基于 FPGA 平台的 Piccolo 功耗分析安全性评估

王晨旭*①② 李景虎② 喻明艳①② 王进祥①
①(哈尔滨工业大学微电子中心 哈尔滨 150001)
②(哈尔滨工业大学(威海)微电子中心 威海 264209)

摘 要：为了评估 Piccolo 密码算法的功耗分析安全性，该文提出一种针对 Piccolo 末轮的攻击模型，基于 SASEBO (Side-channel Attack Standard Evaluation BOard)实测功耗数据对该算法进行了相关性功耗分析攻击。针对 Piccolo 末轮运算中包含白化密钥的特点，将末轮攻击密钥(包括轮密钥 RK_{24L}, RK_{24R}, WK_2, WK_3)分成 4 段子密钥，逐个完成各个子密钥的攻击，使 80 位种子密钥的搜索空间从 2^{80} 降低到$(2\times2^{20}+2\times2^{12}+2^{16})$，使种子密钥的恢复成为可能。攻击结果表明，在实测功耗数据情况下，3000 条功耗曲线即可恢复 80 位种子密钥，证实了该攻击模型的有效性和 Piccolo 硬件面向功耗分析的脆弱性，研究并采取切实有效的防护措施势在必行。

关键词：密码学；数据安全；Piccolo；相关性功耗分析；攻击模型；防护措施；侧信道攻击标准评估板
中图分类号：TP309.2 **文献标识码**：A **文章编号**：1009-5896(2014)01-0101-07
DOI: 10.3724/SP.J.1146.2013.00193

Power Analysis Security Evaluation on Piccolo Based on FPGA Platform

Wang Chen-xu①② Li Jing-hu② Yu Ming-yan①② Wang Jin-xiang①
①(*Microelectronics Center*, *Harbin Institute of Technology*, *Harbin* 150001, *China*)
②(*Microelectronics Center*, *Harbin Institute of Technology*, *Weihai* 264209, *China*)

Abstract: To evaluate Piccolo's security against Power Analysis Attack (PAA), a cipher text attack model is proposed and Correlation Power Analysis (CPA) is conducted on this cipher implementation with measured power traces based on Side-channel Attack Standard Evaluation BOard (SASEBO). Due to the whiten keys for the final round of Piccolo, attacked keys including RK_{24L}, RK_{24R}, WK_2 and WK_3 are divided into four sub-keys, which are disclosed one by one. This approach can reduce the 80-bit primary key search space from 2^{80} to $(2\times2^{20}+2\times2^{12}+2^{16})$ and make it possible to recover the primary key. The attack results show that 3000 measured power traces are enough to recover Piccolo's 80-bit primary key, which proves the attack model's feasibility and Piccolo's vulnerability to CPA against its hardware implementation. So, some countermeasures should be used for Piccolo's hardware implementation.

Key words: Cryptography; Data security; Piccolo; Correlation Power Analysis (CPA); Attack model; Countermeasure; Side-channel Attack Standard Evaluation BOard (SASEBO)

1 引言

密码算法是数据安全的基础，近年来，在无线传感器网络(WSN)和射频识别(RFID)应用中，资源消耗和数据安全这对矛盾体的出现给传统加密算法带来了新的挑战，占用资源少、功耗低的轻量级分组密码算法应运而生。其中，CLEFIA[1]和PRESENT[2,3]是两种最典型的轻量级密码算法，并已于 2012 年成为 ISO 标准；在 CHES2011 会议上，索尼公司继 CLEFIA 之后提出了更加紧凑的轻量级分组密码算法 Piccolo，该算法在 0.13 μm 工艺下只需 683 个等效门(Gate Equivalents, GE)即可实现加密操作[4]。文献[5]对近年来出现的轻量级密码算法进行了硬件资源评估，指出 Piccolo 算法在所有测评的分组密码算法中占用最小的硬件资源，尤其适合在资源受限的环境中，为目前稍显尴尬的 RFID 加密应用提供了一种可行的解决方案。

另一方面，密码算法的安全性也是我们必须要考虑的一个重要问题。密码算法的安全性包括两个方面：算法自身的安全性和实现的安全性。在文献[4]中，作者分别对 Piccolo 的差分分析安全性和线性分析安全性等方面进行了评估，并声称该算法设计是安全的；然而，作者对该算法的实现安全性并未

2013-02-06 收到，2013-07-02 改回
国家自然科学基金(60973162)资助课题
*通信作者：王晨旭 wangchenxu@hit.edu.cn

作出阐释，事实上，近年来，密码算法的实现安全性受到了相关性功耗分析(Correlation Power Analysis, CPA)攻击的严峻挑战[6−8]，它通过分析密码设备在运行过程中产生的功耗、电磁辐射等信息进行密钥攻击，该方法以其成本低、攻击力强、防护困难等特点引起了国内外研究者的极大关注。近几年来，有很多报道称采用这种方法成功破解一些实际的密码芯片或密码设备[9−12]，文献[9]使用功耗分析攻击的方法成功破解了基于 PIC18F2420 单片机实现的 AES 算法；文献[10]搭建了一个改进型攻击平台，采用有限次加密功耗曲线成功攻击了多款接触式 IC 卡的密钥；文献[11]以 Mifare DESFire 非接触式 IC 卡为攻击对象，短时间内即能获取其中的 112 bit 3-DES 密钥，并指出该方法能够很容易地做成专用攻击设备；文献[12]则通过此类方法成功破解了 Virtex-II 内置的 3-DES 密码硬件的密钥，使高端 FPGA 的克隆成为可能。由此可见，在 Piccolo 密码算法付诸量产和应用之前，评估其实现方面的安全性就变得尤其重要，文献[13−15]从不同的角度评测了 Piccolo 在面对差分故障攻击时的安全性，证实了 Piccolo 的硬件实现在故障分析方面的脆弱性，本文则基于 FPGA 硬件平台，首次对该算法的 FPGA 实现在功耗分析方面的安全性进行了评估，提出了一个切实可行的攻击模型，成功地实施了对 Piccolo 的功耗分析攻击。

本文第 2 节简要介绍了本文所采用的一些符号标记，描述了 Piccolo 算法；第 3 节介绍 CPA 攻击的基本原理；第 4 节则描述了 Piccolo 功耗攻击模型和猜测功耗矩阵的建立方法；第 5 节给出了实验配置和部分攻击结果，并对结果进行了简要讨论；最后给出了本文的结论和下一步工作展望。

2 Piccolo 算法简介

Piccolo 分组密码算法的分组长度为 64 bit，支持 80 bit 和 128 bit 两种密钥长度，分别用 Piccolo-80 和 Piccolo-128 表示，对应的迭代轮数分别为 25 轮和 31 轮。本文以 Piccolo-80 为研究对象，为方便解释，下文在无特殊说明的情况下，Piccolo 均指 Piccolo-80。以下首先给出本文所用符号标记的含义，而后对算法做简要介绍。

2.1 符号标记含义说明

$a_{(b)}$ 为二进制数据 a 的长度为 b 位，$\boldsymbol{a}^{\mathrm{T}}$ 为向量或矩阵 $\boldsymbol{a}$ 的转置，$\{a\}_b$ 为用 b 进制表示数据 a，$\{a,b,\cdots\}$ 为将数值 $a,b,\cdots$ 进行拼接，$X(a:b)$ 为选择变量 X 的第 a 位到第 b 位，$\mathrm{HW}(a)$ 为 a 的汉明重量，$\mathrm{HD}(a,b)$ 为 a 和 b 的汉明距离，$\mathrm{HP}(a:b)$ 为 a 位到 b 位的假设功耗值。

2.2 密钥扩展部分

Piccolo 的轮密钥扩展采用了基于置换的实现方法，这使得轮密钥建立时间更短，同时也在一定程度上减少了硬件开销。密钥扩展算法使用 80 bit 的种子密钥 PK 为输入，输出 4 个 16 bit 的白化密钥 $WK_{j(16)}(0\le j<4)$和 50 个 16 bit 的轮密钥 $RK_{i\mathrm{L/R}(16)}$ $(0\le j<25)$，其中 $RK_{i\mathrm{L}(16)}$和 $RK_{i\mathrm{R}(16)}$ $(0\le j<25)$分别表示第 i 轮的左半侧和右半侧轮密钥。白化密钥和轮密钥的生成过程如下：

$\{PK_{0(16)},PK_{1(16)},PK_{2(16)},PK_{3(16)},PK_{4(16)}\}=PK_{(80)}$

$WK_0=\{PK_0^{\mathrm{L}},PK_1^{\mathrm{R}}\},WK_1=\{PK_1^{\mathrm{L}},PK_0^{\mathrm{R}}\}$

$WK_2=\{PK_4^{\mathrm{L}},PK_3^{\mathrm{R}}\},WK_3=\{PK_3^{\mathrm{L}},PK_4^{\mathrm{R}}\}$

for i from 0 to 24 do

$$\{\mathrm{RCON}_{i\mathrm{L}(16)},\mathrm{RCON}_{i\mathrm{R}(16)}\}=\{c_{i+1(5)},c_{0(5)},c_{i+1(5)},\{00\}_2,c_{i+1(5)},c_{0(5)},c_{i+1(5)}\}\oplus\{0\mathrm{f1e2d3c}\}_{16}$$

$$\{RK_{i\mathrm{L}(16)},RK_{i\mathrm{R}(16)}\}=\{\mathrm{RCON}_{i\mathrm{L}(16)},\mathrm{RCON}_{i\mathrm{R}(16)}\}\oplus\begin{cases}\{PK_{2(16)},PK_{3(16)}\};(\text{if } i\%5=0 \text{ or } 2)\\ \{PK_{0(16)},PK_{1(16)}\};(\text{if } i\%5=1 \text{ or } 4)\\ \{PK_{4(16)},PK_{4(16)}\};(\text{if } i\%5=3)\end{cases}$$

end for

式中，$\mathrm{RCON}_{i\mathrm{L}}$ 和 $\mathrm{RCON}_{i\mathrm{R}}$ 分别表示第 i 轮的轮常数的左半部分和右半部分，c_i 为 i 的 5 bit 二进制表示，例如 c_{22}=$\{10110\}_2$。

由上述过程不难看出，最后一轮中所涉及的白化密钥 WK_2 和 WK_3 对应于种子密钥 PK 中的 PK_3, PK_4，轮密钥 $RK_{24\mathrm{L}}$ 和 $RK_{24\mathrm{R}}$ 与 PK 中的 PK_0, PK_1 则只是相差一个轮常数 $\mathrm{RCON}_{24\mathrm{L}}$ 和 $\mathrm{RCON}_{24\mathrm{R}}$。因此，如果能够获取 WK_2, WK_3, $RK_{24\mathrm{L}}$ 和 $RK_{24\mathrm{R}}$，则能够通过其它方式破解 PK_2 以获得 Piccolo 的所有 80 bit 种子密钥。

2.3 数据处理部分

Piccolo 算法采用广义 Feistel 结构，数据处理部分以 64 bit 明文、白化密钥和轮密钥为输入经由 25 轮迭代后产生 64 bit 密文输出，如图 1 所示，从图中可以看出，除最后一轮外，每轮包含两类变换，分别是 F 函数 F: $\mathrm{GF}(2^{16})\rightarrow\mathrm{GF}(2^{16})$和轮置换 RP: $\mathrm{GF}(2^{64})\rightarrow\mathrm{GF}(2^{64})$，最后一轮不包含轮置换 RP。

***F* 函数：** Piccolo 的 F: $\mathrm{GF}(2^{16})\rightarrow\mathrm{GF}(2^{16})$也被称为超级 S 盒[16]，采用 3 层结构，具有更强的混淆能力，如图 2 所示。图中，S 盒 S: $\mathrm{GF}(2^4)\rightarrow\mathrm{GF}(2^4)$ 是 Piccolo 中唯一的非线性操作；$\boldsymbol{M}$ 代表混淆矩阵，$\boldsymbol{M}$ 矩阵的引入使输入 X 的每一位扩散影响输出 Y 的每一位，后文将能看到这一点会增加功耗分析攻击的难度。

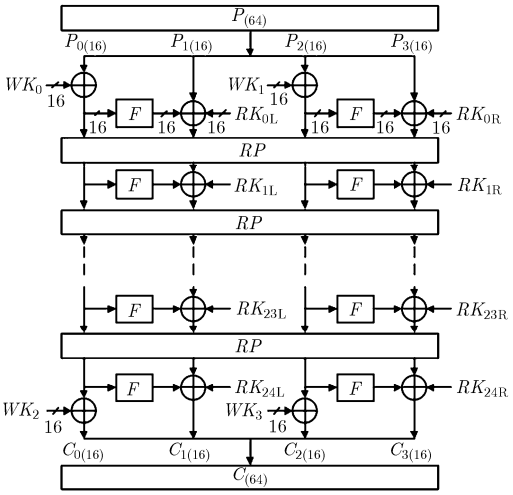

图1 Piccolo密码算法数据处理流程

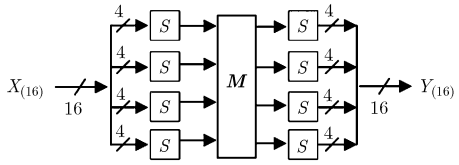

图2 Piccolo密码算法的F函数

上述关系可以用如下过程进行表示。

$$\{X_{0(4)},X_{1(4)},X_{2(4)},X_{3(4)}\}=X_{(16)}$$

$$\{X_{0(4)},X_{1(4)},X_{2(4)},X_{3(4)}\}$$
$$=\{S(X_{0(4)}),S(X_{1(4)}),S(X_{2(4)}),S(X_{3(4)})\}$$

$$(X_{0(4)},X_{1(4)},X_{2(4)},X_{3(4)})^{\mathrm{T}}$$
$$=\boldsymbol{M}\cdot(X_{0(4)},X_{1(4)},X_{2(4)},X_{3(4)})^{\mathrm{T}}$$

$$Y_{(16)}=\{S(X_{0(4)}),S(X_{1(4)}),S(X_{2(4)}),S(X_{3(4)})\}$$

轮置换 *RP*： RP: $\mathrm{GF}(2^{64})\rightarrow\mathrm{GF}(2^{64})$以字节为单位进行置换，其过程如图 3 所示。

3　CPA 攻击的基本原理

在一个典型的同步时序电路中，一个时钟周期内(两个上升沿之间)触发器保持稳定，整个电路的功耗主要是由组合电路翻转带来的动态功耗；当上升沿到来后的极短一段时间内，各触发器状态更新，整个电路的功耗则由触发器的翻转引起的动态功耗组成，其值必定与触发器的翻转个数(以汉明距离予以表示)相关。由此可知，加密过程中密码芯片所产生的功耗必然与运算期间某些中间值存在一定的相关性。CPA 攻击方法的主要思想就是通过分析这种相关性达到破解算法密钥的目的，一般情况下，这种相关性比较小，因此需要大量的加密获取不同情况下的功耗数据，结合统计分析技术才能够成功破解密钥。在统计学理论中，通常用 Pearson 相关系数来评价两个向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 之间的相关性[6]。

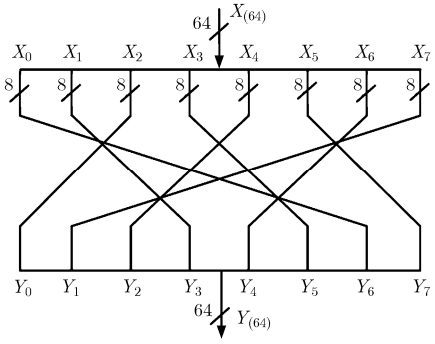

图3 Piccolo密码算法的RP置换过程

$$\rho_{X,Y}=\frac{\mathrm{Cov}(\boldsymbol{X},\boldsymbol{Y})}{\sigma_X\sigma_Y}=\frac{\sum_{d=1}^{D}(X_d-\overline{X})\cdot(Y_d-\overline{Y})}{\sqrt{\sum_{d=1}^{D}(X_d-\overline{X})^2\cdot\sum_{d=1}^{D}(Y_d-\overline{Y})^2}}\quad(1)$$

式中 $\mathrm{Cov}(\boldsymbol{X},\boldsymbol{Y})$为向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 之间的协方差，σ_X 和σ_Y 则代表向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的标准差；X_d和 Y_d分别代表向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的第 d 维分量的值，$\overline{X}$ 和$\overline{Y}$ 分别代表向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的均值。显然，$|\rho_{X,Y}|\le 1$，且$|\rho_{X,Y}|$的值越大说明向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 越相关。

下面以密文攻击(以最后轮的轮密钥为攻击目标)为研究对象，描述密文 CPA 攻击一般步骤：

(1)结合密码算法的实现形式，合理选取密码算法运算期间的某些中间值 $v_i=f_i(C_{\mathrm{sub}},RK_{24,i})$，式中，$C_{\mathrm{sub}}$代表密文的一部分，$RK_{24,i}$代表最后轮的第 i 部分轮密钥(所有的 $RK_{24,i}$组合起来即为最后轮的完整轮密钥)，通常函数 f_i中要包含非线性函数。然后，通过合适的功耗模型将 vi 映射为猜测功耗值 $h_i=g_i(C_{\mathrm{sub}},RK_{24,i})$。$h_i$是 C_{sub}和 $RK_{24,i}$的函数，所以针对 N 个不同的密文 C 和 $RK_{24,i}$所有 K=2^k(k 为$RK_{24,i}$的位数)个不同的猜测值，能够建立一个 $N\times K$ 的猜测功耗矩阵 $\boldsymbol{H}_i$。这一步是能否成功实施 CPA 攻击的关键，其难点在于如何根据密码算法的特点找出一个合理的中间值。

(2)采集加密过程中的实际功耗信息，记录下与之相应的密文。假设每次加密过程的功耗信息由 T 个采样点构成，通过换取 N 条不同明文，执行 N 次加密过程，则可以得到由 $N\times T$ 个采样点构成的实际功耗矩阵 $\boldsymbol{W}$，需要指出，实际功耗信息的获取通常有两种方式：基于功耗模拟工具模拟功耗数据的获取方式[17]和基于真实硬件环境和示波器测量功耗数据的获取方式[9−12]，本文采用后者。

(3)对 $\boldsymbol{H}_{N\times K}$和 $\boldsymbol{W}_{N\times T}$进行统计分析，获得密钥 $RK_{24,i}$ 的最可能值。具体做法是依式(1)分别计算矩阵 $\boldsymbol{H}_{N\times K}$的各列与 $\boldsymbol{W}_{N\times T}$各列之间的 Pearson 相关系

数，获得一个 K 行 T 列的相关系数矩阵 $\boldsymbol{R}$，该过程可表示为图 4。矩阵 $\boldsymbol{R}$ 的第 i 行第 j 列的元素表示 $\boldsymbol{H}$ 的第 i 列向量与 $\boldsymbol{W}$ 的第 j 列向量的相关系数。寻找矩阵 $\boldsymbol{R}$ 中各个元素的最大值，假设最大值位于矩阵的第 ck 行，第 ct 列，则 $(ck-1)$ 代表部分轮密钥 $RK_{24,i}$ 的最可能值，$(ct-1)$ 代表了相关系数最大的时刻，至此就完成了对 RK_i 的攻击。

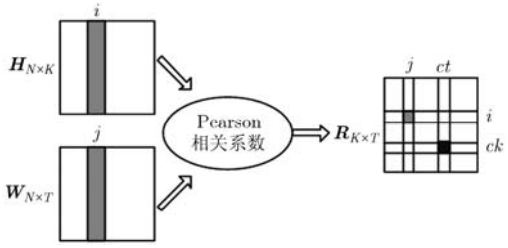

图4 CPA的统计分析过程

4 Piccolo 算法的 CPA 攻击

4.1 Piccolo 的硬件实现

Piccolo 算法的 ASIC 硬件实现方式主要有两种，一是基于轮的并行实现方法，它可以得到较高的数据吞吐率，但消耗的硬件资源较多。二是将输入数据进行分组，每组分别处理，再予以拼接，即串行实现方法，这种方法能够显著地减小硬件资源消耗，683GE 即可实现[4]。本文评估了第 1 种实现方法的安全性，对串行实现方式的评测也可以采用类似方法。并行实现方式对输入 64 bit 明文数据并行处理，每轮的处理结果保存在 64 个触发器中(本文以 DFF(0:63)表示，并用 DFF(0)表示这些触发器的最高有效位)，历经 25 轮运算后输出 64 bit 密文。本文所完成的 Piccolo 设计占用 1624GE, 26 个时钟周期能完成一个分组的加密。

4.2 攻击模型

通常情况下，密文攻击相对于明文攻击更加实用，因此，后文主要以密文攻击为说明对象来评测 Piccolo 算法的实现安全性，即以获取最后轮的轮密钥 $RK_{24\mathrm{L}}$, $RK_{24\mathrm{R}}$ 和白化密钥 WK_2, WK_3(为解释方便，下文将 WK_2, WK_3, $RK_{24\mathrm{L}}$ 和 $RK_{24\mathrm{R}}$ 统称为攻击密钥)为攻击目标。虽然 Piccolo 算法采用了类似 DES 算法的 Feistel 结构，但是由于超级 S 盒和白化密钥的存在，Piccolo 的 CPA 攻击与传统密码算法如 DES, AES 和 PRESENT 相比存在较大难度。

图 5 给出了 Piccolo 算法并行实现方式的最后一轮(称为攻击兴趣轮)的抽象。可以看出，DFF(0:63)在 t_1 时刻的值(即密文输出 C)受到 4 个攻击密钥和第 24 轮处理结果 S_{24}(0:63)的影响，如果按照第 3 节

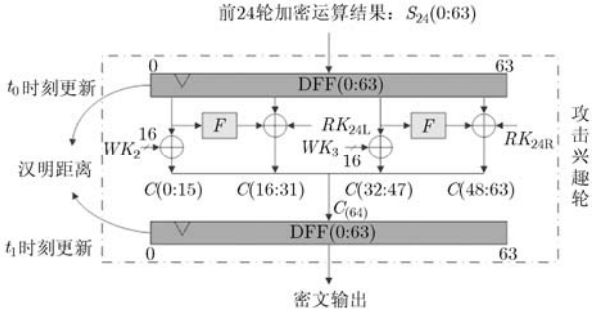

图 5 Piccolo 算法并行实现方式最后一轮抽象图

所述选取 $v=f(C,RK_{24\mathrm{L}},RK_{24\mathrm{R}},WK_2,WK_3)$，则在建立猜测功耗矩阵时需要遍历 64 bit 攻击密钥的所有可能(2^{64})，显然这种方法在当前的计算能力下是不可能实现的。

根据部分相关性原理，如果一个电路结构的整体和电路的功耗具有相关性，电路结构的一部分也和电路的功耗具有一定的相关性，因此，部分触发器的翻转对总功耗也会产生一定的影响，为方便计算，考虑基于 DFF(0:63)中的部分触发器的汉明距离进行功耗建模，将 WK_2, WK_3, $RK_{24\mathrm{L}}$, $RK_{24\mathrm{R}}$ 按下式进行拆分组合为 $\mathrm{SubKey}_{1(20)}$, $\mathrm{SubKey}_{2(12)}$, $\mathrm{SubKey}_{3(20)}$和 $\mathrm{SubKey}_{4(12)}$ 4 段子密钥。

$$\mathrm{SubKey}_{1(20)}=\{WK_2(0:15),RK_{24\mathrm{L}}(0:3)\}$$
$$\mathrm{SubKey}_{2(12)}=RK_{24\mathrm{L}}(4:15)$$
$$\mathrm{SubKey}_{3(20)}=\{WK_3(0:15),RK_{24\mathrm{R}}(0:3)\}$$
$$\mathrm{SubKey}_{4(12)}=RK_{24\mathrm{R}}(4:15)$$

此时，可以将中间值设定为 $v_i=f_i(C,\mathrm{SubKey}_i)$ $(1\le i\le 4)$ 并为之建立猜测功耗矩阵，然后针对这 4 段子密钥逐个进行攻击，这样，对 WK_2, WK_3, $RK_{24\mathrm{L}}$ 和 $RK_{24\mathrm{R}}$ 的攻击问题变成了对这 4 段子密钥的攻击问题，将攻击密钥的搜索空间降低到 $(2\times2^{20}+2\times2^{12})$，给计算创造了可能。

4.2.1 针对 $\mathrm{SubKey}_{1(20)}$ 的功耗建模 为了攻击 $RK_{24\mathrm{L}}$(0:3)，考虑选取倒数第 2 轮的结果 S_{24}(16:19)作为中间值，并使用 DFF(16:19)的汉明距离进行功耗映射，此时，$v_1=f_1(C(16:19),\mathrm{SubKey}_1)=f_1(C(16:19),WK_2,RK_{24\mathrm{L}}(0:3))$，通过对 $\mathrm{SubKey}_{1(20)}$进行遍历，即可通过 C(16:19)恢复出触发器 DFF(16:19)在 t_0 时刻的值，计算 DFF(16:19)在 t_1 时钟沿前后的汉明距离，攻击模型如下：

$$F_{\mathrm{in}}(0:15)=C(0:15)\oplus WK_2$$
$$S_{24}(16:19)=C(16:19)\oplus F(F_{\mathrm{in}})(0:3)\oplus RK_{24\mathrm{L}}(0:3)$$
$$\begin{aligned}\mathrm{HP}(16:19)&=\mathrm{HD}(S_{24}(16:19),C(16:19))\\&=\mathrm{HW}(S_{24}(16:19)\oplus C(16:19))\\&=\mathrm{HW}(F(C(0:15)\oplus WK_2)(0:3)\\&\quad\oplus RK_{24\mathrm{L}}(0:3))\end{aligned}$$

式中 $F_{\text{in}}(0{:}15)$表示 F 函数的输入；HP(16:19)表示基于 DFF(16:19)的功耗模型值。

根据上述模型，对于某个特定的 64 bit 明文，通过对 $\text{SubKey}_{1(20)}$的 2^{20} 次遍历可以得到一个关于 HP(16:19)的 1×2^{20} 的汉明距离矩阵，这个矩阵代表了在不同的 $\text{SubKey}_{1(20)}$猜测下，触发器翻转时刻的猜测的功耗信息。如果对 N 条明文进行计算，则可以得到一个 $N\times2^{20}$ 的矩阵，这个矩阵即为我们攻击 $\text{SubKey}_{1(20)}$所需的猜测功耗矩阵 $\boldsymbol{H}_1$，该矩阵用来与实际功耗矩阵 $\boldsymbol{W}$ 做统计分析。

4.2.2 针对 $\text{SubKey}_{2(12)}$ 的功耗建模　在完成了 $\text{SubKey}_{1(20)}$的攻击后，WK_2 成为了已知量，由图 5 可以发现，$RK_{24\text{L}}(4{:}15)$影响了 DFF(20:31)在 t_1 时刻的值。为了攻击 $RK_{24\text{L}}(4{:}15)$，考虑将倒数第 2 轮的结果 $S_{24}(20{:}31)$作为选取的中间值，即 $v_2=f_2(C(20:31),\text{SubKey}_2)=f_2(C(20:31),RK_{24\text{L}}(4:15))$，并使用 DFF(20:31) 的汉明距离进行功耗映射。对 $\text{SubKey}_{2(12)}$进行遍历，即可通过 $C(20{:}31)$恢复出触发器 DFF(20:31)在 t_0 时刻的值，计算 DFF(20:31)在 t_1 时钟沿前后的汉明距离，建模过程如下：

$$\begin{aligned}\text{HP}(20:31)&=\text{HD}(S_{24}(20:31),C(20:31))\\&=\text{HW}(S_{24}(20:31)\oplus C(20:31))\\&=\text{HW}(F(C(0:15)\oplus WK_2)(4:15)\\&\quad\oplus RK_{24\text{L}}(4:15))\end{aligned}$$

根据上述模型，通过 N 条不同的明文和对 $\text{SubKey}_{2(12)}$进行遍历可以得到一个关于 HP(20:31)的 $N\times2^{20}$ 的猜测功耗矩阵 $\boldsymbol{H}_2$，该矩阵用来与实际功耗矩阵 $\boldsymbol{W}$ 做统计分析以攻击 $RK_{24\text{L}}(4{:}15)$。

4.2.3 针对 $\text{SubKey}_{3(20)}$和 $\text{SubKey}_{4(12)}$的功耗建模

由于 Piccolo 算法左右两侧的高度对称性，对 $\text{SubKey}_{3(20)}$和 $\text{SubKey}_{4(12)}$的攻击方法与 $\text{SubKey}_{1(20)}$和 $\text{SubKey}_{2(12)}$基本一致，所不同的是这里需要分别选取 $S_{24}(36{:}39)$和 $S_{24}(40{:}31)$为中间值进行功耗映射。

5　攻击实验配置与攻击结果

5.1 攻击实验配置

SASEBO(Side-channel Attack Standard Evaluation BOard)系列板是用于评测抗侧信道攻击能力的基准平台[18]，由日本 AIST 信息安全研究中心开发。本文采用 SASEBO-G[19]作为基准搭建了如图 6 所示的功耗采集平台，其中，PC 机采用目前主流计算机，PC 机与 SASEBO-G 通过 RS-232 相连，并通过 GPIB 适配器控制示波器(Agilent MSO8064A)的行为。示波器的通道 2 采用普通单端探头用于接收 SASEBO-G 板上的触发信号，通道 1

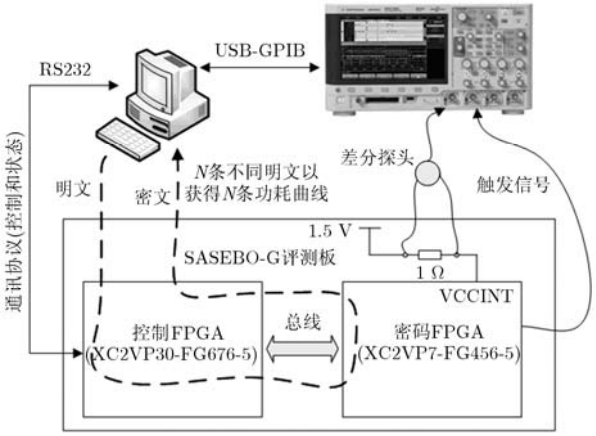

图 6 功耗采集平台硬件原理图

采用差分探头用于探测 SASEBO-G 板上密码 FPGA 芯片的电流变化。

基于图 6 的功耗采集解决方案如下：首先制订 PC 机与下位机(SASEBO-G)之间的通讯协议；在 SASEBO-G 评测板上，Piccolo 算法实现在密码 FPGA 芯片内，控制 FPGA 主要完成上述通讯协议的解析和密码 FPGA 的控制；在 PC 机上用 NI 公司的 LabVIEW®实现流程控制，LabVIEW 产生 64 bit 的随机明文，通过 RS232 经由控制 FPGA 发送到密码 FPGA 芯片中进行 Piccolo 加密，期间密码 FPGA 芯片向示波器产生一个触发信号，保证每次 Piccolo 加密时示波器采集的数据都能准确对齐；控制 FPGA 等待加密结束，将存储在密码 FPGA 中的密文读出并经由 RS232 送往 PC 机；LabVIEW 负责接收并存储密文，然后向示波器发出指令以存储此次采集到的数据，一条功耗曲线采集完毕。

为了达到上述要求并方便控制，我们在实现 Piccolo 密码算法时加入了局部总线接口和可编程寄存器，并在控制 FPGA 中实现了一个基于 ARM7 的最小 SoC 系统，采用固件实现与 PC 机的通讯和协议解析，此最小系统中仅仅包括 ARM7 核，UART 和一个用于与密码 FPGA 芯片通讯的局部总线控制器。

实验中，最小 SoC 系统运行于 24 MHz，而 Piccolo 密码算法运行于 4 MHz，波特率设定为 115200；设定 5000 组明文输入，示波器采样率为 2 GSa/s，由于触发信号保证了每次 Piccolo 加密时示波器采集的数据都能准确对齐，为了降低测量噪声，将示波器设置为 20 次平均采样模式，即每组明文重复执行加密过程 20 次，由示波器将 20 次采集到的数据自动平均，因此，Piccolo 实际执行加密的次数为 5000×20。

5.2 采用 3000 条功耗曲线时的攻击结果

由于对 SubKey_1 和 SubKey_2 与对 SubKey_3 和 SubKey_4 的攻击过程基本一致，因此这里仅仅给出对前两个子密钥的攻击结果。实验中，Piccolo 加密时的种子密钥取 $\{00112233445566778899\}_{16}$，在 3000 条功耗曲线时，计算 $\boldsymbol{H}_1$ 矩阵与 $\boldsymbol{W}$ 矩阵的相关系数矩阵 $\boldsymbol{R}_1$，取相关系数矩阵 $\boldsymbol{R}_1$ 各行的最大值，这些最大值可以构成一个 $2^{20}\times 1$ 的向量，该向量结果见图 7，图中横坐标表示密钥猜测值，纵坐标表示了相应的相关系数。

由图 7 可以发现，当 $x=\{558972\}_{10}=\{8877\mathrm{c}\}_{16}$ 时，相关系数达到最大值 0.0937，这说明在本次攻击中 $\{8877\mathrm{c}\}_{16}=\{1000\_1000\_0111\_0111\_1100\}_2$ 最有可能是 SubKey_1 的真实值，由此可推出 WK_2 的攻击密钥值为 $\{1000\_1000\_0111\_0111\}_2$，而 $RK_{24\mathrm{L}}(0{:}3)$的攻击密钥值为$\{1100\}_2$，事实上，WK_2 和 $RK_{24\mathrm{L}}(0{:}3)$的真实密钥值也的确如此。

图 8 给出了在 3000 条功耗曲线时对 SubKey_2 的攻击结果，在

$$x=\{1853\}_{10}=\{73\mathrm{d}\}_{16}=\{0111\_0011\ \_1101\}_2$$

时获得了最大的相关系数 0.2483，即 $RK_{24\mathrm{L}}(4{:}15)$的攻击密钥值为$\{0111_0011_1101\}_2$。综上，使用 3000 条功耗曲线对 Piccolo 进行密文 CPA 攻击后，得到 $RK_{24\mathrm{L}}=\{\mathrm{c73d}\}_{16}$，$WK_2=\{8877\}_{16}$，这些结果与预期值相同，表明攻击成功。

5.3 安全性评估方法与结果

我们采用成功实施攻击所需的功耗曲线样本数量(Measurements To Disclosure, MTD)来评估密码算法实现的抗功耗分析攻击的能力及安全性[10−12,20]。实验中，我们从采集到的 5000 条功耗曲线中选取 500 条，依照上述 CPA 攻击过程，开始实施 CPA 攻击，之后依次递增 250 条功耗曲线，直至使用 3000 条以上的功耗曲线能够稳定恢复出全部 4 个子密钥。图 9 表示攻击 SubKey_1 时功耗曲线数量对 SubKey_1 猜测值相关系数的影响，可以看出，随着功耗曲线数量的增加，正确 SubKey_1 猜测值与其它 SubKey_1 猜测值相关系数的区别不断加大，大约 3000 条样本就已经可以成功破解 SubKey_1，这说明未加防护措施的 Piccolo 算法硬件实现存在被功耗分析攻击的威胁。

5.4 讨论

5.4.1 相关度 根据上述讨论，在 t_1 时刻，实际 Piccolo 硬件的功耗可近似用 DFF (0:63)全部 64 个触发器的动态功耗表征；但是，在攻击模型建立时，SubKey_1 依赖于 DFF(16:19)共 4 个触发器，而 SubKey_2 则有赖于 DFF(20:31)共 12 个触发器，因此，攻击 SubKey_2 时用到的功耗模型更加接近于真实情况。这造成了在成功攻击 SubKey_2 时的相关系数(0.2483，见图 8)比攻击 SubKey_1 时的相关系数(0.0937，见图 7)要高。

事实上，实验中攻击 SubKey_1 的 MTD_1 为 3000 条，而在成功攻击 SubKey_2 时的 MTD_2 只需要 1750 条，MTD_2 远小于 MTD_1 主要是因为针对 SubKey_2 的功耗模型信噪比更高，与实际功耗具有更高的相关度。

5.4.2 种子密钥的恢复 通过 CPA 攻击实验，我们已成功获取 WK_2, WK_3, $RK_{24\mathrm{L}}$ 和 $RK_{24\mathrm{R}}$，依据轮密钥扩展算法可以容易恢复出 Piccolo 的 PK_0, PK_1, PK_3 和 PK_4。为了获取完整的 80 bit PK，还需单独针对 PK_2 进行攻击，如果使用最简单的方式穷举破解 PK_2，那么采用本文提议的 CPA 攻击方法所需要的完整密钥搜索空间为$(2\times 2^{20}+2\times 2^{12}+2^{16})$。

6 结束语

本文提出了一种密文 CPA 攻击模型，基于 SASEBO 平台和实测结果首次评估了 Piccolo 面向功耗分析攻击的安全性。攻击结果表明，基于此攻击模型在实测功耗数据的情况下，只需 3000 条功耗曲线即可完全恢复出 Piccolo 算法的 80 bit 种子密钥，因此在 Piccolo 的硬件实现中引入相应的抗功耗分析攻击措施是不可忽略的。虽然本文只是成功实施了密文攻击，容易使用相似的方法实现明文攻击，相形之下，由于首轮运算中包含了 RP 函数，所以明文攻击要比密文攻击稍显复杂。

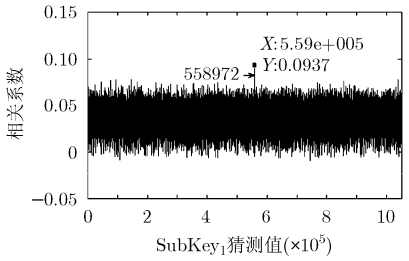

图 7 3000 条功耗样本时对 SubKey_1 的攻击结果

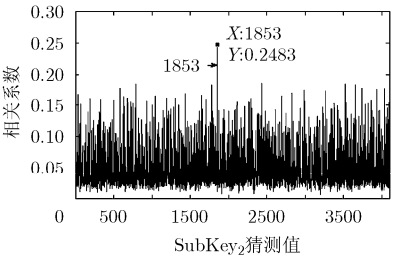

图 8 3000 条功耗样本时对 SubKey_2 的攻击结果

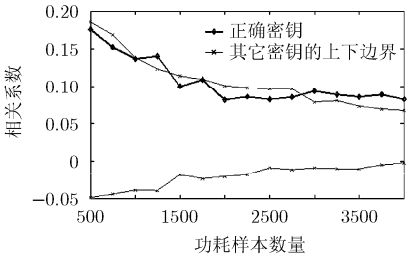

图 9 SubKey_1 猜测值相关系数与功耗样本数量的关系

在本文提出的 CPA 攻击方法中，为获取 80 bit 种子密钥共需要$(2\times2^{20}+2\times2^{12}+2^{16})$次遍历运算，这虽然是可能的，但其时间和空间复杂度都比较大，因此探究降低遍历计算复杂度的方法并研究适用于轻量级分组密码算法的抗功耗分析攻击措施将是下一步的研究重点。

参 考 文 献

[1] ISO/IEC 29192-2-b:2012. CLEFIA: a lightweight block cipher with a block size of 128 bits and a key size of 128, 192 or 256 bits[S]. 2012.

[2] ISO/IEC 29192-2-a:2012. PRESENT: a lightweight block cipher with a block size of 64 bits and a key size of 80 or 128 bits[S]. 2012.

[3] Bogdanov A, Knudsen L R, Leander G, *et al*.. PRESENT: an ultra-lightweight block cipher[C]. Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems(CHES2007), Vienna, Austria, 2007: 450–466.

[4] Shibutani K, *et al*.. Piccolo: an ultra-lightweight blockcipher [C]. Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES2011), Nara, Japan, 2011: 342–357.

[5] Kumar A and Aggarwal A. Lightweight cryptographic primitives for mobile Ad hoc networks[C]. Proceedings of 2012 International Conference on Security in Computer Networks and Distributed Systems(SNDS2012), Trivandrum, India, 2012: 240–251.

[6] Brier E, Clavier C, and Olivier F. Correlation power analysis with a leakage model[C]. Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems(CHES2004), Boston, USA, 2004: 135–152.

[7] Liu H Y, Qian G Y, Satoshi Goto, *et al*.. Correlation power analysis based on switching glitch model[C]. Proceedings of the 7th Web Information Systems and Applications Conference(WISA2010), Huhehot, China, 2010, 6513: 191–205.

[8] Li H Y, Wu K K, and Yu F Q. Enhanced correlation power analysis attack against trusted systems[J]. *Security and Communication Networks*, 2011, (4): 3–10.

[9] Breier J and Kleja M. On practical results of the differential power analysis[J]. *Journal of Electrical Engineering*, 2012, 63(2): 125–129.

[10] 乌力吉, 李贺鑫, 任燕婷, 等. 智能卡功耗分析平台设计与实现[J]. 清华大学学报(自然科学版), 2012, 52(10): 1409–1414.
Wu Li-ji, Li He-xin, and Ren Yan-ting, *et al*.. Smart card power analysis platform design and implementation[J]. *Journal of Tsinghua University*(*Science & Technology*), 2012, 52(10): 1409–1414.

[11] Oswald D and Paar C. Breaking Mifare DESFire MF3ICD40: power analysis and templates in the real world[C]. Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES2011), Nara, Japan, 2011: 207–222.

[12] Moradi A, Barenghi A, Kasper T, *et al*.. On the vulnerability of FPGA bitstream encryption against power analysis attacks[C]. Proceedings of the 18th ACM Conference on Computer and Communications Security(CCS2011), Chicago, USA, 2011: 111–123.

[13] 赵光耀, 李瑞林, 孙兵, 等. Piccolo 算法的差分故障分析[J]. 计算机学报, 2012, 35(9): 1918–1925.
Zhao Guang-yao, Li Rui-lin, Sun Bing, *et al*.. Differential fault analysis on Piccolo[J]. *Chinese Journal of Computers*, 2012, 35(9): 1918–1925.

[14] Jeong K. Differential fault analysis on block cipher Piccolo[OL]. http://eprint.iacr.org/2012/399.pdf. 2012.6.

[15] Li S, Gu D W, Ma Z Q, *et al*.. Fault analysis of the Piccolo block cipher[C]. Proceedings of the 8th International Conference on Computational Intelligence and Security (CIS2012), Guangzhou, China, 2012: 482–486.

[16] Daemen J and Rijmen V. Understanding two-round differentials in AES[C]. Proceedings of the 5th International Conference on Security and Cryptography for Networks (SCN2006), Maiori, Italy, 2006: 78–94.

[17] 刘鸣, 陈弘毅, 白国强. 功耗分析研究平台及其应用[J]. 微电子学与计算机, 2005, 22(7): 134–138.
Liu Ming, Chen Hong-yi, and Bai Guo-qiang. Power analysis research platform and its applications[J]. *Microelectronics & Computer*, 2005, 22(7): 134–138.

[18] 汪鹏君, 张跃军, 张学龙. 防御差分功耗分析攻击技术研究[J]. 电子与信息学报, 2012, 34(11): 2774–2784.
Wang Peng-jun, Zhang Yue-jun, and Zhang Xue-long. Research of differential power analysis countermeasures[J]. *Journal of Electronics & Information Technology*, 2012, 34(11): 2774–2784.

[19] AIST of Japan. SASEBO-G Specification[OL]. http://www.risec.aist.go.jp/project/sasebo. 2012.6.

[20] Kim C K, Schläffer M, and Moon S J. Differential side channel analysis attacks on FPGA implementations of ARIA [J]. *ETRI Journal*, 2008, 30(2): 315–325.

王晨旭：　男，1977 年生，讲师，研究方向为芯片与信息安全.

李景虎：　男，1975 年生，讲师，研究方向为数模混合技术.

喻明艳：　男，1965 年生，教授，研究方向为计算机体系结构.

王进祥：　男，1968 年生，教授，研究方向为 SoC 的可靠性设计.