操作需要 4 个周期，本文设计的流水线时空图如图 5 所示，其中“(1)”~ “(4)”表示要加密的 4 个 32 位数据。当 128 位明文输入时，按照顺序每组 32 位进行拆分。每组数据按照字节替代、MC、XOR 与掩码补偿的顺序进行操作，再对其他 32 位数据依次进行同样的操作，直至执行完 128 位的数据。这样的方式能够确保每个周期所有的硬件模块都在工作，增加了硬件使用率，提高了 S 盒共用掩码方案的加密速度。

S 盒共用掩码方案的流水线设计数据加密结构框架如图 6 所示，该框架主要包括 18 个模块，其中，拆分模块对数据进行拆分，选择模块对多组数据进行选择，重组模块对数据进行重组处理，缓存器主要对数据的传输进行缓冲和存储。

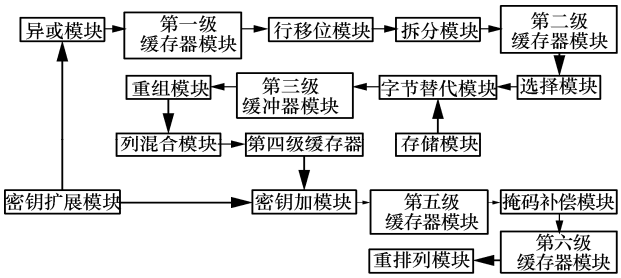


图 6　流水线设计数据加密结构框架

4.5　AES 的整体实现

本文设计的 AES 掩码方案如图 7 所示。该方案首先将明文与随机选择的 4 组掩码值进行异或，然后进行 XOR 操作，这个结果即为中间轮的输入。对于非最后一轮的所有中间轮依次进行 SR、字节替代、MC、XOR 与掩码补偿操作，最后一轮加密依次进行 SR、字节替代、XOR，并进行掩码补偿操作得到最后的密文并输出。图 7 中 K_0~K_{43} 表示的是 44 组密钥值，每组 32 位。

为提高算法的安全性，在数据进行加密时采用乱序的方式对 4 组数据进行字节替代、MC、XOR 和掩码补偿等操作。因为增加了乱序执行这 4 组 32 位数据的操作，所以在这 128 位数据完成上述操作后，要对其进行重新排序，排序之后的结果为下一轮的

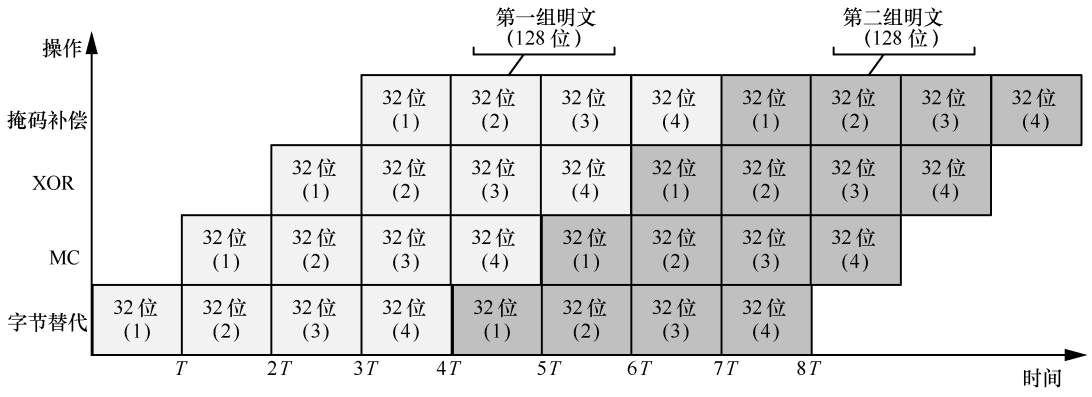


图 5　流水线时空图

数据输入，该操作能够确保正确密文的输出。

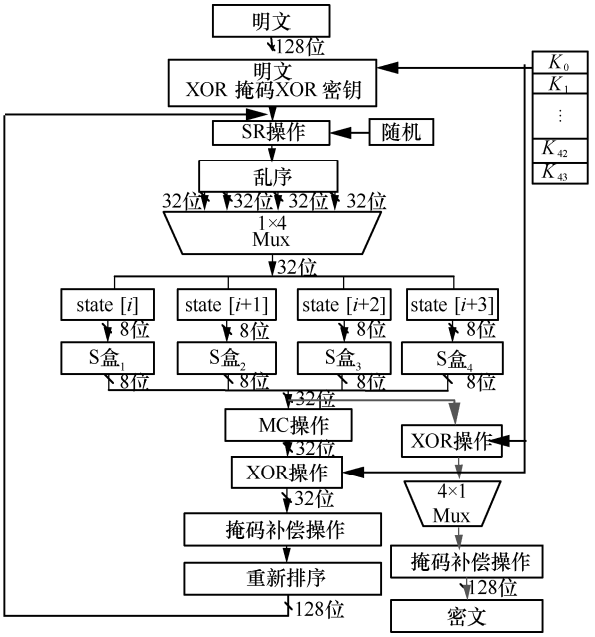


图 7　AES 掩码实现

字节替代操作采用串行查找方式进行，每 128 位数据需要进行 4 次字节替代操作，通过比较 RSM 方案和 S 盒共用掩码方案可知，RSM 方案共占用 16 个掩码 S 盒，S 盒共用掩码方案仅占用 4 个，为 RSM 方案的 25%。但进行 S 盒操作时，128 位数据采用串行查找的方式，依次处理 32 位数据，这样使字节替代操作的吞吐量变为原来的$\frac{1}{4}$，适用于面积受限的应用。

为了增加算法的运算速度，S 盒共用掩码方案利用与固定值掩码方案相同的空间存储方式，在进行加密之前，提前选好掩码值，根据选择的掩码值计算好与之对应的共用 S 盒，并把共用 S 盒存储在 ROM 中。在进行加密时，根据设计方案选取随机掩码和与之对应的掩码 S 盒执行操作，这种方式能够降低占用的 RAM 空间。S 盒共用掩码方案需要在 ROM 中存储的值如下。

1) 具有正交向量特征的掩码值 M_l。

2) 根据设计规则选择的每组掩码值的偏移量 p、q、h、n。

3) 根据选择的输入输出掩码计算得到的掩码 S 盒，即共用 S 盒，数目为 4。

4) 根据输出掩码计算每轮的掩码补偿值。

通过上述的设计方案，能够直接从 ROM 中调用需要的值进行加密运算，RAM 只需要存储临时的变量，如 XOR、SR 等操作的中间值。因为计算共用 S 盒需要花费很多的时间，所以对整个算法的时间花费主要集中在初始阶段[9]，之后的操作当需要某些中间值时，直接从 ROM 中调用即可，通过这样的方式提高了算法的运算速度。与 RSM 方案相比，S 盒共用掩码方案占用更少的面积，并且安全性也有所提高。

5　实验结果和比较

5.1　功能仿真

为了验证 S 盒共用掩码方案的功能正确性，对 S 盒共用掩码方案进行了 Verilog HDL 建模，通过 Modelsim SE 10.0c 仿真软件进行仿真。无掩码 AES 方案、RSM 方案和 S 盒共用掩码方案的字节替代的仿真结果分别如图 8~图 10 所示。

图 8　无掩码 AES 方案字节替代操作

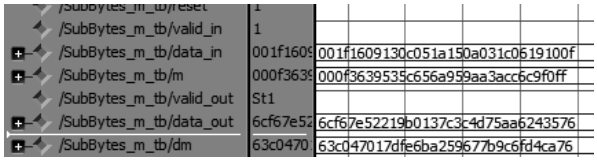


图 9　RSM 方案字节替代操作

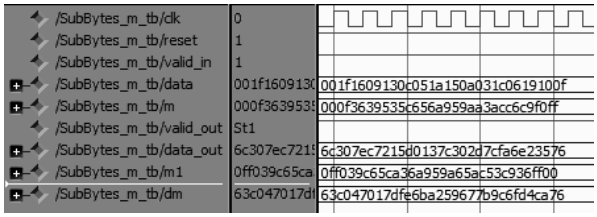


图 10　S 盒共用掩码方案字节替代操作

在测试 3 种方案字节替代的功能时，RSM 方案和 S 盒共用掩码方案采用相同的测试数据。图 8~图 10 中 data_in 和 data 是需要进行字节替代操作的 128 位数据，data_out 是进行字节替代之后的 128 位输出数据，m 是掩码值，dm 是去除掩码后的值，m_1 是共用 S 盒对应输入掩码的输出掩码。可以看出，图 9 和图 10 中的 dm 值与图 8 中的 data_out 值相同，从而证明了 S 盒共用掩码方案中字节替代的正确性。

基于 S 盒共用掩码方案的整个 AES 掩码方案仿真结果如图 11 所示，流水线设计的功能仿真结果如图 12 所示。图 11 中，cipher_key 是密钥值，plain_text

是明文数据，m 是掩码值，cipher_text 是加密得到的密文。从图 12 中能够看出，128 位密文“cipher_text”需要 4 个周期输出，每个周期输出 32 位。从测试结果可以看出，根据输入的明文和密钥，能够输出正确的密文，可以验证本文所提方案的功能正确性。

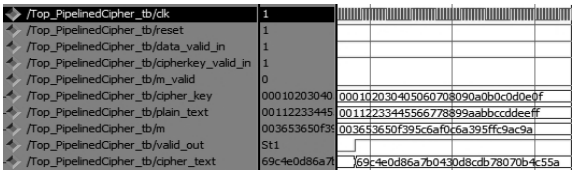


图 11　S 盒共用掩码方案实现 AES 算法

图 12　流水线设计的仿真

5.2　安全性分析

SCA 是利用 AES 操作过程中的中间结果与密钥之间的相关性进行攻击的，因此证明中间结果的概率分布与密钥无关，即可证明算法的安全性。把 128 位的输入数据分为 4 个 32 位的数据进行处理，只用一个字节替代架构，依然能够保证 AES 算法自身的安全性。

引理 1　若被掩码的中间值可表示为 $u^{'}=ux$，其中，u 为计算得到的中间值，x 是随机掩码。那么 $u^{'}$ 的概率分布与密钥是不相关的[20]。

文献[20]中给出了引理 1 论证，因此根据该理论可以证明 S 盒共用掩码方案可以抵抗一阶 SCA。定义评估的相关系数[21]为

$$\rho_{\text{opt}}=\frac{\sigma[\mathrm{E}[C\mid Z]]}{\sigma[C]} \tag{10}$$

利用 SAT(SAT-solver)求解[22]对 S 盒共用掩码方案有 RSM 方案具有的安全性进行证明，即可以抵抗一阶与二阶零偏移量的 CPA，有以下结果。

1) 掩码值非随机的，即掩码值是固定值。一阶相关系数 $\rho_{\text{opt}}^{(1)}$ 与二阶相关系数 $\rho_{\text{opt}}^{(2)}$ 均不等于 0。如果是输入 8 B 数据，那么交互信息 $I[\mathrm{HW}(x\oplus m);x]$ 的平均值是 2.544 2 B，其中，HW(·)表示汉明质量。

2) 若 2 个随机变量互补，那么它们的交互信息为 MIA=1.817 6 B。对于 2 个互补的掩码，m 是均匀分布的一对 $\{\widetilde{m},-\widetilde{m}\}$，例如{0x65，0x9a})，$\rho_{\text{opt}}^{(1)}=0$ 但 $\rho_{\text{opt}}^{(2)}\neq 0$。所以主要的目标是清掉 $\rho_{\text{opt}}^{(1)}$，1.817 6 B 对交互信息来说仍是相当大的。

3) 16 个掩码利用 SAT 求解，$\rho_{\text{opt}}^{(1)}=\rho_{\text{opt}}^{(2)}=0$，能够发现交互信息 $I[\mathrm{HW}(x\oplus m);x]$ 是 0.216 8 B。

根据以上引理和结果可证明，S 盒共用掩码方案可以抵抗一阶和二阶 SCA，有 RSM 具有的安全性。

因为 CPA 是利用中间值与密钥间的联系对加密方案进行攻击的，所以通过证明中间值的概率分布不受密钥的影响，即不相关，就可以证明加密方案可抗 CPA。在文献[14]中给出了一种掩码方案能否抵御 CPA 的定理，如定理 1 所示。

定理 1　若掩码方案存在 n 个中间值 $u_1^{'},u_2^{'},\cdots,u_n^{'}$，它们的概率分布都与密钥无关，而且 $u_1^{'},u_2^{'},\cdots,u_n^{'}$ 的联合概率分布也不受密钥的任何影响，就可以证明该方案能抵御一阶和高阶 CPA。

假设经过掩码的中间值 $u^{'}=ux$，其中，u 是算法运算的中间值，x 是随机掩码。那么 $u^{'}$ 的概率分布与密钥没有关系，因此可证明该算法能够抵御一阶 CPA。同理，假设有 $n(n>1)$ 个中间值 $u_1^{'},u_2^{'},\cdots,u_n^{'}$，$u_n^{'}=u_i\oplus x\ \ (1\leqslant i\leqslant n)$，且 $u_1^{'},u_2^{'},\cdots,u_n^{'}$ 均服从均匀分布，并且被随机变量进行掩码，$u_1^{'},u_2^{'},\cdots,u_n^{'}$ 同样相互独立，并满足

$$\begin{aligned}&P(u_1^{'}=\alpha_1,u_2^{'}=\alpha_2,\cdots,u_n^{'}=\alpha_n)=\\&P(x_1=u_1\oplus\alpha_1,x_2=u_2\oplus\alpha_2,\cdots,x_n=u_n\oplus\alpha_n)=\\&P(x_1=\alpha_1^{'},x_2=\alpha_2^{'},\cdots,x_n=\alpha_n^{'})=P(x_1=\alpha_1^{'})\cdot\\&P(x_2=\alpha_2^{'})\cdots P(x_n=\alpha_n^{'})=\frac{1}{N+1}\end{aligned} \tag{11}$$

其中，$\alpha_1^{'},\alpha_2^{'},\cdots,\alpha_n^{'}$ 为 $u_1^{'},u_2^{'},\cdots,u_n^{'}$ 的任意值；N 为各变量的最大值，当各变量的位数为 8 时，N=255。由此证明 $u_1^{'},u_2^{'},\cdots,u_n^{'}$ 的联合概率分布与密钥不相关，即 $n(n>1)$个被掩码的中间值的联合概率分布与密钥不相关。由于 S 盒共用掩码方案的中间结果的概率分布与密钥不相关，根据定理 1 可证，S 盒共用掩码方案在理论上能够防御高阶 CPA。

此外，文献[14-15]指出，RSM 方案的安全性存在以下 2 个漏洞。

1) RSM 方案的掩码值经过异或操作，得到的值的汉明质量存在特定的规律，即 HW(m_j XOR m_{j+1})=4，HW(m_0 XOR m_{15})=8。

2) 该方案的 S 盒循环偏移量是一个常数，为 1。

S 盒共用掩码方案相邻掩码值经过异或操作后不存在 1)的特征，并且 S 盒共用掩码方案不需要循环移位，因此也没有 2)的缺陷。表 1 给出了 4 种方

案的安全性对比。

综上所述，S 盒共用掩码方案的所有中间值与密钥不相关，可以抵抗一阶和高阶的 CPA；采用掩码值随机选取和字节替代顺序随机执行，使相邻掩码进行异或操作之后的汉明质量为随机值，因此可以抵御基于偏移量的 CPA。

5.3 与其他方案的比较

在相同的实现方式下，对 AES 加密算法的无掩码 AES 方案、RSM 方案[8]、vRSM 方案[13]和 S 盒共用掩码方案法进行 Verilog 建模，并在 FPGA 上实现。所采用的 FPGA 芯片为择 Cyclone III 系列（EP3C120F78017），逻辑综合结果如表 2 所示。

表 2 中，括号内的百分比是相比于无掩码方案所占用的资源，“+”号代表增加的百分比，“−”号代表减少的百分比。从表 2 中能够看出，本文提出的 S 盒共用掩码方案需要的掩码 S 盒数目是 RSM 方案的 $\frac{1}{4}$（不包括密钥扩展模块），因此很大程度地减少了面积开销。S 盒共用掩码方案需要 3 184 个总逻辑单元，总逻辑单元数为 RSM 方案的 58%、vRSM 方案的 61%、无掩码 AES 方案的 63%；其中，时序逻辑单元有 2 766 个，组合逻辑单元有 1 838 个，所需的存储位是 84 545 位，其大小为 RSM 方案、vRSM 方案和无掩码方案的 21%。本文所提方案在执行字节替代操作时采用串行方式，因此每轮字节替代操作需 4 个周期，是其他 3 种方案的 4 倍。

对无掩码 AES 方案、RSM 方案及 S 盒共用掩码方案采用流水线的方式实现，得到的结果如表 3 所示。从表 3 可知，S 盒共用掩码方案使用了 4 352 个总逻辑单元，是 RSM 方案的 39%；组合逻辑单元为 1 986 个，是 RSM 方案的 31%；时序逻辑单元为 4 220 个，是 RSM 方案的 40%；存储位为 81 956 位，是 RSM 方案的 20%。结果显示，S 盒共用掩码方案在占用资源的花费上有着特别大的优势。采用本文设计的流水线设计实现本文方案，4 个 128 位明文加密

表 1 安全性分析

方案	一阶 SCA	高阶 SCA	基于偏移量的一阶 CPA
无掩码 AES 方案	×	×	×
RSM 方案[8]	○	○	×
vRSM 方案[13]	○	○	×
S 盒共用掩码方案	○	○	○

注：“×”代表无法抵抗攻击，“○”代表可抵抗攻击。

表 2 4 种 S 盒实现方案比较

方案	S 盒数目/个	掩码选取	总逻辑单元/个	时序逻辑单元/个	组合逻辑单元/个	存储大小/位	128 位数据加密所需周期
无掩码 AES 方案	16	--	5 080	1 500	5 065	409 600	10
RSM 方案[8]	16	16	5 512（+9%）	220（−85%）	5 497（+9%）	409 600（+0%）	10
vRSM 方案[13]	16	16	5 202（+2%）	220（−85%）	5 399（+7%）	409 600（+0%）	10
S 盒共用掩码方案	4	6 144	3 184（−37%）	2 766（+84%）	1 838（−64%）	84 545（−79%）	40

注：S 盒的数目不包含密钥扩展模块; 掩码值的选择种类也是在同样数目(16)条件下的可能情况。

表 3 流水线实现方案的对比

方案	总逻辑单元/个	时序逻辑单元/个	组合逻辑单元/个	存储大小/位	4×128 位数据加密所需周期
无掩码 AES 方案	9 975	9 369	5 915	409 600	42
RSM 方案[8]	11 119（+11%）	10 530（+12%）	6 459（+9%）	409 600（+0%）	52
S 盒共用掩码方案	4 352（−56%）	4 220（−55%）	1 986（−66%）	81 956（−80%）	91

需要 91 个周期；采用非流水线设计实现本文方案，一个 128 位明文加密需要 79 个周期。因此，对于 4 个 128 位明文加密来说，采用流水线设计和非流水线设计地加速比为

$$S = \frac{4 \times 79}{91} = 3.47 \tag{12}$$

式(12)中 S 值大于 1，说明本文设计的流水线方案能够提高 AES 加密算法的加密速度。

综上，相比于 RSM 方案，S 盒共用掩码方案所需硬件资源大幅度减少。此外，在使用相同数目掩码值（16 个）的情况下，RSM 方案和 vRSM 方案的掩码值选取仅有 16 种可能。本文提出的 S 盒共用掩码方案在选取掩码值时可以随机选择 4 组数据，因此可选择的掩码值共有 24（即 2×3×4）种可能，在选择每组中的第一个掩码值时又有 256（即 4×4×4×4）种可能，因此本文提出的方案中掩码值选择一共有 6 144（即 24×256）种可能，相当于 RSM 方案和 vRSM 方案的 384 倍。掩码方案的安全性与掩码值的随机性成正比，可以看出，本文所提方案的安全性高于 RSM 方案和 vRSM 方案。

6　结束语

本文在文献[8]的基础上，提出了一种 S 盒共用的低熵掩码方案，并把这种方案应用到 AES 算法上。该方案的轮操作中加密的最小单元为 32 位，而不是 128 位，并利用流水线技术提高系统性能。实验结果表明，本文所提方案相对于 RSM 方案大幅度降低了面积复杂度，S 盒的数量从 16 个降低到 4 个。从理论上证明本文所提方案比 RSM 方案有更高的安全性，可以抗一阶、高阶的 SCA 与基于偏移量的 CPA 攻击。

参考文献：

[1] KOCHER P C. Timing attacks on implementations of Diffie- Hellman, RSA, DSS, and other systems[C]//International Cryptology Conference on Advances in Cryptology. Springer, 1996: 104-113.

[2] 肖国镇, 白恩健, 刘晓娟. AES 密码分析的若干新进展[J]. 电子学报, 2003, 31(10): 1549-1554.
XIAO G Z, BAI E J, LIU X J. Some new developments on the cryptanalysis of AES[J]. ACTA Electronica sinica, 2003, 31(10): 1549-1554.

[3] KOCHER P C, JAFFE J, JUN B. Differential power analysis[C]// International Cryptology Conference on Advances in Cryptology. Springer, 1999: 388-397.

[4] TANG M, QIU Z L, GUO Z P, et al. A generic table recomputation-based higher-order masking[J]. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2017, 36(11): 1779-1789.

[5] PAMMU A A, CHONG K S, NE K Z L, et al. High secured low power multiplexer-LUT based AES S-box implementation[C]//International Conference on Information Systems Engineering. Springer, 2016: 3-7.

[6] 黄海, 冯新新, 刘红雨, 等. 基于随机加法链的高级加密标准抗侧信道攻击对策[J]. 电子与信息学报, 2019,41(2): 348-354.
HUANG H, FENG X X, LIU H Y, et al. Random addition-chain based countermeasure against side-channel attack for advanced encryption standard[J].Journal of Electronics & Information Technology, 2019, 41(2): 348-354.

[7] AHN S, CHOI D. An improved masking scheme for S-box software implementations[C]//16th International Workshop on Information Security Applications. KISSC, 2016: 200-212.

[8] NASSAR M, SOUISSI Y, GUILLEY S, et al. RSM: a small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs[C]//Design, Automation & Test in Europe Conference & Exhibition. IEEE, 2012: 1173-1178.

[9] HUANG H, LIU L B, HUANG Q H, et al. Low area-overhead low-entropy masking scheme (LEMS) against correlation power analysis attack[J]. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2019, 38(2): 208-219.

[10] CHARI S, JUTLA C S, RAO J R, et al. Towards sound approaches to counteract power-analysis attacks[C]//International Cryptology Conference on Advances in Cryptology. Springer, 1999: 398-412.

[11] FAHN P N, PEARSON P K. IPA: a new class of power attacks[C]// International Workshop on Cryptographic Hardware and Embedded Systems. Springer, 1999: 173-186.

[12] ITOH K, TAKEBAKA M, TORII N. DPA countermeasure based on the "masking method"[C]//International Conference Seoul on Information Security and Cryptology. Springer, 2001: 440-456.

[13] YAMASHITA N, MINEMATSU K, OKAMURA T, et al. A smaller and faster variant of RSM[C]// Design, Automation and Test in Europe Conference and Exhibition. IEEE, 2014: 205-209.

[14] 徐佩. 智能卡 AES 加密模块抗侧信道攻击掩码技术研究与实现[D]. 重庆: 重庆大学, 2015:20-37.
XU P. Research and implementation with mask technology on AES encryption module of smartcard against side channel attack[D]. Chongqing: Chongqing University, 2015:20-37.

[15] BHASIN S, BRUNEAU N, DANGER J L, et al. Analysis and improvements of the DPA contest v4 implementation[C]// International Conference on Security, Privacy, and Applied Cryptography Engineering. Springer, 2014: 201-218.

[16] LEUNG Y W, WANG Y. An orthogonal genetic algorithm with quantization for global numerical optimization[J]. IEEE Transactions on Evolutionary Computation, 2002, 5(1): 41-53.

[17] 李濛. 基于FPGA 的AES 算法优化与实现[D]. 哈尔滨:黑龙江大学, 2018:30-50.

LI M. Optimization and implementation of AES algorithms based on FPGA[D]. Harbin: Heilongjiang University, 2018:30-50.

[18] PAMMU A A, CHONG K S, GWEE B H. Secured low power overhead compensator look-up-table (LUT) substitution box (S-Box) architecture[C]// IEEE International Conference on Networking. Springer, 2016: 3-6.

[19] KARTHIGAIKUMAR P, CHRISTY N A, MANGAI N M S. PSP CO2: an efficient hardware architecture for AES algorithm for high throughput[J]. Wireless Personal Communications, 2015, 85(1): 305-323.

[20] 汪鹏君, 郝李鹏, 张跃军. 防御零值功耗攻击的 AES SubByte 模块设计及其 VLSI 实现[J]. 电子学报, 2012, 40(11): 2183-2187.
WANG P J, HAO L P, ZHANG Y J. Design of AES SubByte module of anti-zero value power attack and its VLSI implementation[J]. ACTA Electronica sinica, 2012, 40(11):2183-2187.

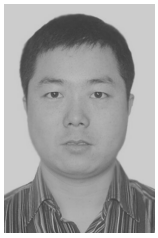
[21] PROUFF E, RIVAIN M, VAN R. Statistical analysis of second order differential power analysis[J]. IEEE Transactions on Computer, 2009: 799-811.

[22] NASSAR M, GUILLEY S, DANGER J L. Formal analysis of the entropy/security trade-off in first-order masking countermeasures against side-channel attacks[C]//International Conference on Cryptology. Springer, 2011:22-39.

[作者简介]

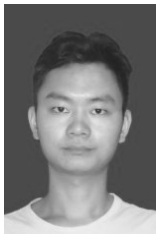
姜久兴（1963− ），男，黑龙江哈尔滨人，博士，哈尔滨理工大学教授、硕士生导师，主要研究方向为集成电路设计。

厚娇（1988− ），女，黑龙江哈尔滨人，哈尔滨理工大学硕士生，主要研究方向为信息安全和集成电路设计。



黄海（1982− ），男，内蒙古巴彦淖尔人，博士，哈尔滨理工大学副教授、硕士生导师，主要研究方向为信息安全、数字信号处理和集成电路设计。

赵玉迎（1990− ），女，黑龙江哈尔滨人，哈尔滨理工大学硕士生，主要研究方向为信息安全和集成电路设计。



冯新新（1991− ），男，江苏淮安人，哈尔滨理工大学硕士生，主要研究方向为计算机网络和信息安全。