

引用格式:朱文锋,郭箏.抗二阶 DPA 分析的 ZUC 算法防护方案及硬件实现[J].微电子学与计算机,2019,36(9):44-49.
[ZHU W F, GUO Z. ZUC algorithm protection scheme and hardware implementation of against second-order DPA analysis [J]. Microelectronics & Computer, 2019, 36(9): 44-49.]

抗二阶 DPA 分析的 ZUC 算法防护方案及硬件实现

朱文锋¹, 郭 箏^{1,2}

(1 上海交通大学 微电子学院, 上海 200010; 2 智巡密码(上海)检测技术有限公司, 上海 200010)

摘 要: 对于不带防护的 ZUC 算法硬件实现, 容易 DPA 攻击的影响. 为此提出了基于二阶算术加、有限域 sbox 二阶全掩码、sbox 固定掩码以及伪轮防护方案的 ZUC 算法防护方案, 理论上可以抗二阶 DPA 攻击, 在 FPGA 上对其进行实现, 并在硬件实现进行了一定的优化, 节省了功耗和面积. 我们通过 FPGA 功耗采集平台, 采集带防护的 ZUC 算法硬件实现的功耗曲线, 对其进行 DPA 攻击, 没有攻击出正确密钥, 表明我们的防护方案实际有效, 大大增加了功耗分析攻击的难度.

关键词: 祖冲之算法; 二阶 DPA; 掩码方案; 硬件实现

中图分类号: TN918.4

文献标识码: A

文章编号: 1000-7180(2019)09-0044-06

ZUC algorithm protection scheme and hardware implementation of against second-order DPA analysis

ZHU Wen-feng¹, GUO Zheng^{1,2}

(1 Microelectronics Institute, Shanghai Jiaotong University, Shanghai 20010, China;

2 Zhixing Password (Shanghai) Testing Technology Co., Ltd, Shanghai 200010, China)

Abstract: For the ZUC algorithm hardware implementation without protection, it is easy to influence the DPA attack. To this end, we propose a ZUC algorithm protection scheme based on second-order arithmetic addition, finite-domain sbox second-order full mask, sbox fixed mask and pseudo-wheel protection scheme, which can theoretically resist second-order DPA attacks. The ZUC algorithm protection scheme is implemented on the FPGA, and the hardware implementation is optimized to save power and area. We use the FPGA power consumption acquisition platform to collect the power consumption curve of the protected ZUC algorithm hardware, and perform DPA attacks on it. No correct key is attacked, indicating that our protection scheme is practical and effective, greatly increasing the power analysis attack difficulty.

Key words: ZUC algorithm; second order DPA; masking scheme; hardware implementation

1 引言

ZUC 算法, 即祖冲之算法, 是 3GPP 机密性算法 EEA3 和完整性算法 EIA3 的核心, 是中国自主设计的加密算法, 于 2011 年 9 月成为 3GPP LTE 第三套加密标准核心算法^[1].

经过长时间的评估程序, ZUC 算法被认为足够强大, 可以抵御许多现在的密码算法的安全性分析方法, 如弱密钥分析, 猜测决定攻击, 代数攻击, 时间攻击等^[2]. 然而和分组密码一样, 该算法容易受到侧信道攻击的影响. 由于侧信道攻击给分组密码算法带来的严重威胁, 学术界做出了很多努力来制定分组密码

的相应对策,所有这些对策都应该能应用在流密码上^[3].文献[4]对 ZUC 进行了差分功耗攻击,在5 000条功耗曲线的情況下攻击出了部分子密钥,表明 ZUC 在抵抗 DPA 攻击方面有明显的安全漏洞.

文献[4]同时提出了一些防护方法,包括随机延时策略和波动动态互补逻辑(Wave Dynamic Differential Logic, WDDL),使 ZUC 能够抵御一阶 DPA 分析.但是,随机延时只能增加 DPA 攻击出正确密钥所需曲线的数目,并不能完全抵抗 DPA 攻击^[5],波动动态互补逻辑不能抵抗汉明距离模型的 DPA 攻击,而掩码技术是目前最广泛使用的功耗分析对策.

本文在 FPGA 上实现了抗二阶 DPA 分析的祖冲之算法硬件实现,包括模加、模乘、sbox 有限域二阶全掩码方案,伪轮防护方案,增大了 ZUC 算法抵抗 DPA 攻击的能力,是第一个包括掩码防护方案的 ZUC 算法硬件实现.最后基于 FPGA 功耗采集工具采集 10 W 条功耗曲线,对功耗曲线进行一阶 DPA 攻击,没有攻击出正确密钥,表明防护方案可以抵御一阶 DPA 攻击.

2 ZUC 算法及对 ZUC 算法的 DPA 攻击

ZUC 是一个同步流密码算法,由 128 位初始密钥 k 和 128 位的初始向量 iv 作为输入,输出 32 位字的密钥流(每 32 位称为一个密钥字).128-EEA3 算法和 128-EIA3 算法定义于 ZUC 算法之上,EEA3 算法思想是将明文直接与 ZUC 产生的密码流进行异或实现加密,EIA3 算法的思想是根据明文以比特为单位进行判断,将 ZUC 产生的密码流做迭代运算生成 MAC 值.

2.1 ZUC 算法流程

ZUC 算法在逻辑上采用 3 层结构设计:线性反馈以为寄存器 LFSR、比特重组(BitReorganization)和非线性函数 $F(X_0, X_1, X_2)$,具体如图 1 所示^[6].

ZUC 算法有两种运行模式:初始化模式和工作模式.初始化模式一般是在密码算法开始的时候运行的,目的是对初始的输入进行混淆;工作模式是在初始化模式运行一段时间之后进行的正常输出密文流的过程.

2.2 针对 ZUC 的 DPA 攻击

Joseph Lanoet 指出,在实际应用中,为了保持发送方和接收方的同步,流密码通常需要经常重新同步,因此,我们可以将初始值视为 DPA 中的明文,同时将生成的密钥流视为密文^[6],进行 DPA 攻击.

文献[4]中介绍了对 ZUC 的 DPA 攻击方法,可

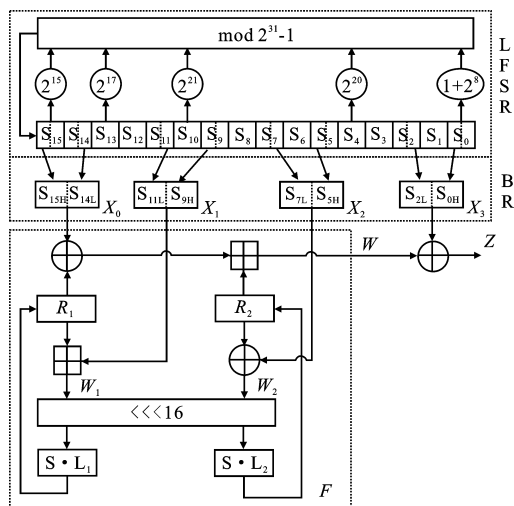


图 1 ZUC 算法流程图

以利用 DPA 攻击出密钥信息. 攻击流程如下:

- (1) 采集功耗曲线;
- (2) 在第一轮初始化阶段利用 DPA 来恢复初始状态 k_9 和 k_5 ;
- (3) 利用 R_1 、 R_2 的值以及在第一轮中到处的 16 个 LFSR 单元的状态用于初始化阶段的第二轮, 恢复 k_{10} 和 k_6 ;
- (4) 在接下来的几轮中使用相同的策略, 恢复 k_{11} 和 k_7 、 k_{12} 和 k_8 、 k_{13} 和 k_9 ;
- (5) 利用穷举和 DPA 攻击攻击出剩下的密钥.

3 ZUC 算法防护方案设计

功耗分析依据的主要原理是密码设备的瞬时功耗跟执行的操作和处理的数据有一定的相关性^[7]. 因此我们只要降低或者消除这种依赖关系, 功耗分析的效率就会大大降低.

随机伪轮的目标是使攻击者无法通过观察功耗曲线轮廓确定当前时刻 ZUC 算法所进行的操作, 而掩码技术通过加入随机数作为掩码, 使 ZUC 算法硬件电路执行过程中的中间值随机化, 从而让能量消耗与真实中间值无关. 本文提出的防护方案结合了伪轮和掩码方案, 能降低真实中间值和实际功耗之间的相关性, 从而使 DPA 攻击变得困难.

3.1 总体设计

首先在初始化阶段把最开始的状态 $S_0 \cdots S_{15}$ 分解成三个随机数 $S_0^0 \cdots S_{15}^0$ 和 $S_0^1 \cdots S_{15}^1$ 和 $S_0^2 \cdots S_{15}^2$, 并满足 $S_0^0 \cdots S_{15}^0 \oplus S_0^1 \cdots S_{15}^1 \oplus S_0^2 \cdots S_{15}^2 = S_0 \cdots S_{15}$, 如图 2 所示. 我们将两个 496 bit 的随机数作为 $S_0^0 \cdots S_{15}^0$ 和 $S_0^1 \cdots S_{15}^1$, 而 $S_0^2 \cdots S_{15}^2$ 通过计算获得.

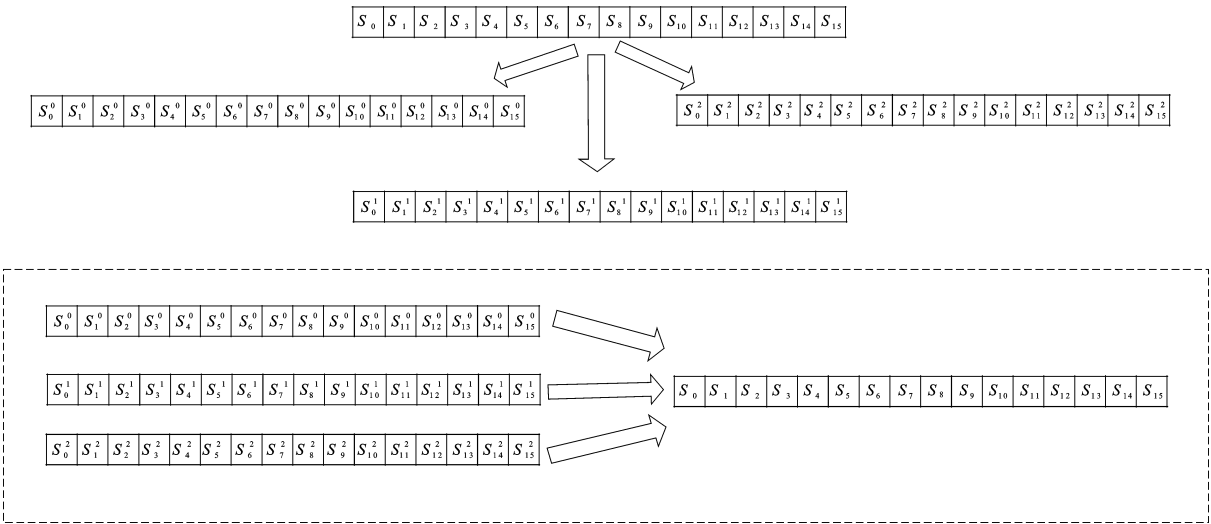


图 2 初始状态分解

图 2 中的三个状态 $S_0^0 \cdots S_{15}^0$ 和 $S_0^1 \cdots S_{15}^1$ 和 $S_0^2 \cdots S_{15}^2$ 分别作为图 3 中 3 个模块的原始输入. 线性部分由输入中间值的一个状态生成输出中间值的

一个状态,而非线性部分需要同时利用中间值的三个状态来生成输出中间值的三个状态到三个模块中.

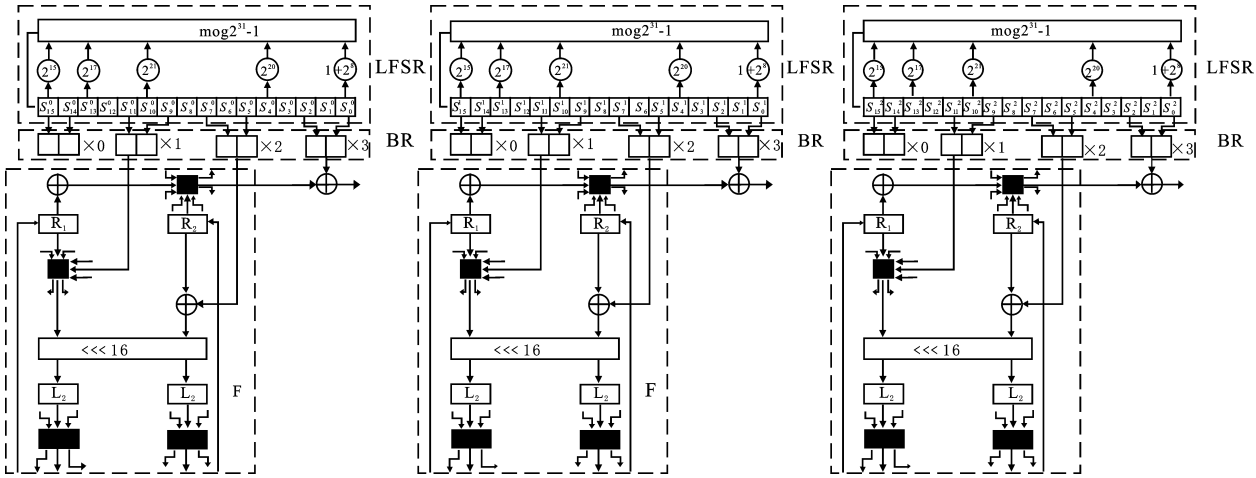


图 3 ZUC 算法掩码的结构

掩码每一轮都更新,计算得到的 S_{15} 都需要异或上新的随机数,达到更新掩码的目的.

在掩码的基础上添加伪轮操作,进一步增加防护方案的安全性.

3.2 非线性模块设计

非线性掩码部分更容易泄露中间值,我们针对非线性掩码部分增加了一些特定的设计方法来进一步增强 ZUC 算法的防护能力.

3.2.1 有限域 s 盒

祖冲之算法的两个 S 盒中. 其中一个有标准的数字表达式,其基本与 AES 类似,如公式(1)所示:

$$\text{Sbox}(x) = M \cdot x^{-1} + B \quad (1)$$

由于矩阵乘法为线性运算,其核心运算为二元

扩域上的求逆,在该表示下可实现 S 盒二阶有限域全掩码,求逆可由公式(2)表示:

$$x^{-1} = x^{254} \bmod x^8 + x^7 + x^3 + x + 1 \quad (2)$$

其具体实现可将求逆操作转化为若干模乘、模平方以及异或运算,其中非线性操作只有模乘,而模平方及异或运算在二元扩域上为线性操作,因此掩码方案核心是在非线性模乘运算.

3.2.2 有限域模乘全掩码

参照有限域模乘分配律,二阶模乘掩码方案如算法 1 所示,其中 a_0, a_1, a_2 为真实乘数 a 的三个状态, b_0, b_1, b_2 为真实被乘数 b 的三个状态.

Algorithm 1GF(2n) 域上 2 阶模乘运算

$$\text{Require: } a_0 \oplus a_1 \oplus a_2 = a, b_0 \oplus b_1 \oplus b_2 = b,$$

Ensure: $c_0 \oplus c_1 \oplus c_2 = a \cdot b$,

```

1: for  $i = 0$  to  $2$  do
2:    $c_0 \leftarrow 0$ 
3:   for  $j = 0$  to  $2$  do
4:      $c_0 \leftarrow c_0 \oplus (a_i \cdot b_j)$ 
5:   end for
6: end for

```

3.2.3 固定掩码 S 盒

祖冲之算法的两个 S 盒中, 另外一个不能用标准的数学表达式表示, 采用固定掩码 S 盒方案, ZUC 的 S 盒记为 $\text{Sbox}(a) = b$, 而带掩码的 S 盒记为 $\text{Sbox}_1(a \oplus m_i) \oplus m_i = \text{Sbox}(a)$, 其中 m_i 为输入掩码, 同时也是输出掩码。八套掩码 S 盒是预计算好的, 在实现时通过三个 bit 值来随机选择哪个 S 盒。输入的三个状态为 X_1, X_2, X_3 , 输出的三个状态为 Z_1, Z_2, Z_3 , mask 为输入的随机数, 方案流程如图 4 所示。

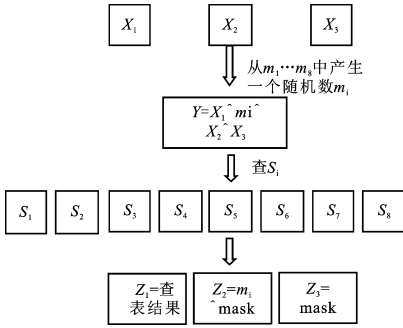


图4 固定掩码方案

3.2.4 算术加全掩码

参照算术加高阶掩码方案, 整体方案算法 2 所示, 其中 k 是指比特个数, 本文中为 32. 其中 SecAnd 是值带掩码的逐比特与操作, 为单比特操作, 方案如图 5.

Algorithm 2 算术加掩码方案

Input: (x_i) and (y_i) for $0 \leq i \leq 2$

Output: (z_i) for $0 \leq i \leq 2$,

with $z_0 \oplus z_1 \oplus z_2 = x_0 \oplus x_1 \oplus x_2 + y_0 \oplus y_1 \oplus y_2$

```

1:  $(w_i)_{0 \leq i \leq 2} \leftarrow \text{SecAnd}((x_i)_{0 \leq i \leq 2}, (y_i)_{0 \leq i \leq 2})$ 
2:  $(u_i)_{0 \leq i \leq 2} \leftarrow 0$ 
3:  $(a_i)_{0 \leq i \leq 2} \leftarrow (x_i)_{0 \leq i \leq 2} \leftarrow (y_i)_{0 \leq i \leq 2}$ 
4: for  $j = 1$  to  $k - 1$  do
5:    $(ua_i)_{0 \leq i \leq 2} \leftarrow \text{SecAnd}((u_i)_{0 \leq i \leq 2}, (u_i)_{0 \leq i \leq 2})$ 
6:    $(u_i)_{0 \leq i \leq 2} \leftarrow (ua_i)_{1 \leq i \leq n} \oplus (w_i)_{1 \leq i \leq n}$ 
7:    $(u_i)_{1 \leq i \leq n} \leftarrow 2(u_i)_{1 \leq i \leq n}$ 
8: end for
9:  $(z_i)_{0 \leq i \leq 2} \leftarrow (x_i)_{0 \leq i \leq 2} \oplus (y_i)_{0 \leq i \leq 2} \oplus (u_i)_{0 \leq i \leq 2}$ 

```

10: return $(z_i)_{0 \leq i \leq 2}$

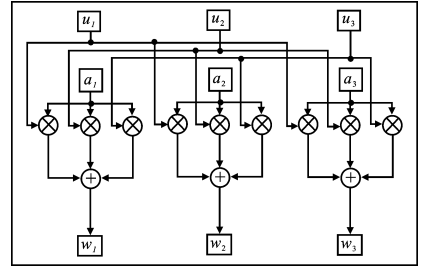


图5 逐比特与逻辑图

3.2.5 伪轮

伪轮是在初始化模式时, 在每一轮真正的轮操作之前随机添加 X 个伪轮, 且伪轮的轮廓和真实轮一模一样, X 的取值在 1~16 之间, 从而无法确定真正的每一轮在哪。

4 硬件实现

由于添加了伪轮, 且实际实现中, 任何一个中间值都要拆分成三个随机数表示, 实现难度较大. 我们基于 Spartan-6 LX75 型号的 sakura-g 功耗采集专用的 FPGA 开发板, 用 Verilog 语言硬件实现了带防护的 ZUC 算法. 且能在 FPGA 上正常采集功耗曲线和明密文信息。

4.1 系统框图

带防护的 ZUC 算法硬件实现的系统框图如图 6 所示, 其中 bitReconstruction 为比特重组模块, F 为 F 函数模块, Control 模块控制各个模块使其正常工作。

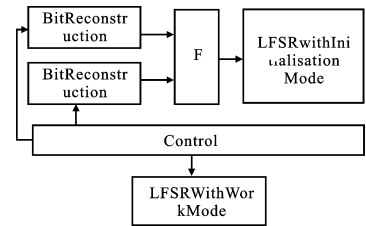


图6 总体结构图

4.2 硬件实现与优化

对带防护的 ZUC 算法进行了硬件实现, 根据功能分别实现了比特重组、算术加、线性变换函数 L_1 和 L_2 、固定掩码 sbox、有限域实现 sbox 等这些不带时序功能的模块. LFSR 模块每一轮用到 6 次模加操作, 而模加模块功耗较大, 因此将每一轮拆分为 8 个时钟周期进行, 并将功耗较大的固定掩码 sbox 和有限域实现 sbox 放在这 8 个时钟周期的不同周期执行, 从而降低了功耗和面积。

通过时序控制逻辑,产生伪轮操作,并保证各个模块协调正常运行.

本文对硬件实现采取了一些优化策略.

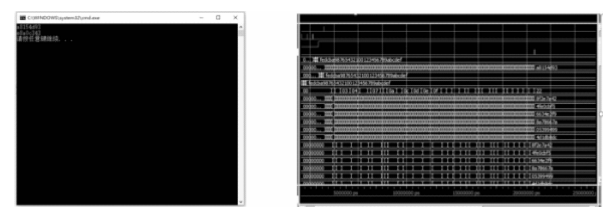
(1)采用了分时复用的策略,将一轮的计算分散在 8 个时钟周期,将 S 盒的面积减小到原来的 1/2,将模加模块的面积和瞬时功耗降低到原来的 1/6.

(2)通过从外部输入随机数,降低了电路产生随机数的硬件开销.

(3)复用部分寄存器,减小面积.

4.3 仿真和 FPGA 实际运行结果

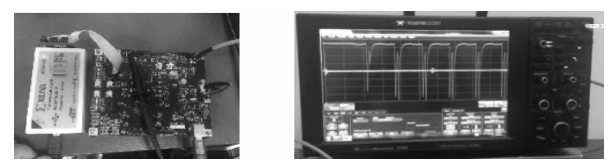
给出初始密钥和初始向量,仿真结果如图 7 所示,其中不带防护的软件实现输出结果如下: 0xa8154d93,与硬件代码仿真结果一致,且伪轮能正常发挥功能.能说明 ZUC 防护方案设计无逻辑错误.



(a)软件仿真测试结果 (b)modelsim 仿真结果

图 7 软件仿真测试结果和 modelsim 仿真结果

且算法能够在 FPGA 上实现,并能正确采集功耗曲线,密文信息也正确,采集环境和功耗波形如图 8 所示.



(a)采集环境 (b)功耗波形

图 8 采集环境和功耗波形图

5 实验结果与分析

5.1 综合结果

我们通过 Verilog 实现的带防护的 ZUC 算法,最终 FPGA 综合得到具体的实验数据如表 1 所示.功耗和面积整体整体较优.

表 1 综合后的实验数据表

FPGA 型号	硬件占 用资源 (ALUTS)	寄存 器数	工作主频 /MHz	功耗 估计 /W
Spartan-6	8 278	4 209	3	0.081

5.2 防护方案效果分析

高阶 DPA 利用了某种联合泄露,该联合泄露基于出现在密码设备中的多个中间值,二阶掩码利用与同一个掩码相关的两个中间值的联合信息泄露.

对于一个中间值,其由三个随机数表示且与算法中的其它变量均独立,随机数均匀,则该中间值可以抵御二阶 DPA 攻击[8]. ZUC 算法防护方案中,中间值均由 3 个随机状态表示,且算术加、摸乘、有限域 S 盒均为 2 阶全掩码,则这些部分理论上可以抵御二阶 DPA 的攻击.

固定掩码方案,不能在理论上抵抗旁路分析,但是会大大增加攻击者的攻击难度,具有很好的实际效果.

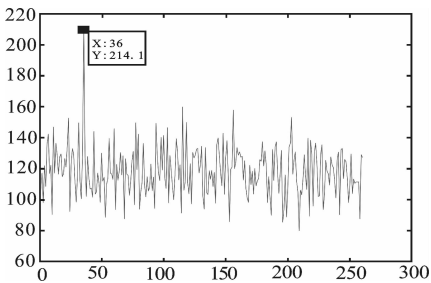
而伪轮防护方案使攻击者每次得到一定错误的旁路信息,即每次只有 1/16 的可能得到正确的旁路信息,根据相关理论[9],采用伪轮防护方案后,攻击成功所需的曲线条数是未加防护的 $\frac{1}{162}$ 倍.

5.3 实际攻击结果

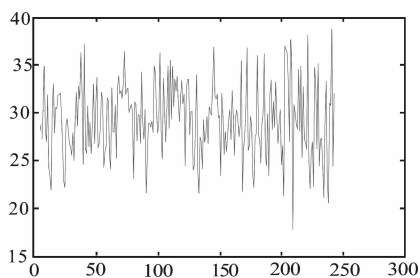
基于 FPGA 功耗采集平台,我们对本文实现的带防护的 ZUC 算法采集实际的功耗曲线,每一条曲线对应一个随机的初始向量和相应的密文.

参照文献[4]中 DPA 攻击方法,对不带防护的 ZUC 算法,在第一轮初始化阶段利用 DPA 来恢复密钥 k_9 ,我们采用 T-test[10]的 DPA 攻击方法,在 10W 条曲线的情况下,攻击的结果如图 9(a)所示:正确的密钥(0x23,十进制数为 35)的密钥相关性最高.证明了该攻击方法实际有效.选用同样的攻击方法对带防护的 ZUC 硬件实现的功耗曲线进行攻击,位置同样选取第一轮初始化过程中 R_1 、 R_2 寄存器赋值的时刻.在 10 W 条曲线的情况下,攻击结果如图 9(b)所示:各个密钥之间相关性差异不大,且正确密钥相关性较低.

所以可以得出结论,本文的防护方案可以抵御一阶 DPA 攻击,且增加了 DPA 攻击的难度.



(a)不带防护攻击结果



(b)带防护的攻击结果

图9 加防护前后攻击结果对比

6 结束语

针对不带防护的 ZUC 算法容易受到 DPA 攻击的情况,参照分组密码算法中常用且实际效果较好的防护方案,我们提出了抗二阶 DPA 分析的 ZUC 算法防护方案^[11],应用了二阶算术加、模乘、有限域 sbox 全掩码方案以及二阶线性掩码和固定掩码 sbox 以及伪轮方案,增加 ZUC 算法抵抗 DPA 攻击的能力。并且我们将其在 FPGA 上实现,并采集功耗曲线,进行实际的 DPA 攻击,实现结果表明,DPA 攻击不能攻击出防护方案的密钥。

理论分析和实验结果表明,和文献[4]中提出的伪轮和波动动态互补逻辑防护方案相比,我们提出的结合掩码和伪轮的防护方案更能提高 ZUC 算法抵御 DPA 攻击的能力。在接下来的研究中,我们将从理论上提升防护方案的安全性,并在不牺牲速度的基础上进一步减小硬件实现的功耗和面积。

参考文献:

- [1] KOCHER P, JAFFE J, JUN B. Differential power analysis[C]// Annual International Cryptology Conference. Springer Berlin, Heidelberg, 1999: 388-397.
- [2] 黄橙. 分组密码算法实现效率研究[D]. 成都:四川大学, 2005.

- [3] 冯秀涛. 3GPP LTE 国际加密标准 ZUC 算法[J]. 信息安全与通信保密, 2011, 12 (9):45-46.
- [4] MANSOURI S S, DUBROVA E. An architectural countermeasure against power analysis attacks for FSR-based stream ciphers[J]. Lecture Notes in Computer Science (LNCS), 2012(7275):54-68.
- [5] LANO J, MENTENS N, PRENEEL B, et al. Power analysis of synchronous stream ciphers with resynchronization mechanism[C]//The State of the Art of StreamCiphers-SASC 2004, Workshop Record. [S. l]. 2004:327-333.
- [6] Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3[R]. Document 4: Design and Evaluation Report. ETSI/SAGE Specification, Version: 2. 0, 2011.
- [7] TANG M, CHENG P P, QIU Z L. Differential power analysis on ZUC algorithm [EB/OL]. (2013-12-01). IACR ePrint. <http://eprint.iacr.org/2012/299.pdf>.
- [8] MANGAND S, OSWALD E, POPP T. Power analysis attacks:revealingthe secrets of smart carts[M]. Springer US. 2007.
- [9] 郁宁亚, 朱宇霞. 基于 FPGA 的祖冲之序列密码算法实现[J]. 信息技术, 2015, 38(9):125-129.
- [10] STEFAN MANGARD, 曼格德, 奥斯瓦尔德, et al. 能量分析攻击[M]. 北京:科学出版社, 2010.
- [11] RIVAIN M, PROUFF E. Provably secure higher-order masking of AES[C]// International Conference on Cryptographic Hardware & Embedded Systems. Santa Barbara, CA, USA, 2010(6225): 413-427.

作者简介:

朱文锋 男, (1994-), 硕士研究生. 研究方向为分组密码算法掩码方案设计和攻击. E-mail: zhuwenfengzi@foxmail.com.

郭 箐 男, (1980-), 博士, 工程师. 研究方向为集成电路安全分析和测试.