ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2009/03(04)-0405-08
DOI:10.3778/j.issn.1673-9418.2009.04.007

E-mail:fcst@public2.bta.net.cn
http://www.ceaj.org
Tel:+86-10-51616056

抗 DPA 攻击的 AES 算法研究与实现 *

郑新建[1+],张翌维[1],彭 波[2],沈绪榜[1]
1. 西安微电子技术研究所,西安 710054
2. 中兴集成电路设计公司,广东 深圳 518057

Research and Implementation of DPA Resistant AES Algorithm*

ZHENG Xinjian[1+], ZHANG Yiwei[1], PENG Bo[2], SHEN Xubang[1]
1. Xi'an Microelectronics Technology Institute, Xi'an 710054, China
2. ZTEIC Corporation, Shenzhen, Guangdong 518057, China
+ Corresponding author: E-mail: addoil_zh@163.com

ZHENG Xinjian, ZHANG Yiwei, PENG Bo, et al. Research and implementation of DPA resistant AES algorithm. Journal of Frontiers of Computer Science and Technology, 2009,3(4):405-412.

Abstract: To improve the DPA (differential power analysis) resistance of a cryptographic device, Mask is used to make the power consumption independent of the intermediate values. High order DPA can attack cryptographic device with simple Masks. A DPA resistant AES (advanced encryption standard) Mask algorithm with several random Masks is proposed. The algorithm is implemented on an 8 bit MCU. The result shows that the DPA resistant AES algorithm can defend DPA and high order DPA analysis efficiently.

Key words: differential power analysis (DPA); Mask; advanced encryption standard (AES); Sbox

摘 要: Mask 技术破坏了加密过程中的功率消耗与加密的中间变量之间的相关性,提高了加密器件的抗 DPA 攻击能力。简单地对算法流程添加 Mask 容易受到高阶 DPA 攻击的。提出了一种对 AES 加密过程的各个操作采用多组随机 Mask 进行屏蔽的方法,并在 8 bit 的 MCU 上实现了该抗攻击的 AES 算法。实验结果表明,添加 Mask 后的抗 DPA 攻击 AES 算法能够有效地抵御 DPA 和高阶 DPA 的攻击。

关键词:差分功耗攻击;掩码技术;高级加密标准;S 盒
文献标识码:A **中图分类号:**TP309

* the National High-Tech Research and Development Plan of China under Grant No.2005AA1Z1080,2007AA012459 (国家高技术研究发展计划(863)).
Received 2008-11, Accepted 2009-04.

1 引言

当前信息安全逐渐成为信息领域的重要研究方向,而加密器件作为信息安全的主要载体,其攻击和防护的研究也是当前信息安全的研究热点。差分功耗攻击[1-2](differential power analysis,DPA)是加密器件攻击的一个主要手段,DPA 攻击是利用加密器件产生的功耗与所处理数据的相关性来获取密钥。DPA 攻击的目的是通过分析大批量的功耗样本曲线来寻找加密器件的密钥信息。DPA 分析不需要攻击者对被攻击器件有深入地了解,它还可以在功耗样本曲线噪声很大的情况下获得密钥信息。

针对加密器件中功耗的泄漏,很多学者提出了许多抗功耗攻击的方法,Mask 技术是一种主要的方法。Mask 技术能够在加密过程中引入与实际处理数据不相关的另外一个变量,从而使加密器件所泄漏的功耗信息与实际处理数据相关性大大降低。文献[3]提出了一种 DES 的抗功耗分析的改进算法,在 DES 的加密过程中引入了 Mask,但是该算法仅仅对 DES 加密过程的一个部分进行了 Mask,其余部分仍然有功耗信息的泄漏。当攻击者选择合适的攻击模型后,仍然可以得到很好的 DPA 攻击效果。而且,仅仅添加一个 Mask 的算法对高阶 DPA 的防御度不够。另外,文献[4-5]等针对电路 cell 级的功耗平均和掩盖方法进行了探讨,但这些方法大多需要较大的硬件代价,也需要特殊的后端工具的支持,不易实现。

本文针对 AES(advanced encryption standard)的加密流程进行研究,提出了对整个加密过程进行 Mask 的方法,并且在算法中应用多个 Mask,有效地抵御了一阶或更高阶的 DPA 分析。首先分析了 AES 的加密过程以及 DPA 对 AES 的攻击。然后对 AES 加密过程的各个操作进行了 Mask 方法的研究,最后利用 8 bit 的 MCU 实现了抗功耗分析多 Mask 的 AES 算法,并分析了其实现代价以及不足之处。

2 DPA 对 AES 加密过程的分析

2.1 AES 的加密过程

AES[6]是美国国家标准技术协会(NIST)在 2001 年发布的高级加密标准。其输入为一个 128 位的分组,该分组被编排为一个称作状态阵列 State 的 4×4 字节矩阵,每一轮加/解密时都对其进行修改。在加/解密过程完成后,该状态阵列转换为 128 位的线性串输出。128 位密钥类似地转换为字节方矩阵,初始输入密钥经过处理后,扩展为 10 组不同的密钥用于 10 轮加/解密。

一轮典型的加密过程如图 1,由 4 个阶段组成:

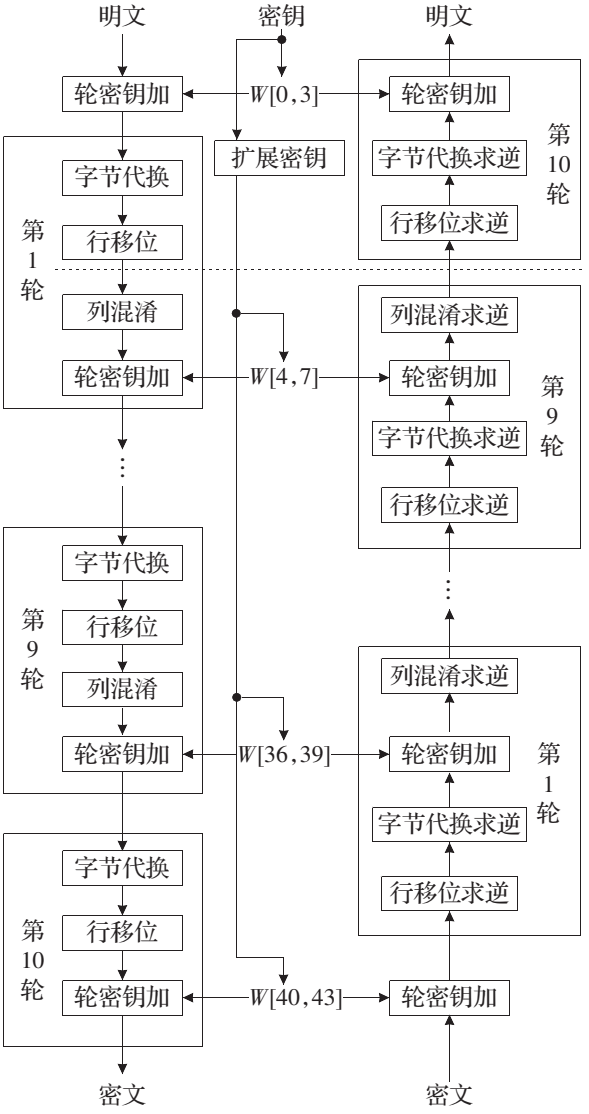

Fig.1 Process of the AES encryption

图 1 AES 加解密过程

(1)字节代换 SubBytes:用一个 S 盒完成分组中的字节代换;

(2)行移位 ShiftRows:一个简单的置换操作;

(3)列混淆 MixColumns:一个利用域 $GF(2^8)$ 上的算术特性的代换;

(4)轮密钥加 AddRoundKey:利用当前分组和扩展密钥的一部分进行按位异或。

对于加/解密操作,算法由轮密钥加开始,接着进行 9 轮迭代运算,每轮都包含所有 4 个阶段的变换,然后执行只包含 3 个阶段的最后一轮运算。

2.2 AES 加密的 DPA 分析

首先选择攻击的中间变量为 S 盒的某一位输出,然后通过硬件的功耗分析平台采集多组随机数据的加密功耗样本。图 2 所示为一轮 AES 的加密过程。进行放大并分析后,可以确定 S 盒操作的执行时间,也就确定了 DPA 攻击的时间。

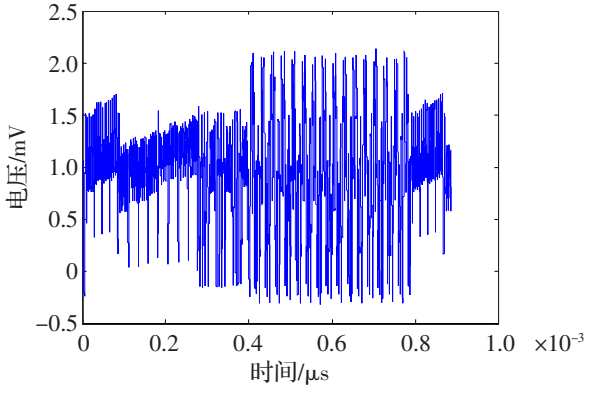

Fig.2 SPA analysis on AES one round

图 2 AES 加密一轮过程的 SPA 分析

在确定了 DPA 分析设定的中间结果相关操作的具体发生位置后,对 S 盒及轮密钥加的操作进行差分功耗分析。首先用 N 组大批量的随机数据进行加密(取 N=10 000),并记录明文和密文,然后应用针对 AES 的 DPA 分析盒函数,对结果进行划分。当划分正确(也就是 8 bit 的子密钥猜测正确时)时差分功耗出现峰值,如图 3 所示。

2.3 高阶 DPA 攻击

高阶 DPA 攻击主要是针对 Mask 技术进行攻击

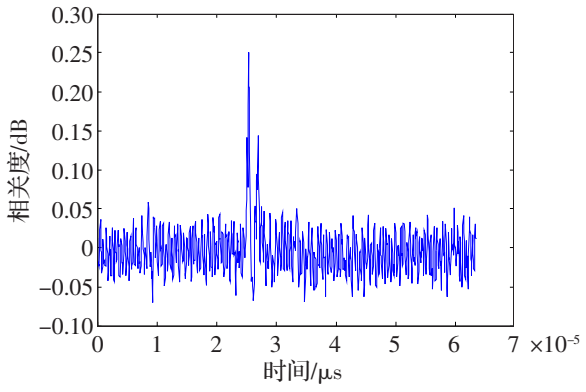

Fig.3 DPA analysis with correct key on AES

图 3 子密钥猜测正确的差分功耗

的。虽然 Mask 技术可以抵御 DPA 攻击,但是如果添加的 Mask 为单一的 Mask 或者添加 Mask 没有完全屏蔽加密过程的中间结果。那么 Mask 后的加密过程仍然是有可能被攻击的。高阶的 DPA 攻击就利用了两个或多个中间变量,可以解掉加密过程中添加的 Mask 从而攻击出算法的秘密信息[7-8]。在 DPA 攻击过程中,首先选择两个中间变量 u 和 v,假设算法仅仅利用了布尔型的 Mask 进行了防护,那么这两个中间变量将会变成 $u_m=u\oplus m$ 和 $v_m=v\oplus m$。此时仅仅对任意一个中间件变量进行 DPA 攻击,都将受到随机变量 m 的影响而得不到真正的密钥。但是,高阶 DPA 攻击可以利用 u 和 v 产生第三个中间变量记为 $w=u\oplus v$ 作为攻击的对象。由于算法仅仅利用了布尔型的 m 进行了防护,所以有:

$$w=u\oplus v=u\oplus m\oplus v\oplus m=u_m\oplus v_m$$

从而变量 w 就被从 Mask 中解了出来,其值就与密码算法实际执行的功耗可以进行相关,可以作为攻击者的攻击中间变量。其余的攻击方法与一阶 DPA 相同。

3 AES 的 Mask 算法

Mask 技术利用一个攻击者不可能获得的随机变量 m 对 DPA 攻击的中间变量 v 进行掩盖,从而得到掩盖后的中间变量 v_m($v_m=v*m$),使攻击者每次获取的功耗信息均是 v_m 所造成的,而且由于 m 是随机变化的,每次加密并不相同,所以攻击者将无法获得中间变量 v 所带来的功率消耗与秘密信息的相关性[9-10]。

由于加密过程中的各个操作均是与数据相关的,所以要求整个加密过程中所有的中间变量都被 Mask 所屏蔽。

典型的 Mask 分为两种:布尔型的 Mask,一般用异或操作实现;算术型的 Mask,一般用模加模乘来实现。异或操作实现无论硬件还是软件的实现代价都是很小的。分组加密算法中一般应用了线性和非线性的函数,对于线性函数 $f(x*y)=f(x)*f(y)$,那么可以利用布尔型的屏蔽电路,其实现代价很小。对于非线性的操作,例如 AES 中的 Sbox 字节代换操作,直接应用布尔型的 Mask 电路是不成立的,因为 $s(x\oplus m)\neq s(x)\oplus s(m)$,而 AES 的 Sbox 是基于有限域的求逆操作,所以可以采用算术模乘的方法添加 Mask,如式(1):

$$f(x\times m)=(x\times m)^{-1}=f(x)\times f(m) \tag{1}$$

另外,抗功耗分析的 Mask 技术也可以应用于电路级别,其中比较常用的有双环预充电逻辑[11]、互补逻辑等,这些实现方法一方面带来了电路面积的急剧扩大,另外也需要特殊的后端工具的支持。

AES 算法主要由字节代换(SBox)操作、行移位(ShiftRow)、列混淆(MixColum)和轮秘钥加(AddRoundKey)等操作组成。下面就各个操作分析 Mask 方法。

3.1 Sbox 的 Mask 方法

Sbox 是 AES 算法中的非线性操作,直接的布尔型 Mask 是不适用的。一般可以将 Sbox 的实现分为两种方式。

软件 Sbox 可以利用查表实现,也就是说对于每个输入变量 v,其输出为表 T 中的某个对应数据。所以可以对这个表 T 加 Mask 来屏蔽 Sbox 操作的中间变量。首先生成一个掩盖码 m,然后通过查原始 T 表中所有可能的 v 生成 $T(v)\oplus m$,并将这些值存储到一个新表 T_m,则有:$T_m(v\oplus m)=T(v)\oplus m$ 最后利用 $v\oplus m$ 进行加密过程中的查表 T_m 操作。查表的结果可以通过一个 m 的 Mask 得到真正的 $T(v)$。

AES 利用硬件实现时,AES 的 Sbox 是 $GF(256)$ 域上的求逆操作,而 $GF(GF256)$ 上的值 v 可以分解为两个 $GF(16)$ 上的值 v_hx+v_l,则,Sbox 操作可以分解如下:

$$(v_hx\oplus v_l)^{-1}=v_h'x\oplus v_l' \tag{2}$$

$$v_h'=v_h\times w' \tag{3}$$

$$v_l'=(v_h\oplus v_l)\times w' \tag{4}$$

$$w'=w^{-1} \tag{5}$$

$$w=(v_h^2\times p0)\oplus(v_h\oplus v_l)\oplus v_l^2 \tag{6}$$

分解过程是一个线性过程,可以添加 Mask 如下:

$$((v_h\oplus m_h)x\oplus(v_l\oplus m_l))^{-1}=(v_h'\oplus m_h')x\oplus(v_l'\oplus m_l') \tag{7}$$

然后需要对模乘和模加做 Mask,这样上面的求逆过程变成如下:

$$v_h'\oplus m_h'=v_h\times w'\oplus m_h' \tag{8}$$

$$v_l'\oplus m_l'=(v_h\oplus v_l)\times w'\oplus m_l' \tag{9}$$

$$w'\oplus m_w'=w^{-1}\oplus m_w' \tag{10}$$

$$w\oplus m_w=(v_h^2\times p0)\oplus(v_h\oplus v_l)\oplus v_l^2\oplus m_w \tag{11}$$

这样 $GF(256)$ 域的求逆操作,就变成了公式(10)中的 $GF(16)$ 域上的求逆 w^{-1} 操作。同样可以再将 $GF(16)$ 域上的求逆操作分解为 $GF(4)$ 上的操作,而,$GF(4)$ 上的求逆操作就是平方操作,即

$$x^{-1}=x^2 \tag{12}$$

所以有:

$$(x\oplus m)^{-1}=(x\oplus m)^2=x^2\oplus m^2 \tag{13}$$

这样就得到了完全 Mask 的 Sbox,不过这种硬件实现方式带来的是资源的增加,一个 Mask 后的 Sbox 实现需要 9 个乘法操作[12],而原始的 Sbox 仅仅需要 3 个乘法操作[13]。另外硬件的 Mask 电路还会带来硬件关键路径变长,文献[14]指出,带 Mask 电路的 AES 硬件会比不带 Mask 的 AES 速度慢 2~3 倍。所以仍然采用查表法进行设计。

3.2 MixColum 的 Mask 方法

MixColum 操作是线性操作,可以添加布尔型的 Mask。AES 标准中 MixColum 操作定义如下式:

$$\begin{aligned}
s_{0,c}' &= (\{02\}\bullet s_{0,c})\oplus(\{03\}\bullet s_{1,c})\oplus s_{2,c}\bullet s_{3,c}\\
s_{1,c}' &= s_{0,c}\oplus(\{02\}\bullet s_{1,c})\oplus(\{03\}\bullet s_{2,c})\bullet s_{3,c}\\
s_{2,c}' &= s_{0,c}\oplus s_{1,c}\oplus(\{02\}\bullet s_{2,c})\oplus(\{03\}\bullet s_{3,c})\\
s_{3,c}' &= (\{03\}\bullet s_{0,c})\oplus s_{1,c}\oplus s_{2,c}\oplus(\{02\}\bullet s_{3,c})
\end{aligned} \tag{14}$$

因此,对于 MixColum 操作,仅仅一个布尔型 Mask 是无效的,如果所有变量 S_{ij} 都为 $S_{ij}\oplus m$,那么从上面公式可以看出,S_{ij}' 相当于没有加罩。所以对 MixColum 的 Mask 至少要有两组不同的 Mask,最简单的方法对 State 的每一行添加不同的 Mask(m_1,m_2,m_3,m_4),则得到 MixColum 的屏蔽算法,如公式(15):

$$\begin{aligned}
s_{0,c}' &= (\{02\}\bullet s_{0,c}\oplus m_1)\oplus(\{03\}\bullet s_{1,c}\oplus m_2)\oplus\\
&\quad (s_{2,c}\oplus m_3)\oplus(s_{3,c}\oplus m_4)\\
s_{1,c}' &= (s_{0,c}\oplus m_1)\oplus(\{02\}\bullet s_{1,c}\oplus m_2)\oplus\\
&\quad (\{03\}\bullet s_{2,c}\oplus m_3)\oplus(s_{3,c}\oplus m_4)\\
s_{2,c}' &= (s_{0,c}\oplus m_1)\oplus(s_{1,c}\oplus m_2)\oplus\\
&\quad (\{02\}\bullet s_{2,c}\oplus m_3)\oplus(\{03\}\bullet s_{3,c}\oplus m_4)\\
s_{3,c}' &= (\{03\}\bullet s_{0,c}\oplus m_1)\oplus(s_{1,c}\oplus m_2)\oplus\\
&\quad (s_{2,c}\oplus m_3)\oplus(\{02\}\bullet s_{3,c}\oplus m_4)
\end{aligned} \tag{15}$$

这样就保证了整个MixColum 操作的中间变量均被 Mask 所屏蔽。

3.3 其他操作的 Mask 方法

AES 算法流程中除 Sbox 操作和 MixColum 操作外,还有 AddRoundKey 和 ShiftRow 操作,分析如下:

(1)AddRoundKey

轮秘钥加操作为线性操作,可以直接用布尔型 Mask 进行屏蔽,但是由于加 Mask 的 AES 算法流程保证对产生的轮秘钥进行过布尔型的 Mask 屏蔽,也就是说实际的轮秘钥加操作为:$d\oplus(k\oplus m)$,相当于 $(d\oplus k)\oplus m$,也就是说对轮秘钥加实现算法进行了布尔型的 Mask,所以此处不需要再添加新的 Mask。

(2)ShiftRows

AES 的移位操作是紧跟在 Sbox 操作之后的,其操作是按行进行相应 byte 位置的移位,由于在进行移位操作之前所有的中间变量已经进行了 Mask,所以移位操作无需再添加新的 Mask。

4 Mask 算法的实现及分析

DPA 攻击者可能利用 AES 加密过程中任何的中间变量所泄露的功耗信息,所以有必要对整个加密过程进行 Mask,也就是说对开始输入的明文和轮秘钥进行 Mask,仅仅在最后一轮加密结束时解开此 Mask 即可。

整个加密过程中 Mask 的情况如图 4 所示。

加密过程中一轮的输入包括明文和轮秘钥,明文为上一轮的 MixColum 的输出,也就是带 m_1',m_2',m_3',m_4'掩盖的数据,秘钥为带 $m_1'\oplus m$,$m_2'\oplus m$,$m_3'\oplus m$,$m_4'\oplus m$ 掩盖的数据。

经过 AddRoundKey 操作后 Mask 变为 m,然后通过查表 T_m(预计算表格),得到的结果 Mask 变为 m',ShiftRows 操作并不改变 AES 的 Mask,在 MixColumn 执行之前,将 Mask 更改为按行不同的 m_1,m_2,m_3,m_4,执行 MicColumn 操作后 Mask 变为 m_1',m_2',m_3',m_4',然后被输入到下一轮,进行下一轮的加密过程。

由于行移位操作没有引入新的 Mask,所以最后一轮结束后得到的就是仅带 Mask 屏蔽 m 的密文,然后对其进行解 Mask 操作得到真正的密文。

仍然采用第 1 章中 8 bit 的 MCU 来实现加 Mask 后的 AES,应用相同的差分功耗分析,结果如图 5。对比图 3 和图 5 可以看出,应用了整体加罩方法的 AES 加密算法其 DPA 攻击的相关度从 0.25 降低到 0.1 以下,其抗 DPA 分析特性有了很大提高。

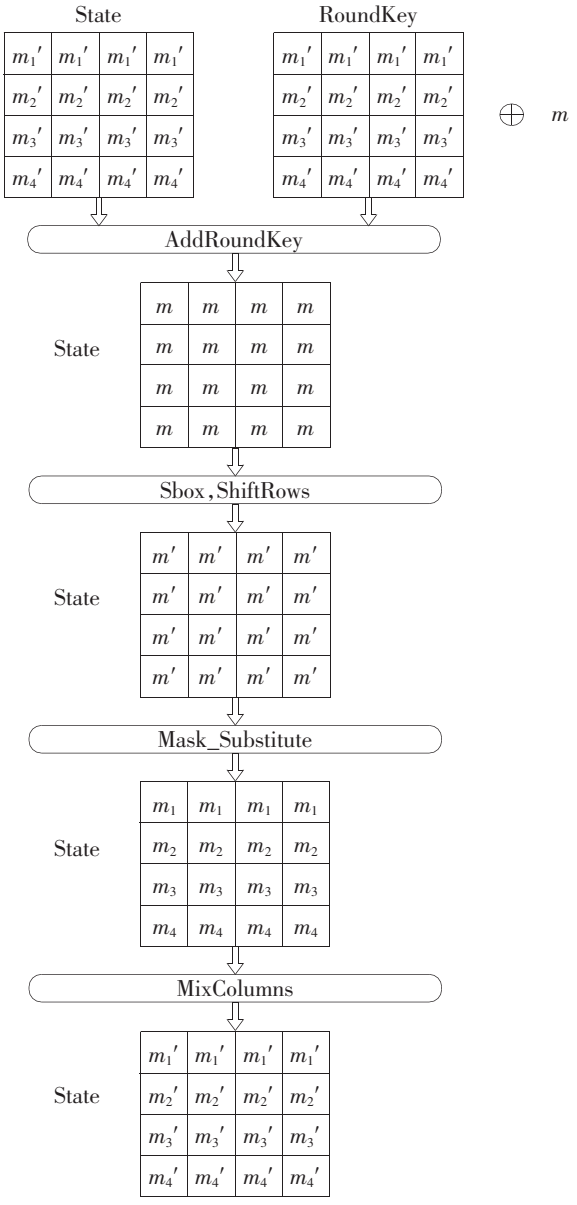

Fig.4 AES encryption with Mask

图 4 加密过程中的 Mask

5 结束语

Mask 技术能够在加密过程中引入与实际处理数据不相关的另外一个随机变量,从而使加密器件所泄漏的功耗信息与实际处理数据相关性大大降低。通常的 Mask 技术仅针对加密过程中的某一个中间结果[3],当攻击者选用合适的模型时,防护将会失效。而且采用一组随机数据对加密过程进行 Mask 对二阶或者更高阶的 DPA 攻击是无效的[7]。本文针对 DPA 攻击

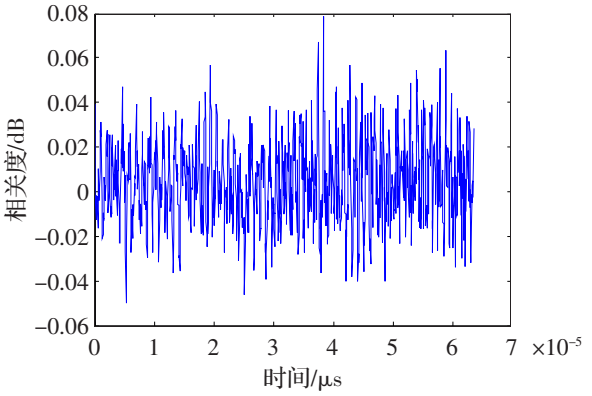

Fig.5 DPA analysis with correct key on the Masked AES

图 5 添加 Mask 后 AES 子密钥猜测正确时的 DPA 结果

的过程,对 AES 整个加密过程进行多组 Mask 同时屏蔽所有操作中间变量,并且由于采用的 Mask 既包括线性的轮密钥加等操作,又包括非线性的 Sbox 操作,所以攻击者很难利用选择两个或多个中间变量来解除 Mask 的高阶 DPA 攻击进行攻击,能够有效地抵御一阶及高阶的 DPA 分析,与文献[3-4]等方法相比有更高的安全特性。

这种全流程多 Mask 算法也存在不足之处[15],首先,算法执行效率较低。增加 Mask 后的 AES 算法的实现性能比原始 AES 算法有所下降,其下降原因主要是需要预计算 Sbox 表格 T_m。在进行加密之前需要进行预计算 Sbox 掩盖后的新的存储表 T_m,T_m 的计算需要遍历各个不同的 Sbox 的输入,并产生一个新的 Sbox 表,在同样的 8 bit 的 MCU 平台上实现,其性能降低为原来的约 65%。其次,当算法实现的硬件 MCU 会泄露所处理数据的汉明权重时,攻击者仍有可能利用基于模板的 DPA 方法进行攻击[16-17]。

References:

[1] Kocher P C, Jaffe J, Jun B. Differential power analysis[C]// the 19th Annual International Cryptology Conference, Santa Barbara, California, USA, Aug, 1999:388-397.

[2] Messerges T S, Dabbish E A, Sloan R H. Investigations of

power analysis attacks on smartcards[C]//USENIX Workshop on Smartcard Technology, May, 1999:151-162.

[3] Jiang Huiping, Mao Zhigang. Advanced DES algorithm against differential power analysis and its hardware implementation[J]. Chinese Journal of Computers, 2004,27(3):334-338.

[4] Han Jun, Zeng Xiaoyang, Tang Ting'ao. VLSI design of anti attack DES circuits[J]. Chinese Journal of Semiconductors, 2005:1646-1652.

[5] Tiri K, Verbauwhede I. A digital design flow for secure integrated circuits[J]. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 2006,25(7): 1197-1208.

[6] National Institute of Standards and Technology. FIPS 197 Advanced Encryption Standard[S]. 2001-11.

[7] Akkar M L, Goubin L. A generic protection against high-order differential power analysis[C]//LNCS2887:FSE2003, 2003: 192-205.

[8] Joye M, Paillier P, Schoenmakers B. On second order differential power analysis[C]//LNCS 3659: Cryptographic Hardware and Embedded Systems (CHES 2005), 2005: 293-308.

[9] Itoh K, Takenaka M, Torii N. DPA countermeasure based on the masking method[C]//LNCS 2288: Proceedings of the 4th International Conference Seoul on Information Security and Cryptology, 2002:440-456.

[10] Saputra H. Masking the energy behavior of DES encryption[C]// Proceedings of DATE 2003, 2003:253-255.

[11] Pop T, Mangard S. Masked dual-rail pre-charge logic: DPA resistance without routing constraints[C]//LNCS 3685: Cryptographic Hardware and Embedded Systems (CHES 2005), 2005:172-186.

[12] Oswald E, Mangard S, Pramstaller N, et al. A side channel analysis resistant description of the AES sbox[C]//LNCS 3557: 12th International Workshop, FSE2005, 2005:413-423.

[13] Wolkerstorfer J, Oswald E, Lamberger M. An ASIC implementation of the AES Sboxes[C]//LNCS 2271: CT-RSA 2003, 2003:67-78.

[14] Pramstaller N, Oswald E, Mangard S, et al. A masked AES ASIC implementation[C]//Austrochip 2004,Villach, Austria, Oct, 2004:77-82.

[15] Herbst C, Oswald E, Mangard S. An AES smart card implementation resistant to power analysis attacks[C]//LNCS 3989: ACNS2006, 2006:239-252.

[16] Agrawal D, Rao J R, Rohatgi P, et al. Templates as master keys[C]//LNCS 3659: Cryptographic Hardware and Embedded Systems (CHES 2005), 2005:15-29.

[17] Peeters E, Standaert F X, Doncker N, et al. Improved higer-order side-channel attacks with FPGA experiments[C]//LNCS 3659: Cryptographic Hardware and Embedded Systems (CHES 2005), 2005:303-329.

附中文参考文献:

[3] 蒋惠萍,毛志刚.一种抗差分功耗攻击的改进 DES 算法及其硬件实现[J].计算机学报,2004,27(3):334-338.

[4] 韩军,曾晓洋,汤庭鳌.DES 密码电路的抗差分功耗分析设计[J].半导体学报,2005:1646-1652.

ZHENG Xinjian was born in 1980. He received his M.S. degree in Computer Architecture from Xi'an Microelectronics Technology Institute in 2005. Now he is a Ph.D. candidate at Xi'an Microelectronics Technology Institute. His research interests include computer architecture, analysis and defense of encryption chips, etc.

郑新建(1980-),男,山东淄博人,2005 年于西安微电子技术研究所获硕士学位,现为西安微电子技术研究所博士研究生,主要研究领域为计算机系统结构,密码算法芯片的攻击与防御等。

ZHANG Yiwei was born in 1980. He received his M.S. degree in Computer Architecture from Xi'an University of Technology in 2005. Now he is a Ph.D. candidate at Xi'an Microelectronics Technology Institute. His research interests include computer architecture, analysis and defense of encryption chips, etc.

张翌维(1980-),男,陕西西安人,2005 年于西安理工大学获硕士学位,现为西安微电子技术研究所博士研究生,主要研究领域为计算机系统结构,密码算法芯片的攻击与防御等。

PENG Bo was born in 1975. He received his Ph.D. degree in Computer Architecture from South China University of Technology in 2001. Now he is a senior engineer at ZTEIC Corporation. His research interests include encryption algorithm and application, etc.

彭波(1975-),男,江西萍乡人,2001 年于华南理工大学获博士学位,现为中兴集成电路设计公司高级工程师,主要研究领域为密码算法及其应用等。

SHEN Xubang was born in 1933. He graduated from Beijing University in 1957. Now he is a doctoral supervisor at Xi'an Microelectronics Technology Institute, member of the Chinese Academy of Sciences. His research interests include computer architecture, VLSI, etc.

沈绪榜(1933-),男,湖南临澧人,1957 年毕业于北京大学,现为西安微电子技术研究所博士生导师,中国科学院院士,主要研究领域为计算机系统结构,超大规模集成电路设计等。