Piccolo 相关性功耗分析攻击技术研究

王晨旭[1,2]，赵占锋[2]，喻明艳[1]，王进祥[1]，姜佩贺[2]

（1. 哈尔滨工业大学 微电子中心，150001 哈尔滨；2. 哈尔滨工业大学 信息与电气工程学院，264209 山东 威海）

摘 要：为了评测轻量级密码算法 Piccolo 抗功耗分析攻击的能力，提出一种针对首轮的功耗分析攻击模型，搭建了功耗模拟采集平台，对该算法进行了相关性功耗分析攻击. 针对 Piccolo 算法首轮运算中包含白化密钥和轮置换操作的特点，将首轮相关攻击密钥（包括轮密钥 RK_{0L}、RK_{0R}、WK_0、WK_1）分成6段子密钥，逐个完成各段子密钥的攻击，将80位种子密钥的搜索空间从 2^{80} 降低到（$2\times2^{20}+2\times2^4+2\times2^8+2^{16}$），使得密钥的恢复成为可能. 攻击结果表明，只需500条功耗曲线即可恢复首轮攻击密钥，由此可见，未加任何防护措施的 Piccolo 硬件实现极易遭受相关性功耗分析攻击，研究并采取切实有效的防护措施势在必行. 据现有资料，这是首次评估 Piccolo 密码算法在相关性功耗分析攻击方面的安全性.

关键词：轻量级密码算法；Piccolo；相关性功耗分析；功耗分析攻击模型；防护措施

中图分类号：TP309.2　　**文献标志码**：A　　**文章编号**：0367－6234(2013)09－0017－06

Research on correlation power analysis attack against Piccolo

WANG Chenxu[1,2], ZHAO Zhanfeng[2], YU Mingyan[1], WANG Jinxiang[1], JIANG Peihe[2]

(1. Microelectronics Center, Harbin Institute of Technology, 150001 Harbin, China;
2. School of Information & Electrical Engineering, Harbin Institute of Technology, 264209 Weihai, Shandong, China)

Abstract: To evaluate an ultra-lightweight blockcipher Piccolo's ability to counteract Power Analysis Attack (PAA), an attack model, which focuses on the first round of Piccolo, was proposed and Correlation Power Analysis (CPA) was conducted on this cipher based on a power simulation acquisition platform. Due to the whiten keys and round permutation for the first round of Piccolo, attacking keys including RK_{0L}, RK_{0R}, WK_0 and WK_1 were divided into six sub-keys, which were disclosed one by one. This approach can reduce the 80－bit primary key search space from 2^{80} to ($2\times2^{20}+2\times2^4+2\times2^8+2^{16}$) and make it possible to recover the primary key. The attack results show that 500 power traces are enough to recover Piccolo's 80－bit primary key. It is concluded that the hardware implementation of Piccolo without any countermeasure is vulnerable to CPA and some countermeasures should be used. This work is the first known report about the security of Piccolo against PAA.

Key words: lightweight blockcipher; Piccolo; correlation power analysis (CPA); power analysis attack model; countermeasure

移动数字终端，无线传感器网络（WSN），射频识别（RFID）等技术的应用日趋广泛，这些技术对终端设备的硬件资源、能耗和终端数据安全等方面提出了更严格的要求. 由于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，轻量级分组密码算法应运而生. 轻量级分组密码算法是在确保加密数据安全的前提下，利用最少的硬件资源，最低的功耗实现的一类加密算法，例如，Sony 公司提出的 CLEFIA 密码算法以及在 CHES2007 上提出的 PRESENT 密码算法已经在 2012 年成为轻量级密码算法的 ISO 标准[1－3]. 作为 CLEFIA 的派生算法，Piccolo 分组密码算法于 CHES2011 上由 Sony 公司提出，

收稿日期：2013－01－17.
基金项目：国家自然科学基金资助项目(60973162).
作者简介：王晨旭(1977—)，男，博士研究生，讲师；
王进祥(1968—)，男，教授，博士生导师.
通信作者：王晨旭，wangchenxu@hit.edu.cn.

它具有良好的硬件实现效率，在 0.13 μm 工艺下仅需 683 个等效门(Gate Equivalents，GE)即可实现加密操作，非常适合在资源受限的环境中使用[4]，展示出了非常好的使用前景.

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，近年来，密码算法的实现安全性受到了侧信道攻击(Side-Channel Attack，SCA)的严峻挑战[5]，它是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的一种方法，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人员的极大关注. 相关功耗分析(Correlation Power Analysis，CPA)是 SCA 的一种，通过建立功耗模型，分析假设功耗与实际功耗曲线(Power Trace)之间的相关性，借助统计方法来完成密钥攻击[6]. 本文首次对该算法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功耗分析攻击模型，成功地实施了对 Piccolo 的功耗分析攻击.

1 相关性功耗分析攻击简介

Piccolo 分组密码算法的分组长度为 64 位，支持 80 位和 128 位两种密钥长度，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. Piccolo 算法结构是广义 Feistel 结构的变种，轮变换包括 Feistel 函数 F 和轮置换函数 RP. 本文的研究对象为 Piccolo－80，并用 Piccolo 指代 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.

1.1 符号标记

$a_{(b)}$：二进制数据 a 的长度为 b 位.

a'：向量或矩阵 a 的转置.

$\{a\}_b$：用 b 进制表示数据 a.

$\{a,b,\cdots\}$：将数值 $a,b,\cdots$ 进行拼接.

$X(a:b)$：选择变量 X 的第 a 位到第 b 位.

$HW(a)$：a 的汉明重量.

$HD(a,b)$：a 和 b 的汉明距离.

$HP(a:b)$：a 位到 b 位的假设功耗值.

1.2 CPA 攻击过程

在 CPA 攻击中，针对首轮的明文攻击和针对最后一轮的密文攻击是两种主要的攻击方式，两种攻击方式的基本原理和攻击方法相似，但相形之下，由于首轮运算中包含了轮置换函数，所以明文攻击要比密文攻击复杂度高. 本文选择明文攻击评测 Piccolo 密码算法抗功耗分析的能力，攻击目标是获取首轮轮密钥 RK_{0L}、RK_{0R} 和白化密钥 WK_0，WK_1(为解释方便，下文将 WK_0、WK_1、RK_{0L} 和 RK_{0R} 统称为攻击密钥)，针对明文的 CPA 攻击主要分为以下 4 个步骤：

1) 利用 HDL 语言完成 Piccolo 算法的硬件设计.

2) 采集不同明文加密时的功耗信息，建立矩阵 $\boldsymbol{P}_{act}$，同时记录对应的明文.

3) 基于汉明距离建立假设功耗模型，建立假设功耗矩阵 $\boldsymbol{P}_{hyp}$. 利用明文和密钥猜测值推算出加密过程的某一中间值，将每一条明文的该过程映射为功耗信息，形成假设功耗矩阵 $\boldsymbol{P}_{hyp}$. 这一步是能否成功实施 CPA 攻击的关键.

4) 对 $\boldsymbol{P}_{act}$ 和 $\boldsymbol{P}_{hyp}$ 进行数学统计分析，完成对攻击密钥的攻击，获得攻击密钥的最可能值.

2 攻击模型

2.1 Piccolo 算法的硬件实现

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[7]. 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，这种方法能够显著的减小硬件资源消耗，683GE 即可实现[4]. 在基于轮的并行实现方法中，Piccolo－80 每轮计算的 64 位中间结果被记录在触发器 DFF(0:63)中(本文用 DFF(0)表示这些触发器的最高有效位)，由于这里选择了明文攻击，因此只关心触发器在首轮的数据变化. Piccolo－80 的首轮的硬件抽象如图 1.

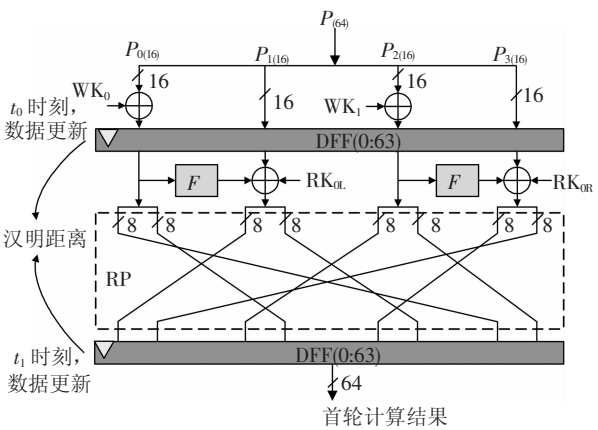


图 1 Piccolo 密码算法首轮运算硬件抽象

2.2 假设功耗矩阵建模

根据 CMOS 电路的固有属性，当触发器的值发生变化时将产生功耗，因此在 t_1 时刻前后一段时间的功耗，可以用触发器翻转的个数予以表示，即可以对其使用触发器翻转前后(图 1 中的深色部分)的汉明距离进行功耗建模.

由于密钥未知，在某一固定明文下，遍历所有

可能的密钥值，根据该功耗模型获取这一加密过程在某一时刻的假设功耗信息，之后，再通过换取不同的明文执行上述过程构成假设功耗矩阵. 如果对 N 个明文进行计算，所有密钥遍历位数为 k，可以得到一个 $N \times 2^k$ 的假设功耗矩阵 $\boldsymbol{P}_{\text{hyp}}$.

根据图1，触发器 $\text{DFF}(0:63)$ 在 t_0 时刻的值受到明文 P 和 WK_0、WK_1 的影响，而后 $\text{RK}_{0\text{L}}$、$\text{RK}_{0\text{R}}$ 的不同造成了触发器在 t_1 时刻值的不同，因此只需对 WK_0、WK_1、$\text{RK}_{0\text{L}}$、$\text{RK}_{0\text{R}}$ 进行 $2^{(16+16+16+16)} = 2^{64}$ 次遍历，即可完成对假设功耗矩阵的建模. 然而，这种方法工作量巨大，在有限的时间内无法完成，对攻击几乎没有意义.

由于部分触发器的功耗与总功耗也会存在一定的相关性，为了方便计算，采用分而治之的思想，我们可以基于 $\text{DFF}(0:63)$ 中部分触发器进行功耗建模，将 WK_0，WK_1，$\text{RK}_{0\text{L}}$，$\text{RK}_{0\text{R}}$ 按下式进行重新排列.

$$\begin{aligned}
\text{SubKey}_{1(20)} &= \{\text{WK}_0(0:15), \text{RK}_{0\text{L}}(0:3)\},\\
\text{SubKey}_{2(4)} &= \text{RK}_{0\text{L}}(4:7),\\
\text{SubKey}_{3(20)} &= \{\text{WK}_1(0:15), \text{RK}_{0\text{L}}(0:3)\},\\
\text{SubKey}_{4(4)} &= \text{RK}_{0\text{R}}(4:7),\\
\text{SubKey}_{5(8)} &= \text{RK}_{0\text{L}}(8:15),\\
\text{SubKey}_{6(8)} &= \text{RK}_{0\text{R}}(8:15).
\end{aligned}$$

之后对六段子密钥 $\text{SubKey}_{1(20)}$、$\text{SubKey}_{2(4)}$、$\text{SubKey}_{3(20)}$、$\text{SubKey}_{4(4)}$、$\text{SubKey}_{5(8)}$、$\text{SubKey}_{6(8)}$ 分别建立假设功耗矩阵，逐个进行攻击. 这样，可以将攻击密钥的搜索空间降低到了 $(2 \times 2^{20} + 2 \times 2^4 + 2 \times 2^8)$，给计算创造了可能.

2.2.1　针对 $\text{SubKey}_{1(20)}$ 的假设功耗矩阵建模

由于首轮的 RP 置换，$\text{RK}_{0\text{L}}(0:3)$ 的不同对 $\text{DFF}(0:3)$ 在 t_1 时刻的值构成影响. 为攻击 $\text{RK}_{0\text{L}}(0:3)$，需要对 $\text{DFF}(0:3)$ 在所有相关轮密钥可能值下的汉明距离进行计算. 由于 $\text{DFF}(0:3)$ 不仅受到 $\text{RK}_{0\text{L}}(0:3)$ 的影响，还要受到非线性函数 F 输出的制约，因此 WK_0 同样影响 $\text{DFF}(0:3)$ 的值. 通过对 $\text{RK}_{0\text{L}}(0:3)$ 和 WK_0 的值（即上文中的 $\text{SubKey}_{1(20)}$）进行遍历，即可通过 $P(0:15)$ 和 $P(16:19)$ 恢复出触发器 $\text{DFF}(0:3)$ 在不同的子密钥下 t_0 和 t_1 时刻的值，这样就完成了 $\text{DFF}(0:3)$ 汉明距离的计算. 攻击模型如下：

$$\begin{aligned}
\text{DFF}_{t_0}(0:3) &= P(0:3) \oplus \text{WK}_{0\text{g}}(0:3),\\
F_{\text{out}} &= F(P(0:15) \oplus \text{WK}_{0\text{g}}),\\
\text{DFF}_{t_1}(0:3) &= F_{\text{out}}(0:3) \oplus P(16:19) \oplus \text{RK}_{0\text{Lg}}(0:3),\\
HP(0:3) &= HD(\text{DFF}_{t_0}(0:3), \text{DFF}_{t_1}(0:3)) =\\
&\quad HW(\text{DFF}_{t_0}(0:3) \oplus \text{DFF}_{t_1}(0:3)).
\end{aligned}$$

式中：$F_{\text{out}}(0:3)$ 表示 F 函数输出的高4位；$\text{WK}_{0\text{g}}$ 表示白化密钥 WK_0 的猜测值；$\text{RK}_{0\text{Lg}}(0:3)$ 表示轮密钥 $\text{RK}_{0\text{L}}$ 高4位的猜测值. $\text{WK}_{0\text{g}}$ 与 WK_0 同样具有16位宽度，所以 $\text{WK}_{0\text{g}}$ 将会有 2^{16} 个可能的猜测值，根据上述模型，通过对 $\text{SubKey}_{1(20)}$ 的 2^{20} 次遍历可以得到一个 1×2^{20} 的汉明距离矩阵，这个矩阵代表了在不同子密钥猜测下，触发器翻转时刻的猜测的功耗信息，如果对 N 个明文进行计算，则可以得到一个 $N \times 2^{20}$ 的矩阵，这个矩阵即为我们攻击 $\text{SubKey}_{1(20)}$ 所需的假设功耗矩阵 $\boldsymbol{P}_{\text{hyp1}}$，利用该矩阵和后文的统计分析技术即可得到 WK_0 和 $\text{RK}_{0\text{L}}(0:3)$.

2.2.2　针对 $\text{SubKey}_{2(4)}$ 的假设功耗矩阵建模

在完成了 WK_0 和 $\text{RK}_{0\text{L}}(0:3)$ 的攻击后，WK_0 已经成为了已知量，由于 RP 置换，$\text{RK}_{0\text{L}}(4:7)$ 影响了 $\text{DFF}(4:7)$ 在 t_1 时刻的值. 对 $\text{RK}_{0\text{L}}(4:7)$（即上文的 $\text{SubKey}_{2(4)}$）进行遍历，计算 $\text{DFF}(4:7)$ 在不同密钥猜测的情况下时钟沿前后的汉明距离，就得到了攻击 $\text{RK}_{0\text{L}}(4:7)$ 所需的假设功耗矩阵 $\boldsymbol{P}_{\text{hyp2}}$. 建模过程如下：

$$\begin{aligned}
\text{DFF}_{t_0}(4:7) &= P(4:7) \oplus \text{WK}_0(4:7),\\
F_{\text{out}} &= F(P(0:15) \oplus \text{WK}_0),\\
\text{DFF}_{t_1}(4:7) &= F_{\text{out}}(4:7) \oplus P(20:23) \oplus \text{RK}_{0\text{Lg}}(4:7),\\
HP(4:7) &= HD(\text{DFF}_{t_0}(4:7), \text{DFF}_{t_1}(4:7)) =\\
&\quad HW(\text{DFF}_{t_0}(4:7) \oplus \text{DFF}_{t_1}(4:7)).
\end{aligned}$$

由上述模型可知，基于 N 条明文并对 $\text{SubKey}_2(4)$ 进行遍历后得到 $N \times 24$ 的假设功耗矩阵 $\boldsymbol{P}_{\text{hyp2}}$.

2.2.3　针对 $\text{SubKey}_{3(20)}$ 和 $\text{SubKey}_{4(4)}$ 的假设功耗矩阵建模

对 WK_1 和 $\text{RK}_{0\text{R}}(0:3)$ 的攻击过程与对 WK_0 和 $\text{RK}_{0\text{L}}(0:3)$ 的攻击过程基本一致，唯一的区别就是这里需要对 $\text{DFF}(32:35)$ 的汉明距离进行建模以完成攻击.

在完成了 WK_1 和 $\text{RK}_{0\text{R}}(0:3)$ 的攻击后，WK_1 已经成为了已知量，对 $\text{RK}_{0\text{R}}(4:7)$ 的功耗建模与对 $\text{RK}_{0\text{L}}(4:7)$ 的功耗建模过程基本一致，所不同的是这里需要关注 $\text{DFF}(36:39)$ 的汉明距离.

2.2.4　针对 $\text{SubKey}_{5(8)}$ 和 $\text{SubKey}_{6(8)}$ 的假设功耗矩阵建模

在得到 WK_0 和 WK_1 后，即可通过如下两个等式针对 $\text{RK}_{0\text{L}}(8:15)$（即上文的 $\text{SubKey}_{5(8)}$）完成功耗建模，该过程相对比较简单.

$$\text{DFF}_{t_0}(40:47) = P(40:47) \oplus \text{WK}_1(8:15),$$

$$F_{\text{out}} = F(P(0:15) \oplus \text{WK}_0),$$

$$\text{DFF}_{t_1}(40:47) = F_{\text{out}}(8:15) \oplus P(24:31) \oplus \text{RK}_{0\text{Lg}}(8:15),$$

$$HP(40:47) = HD(\text{DFF}_{t_0}(40:47), \text{DFF}_{t_1}(40:47)) = HW(\text{DFF}_{t_0}(40:47) \oplus \text{DFF}_{t_1}(40:47)).$$

为攻击 $\text{RK}_{0\text{R}}(8:15)$（即上文的 $\text{SubKey}_{6(8)}$），需要关注 $\text{DFF}(8:15)$ 的汉明距离，其建模过程与 $\text{RK}_{0\text{L}}(8:15)$ 的功耗建模过程相似.

3 实验配置及 CPA 攻击结果

3.1 实验配置

功耗分析攻击不同于普通的侧重于平均功耗的功耗分析，它主要关注芯片工作过程中瞬态功耗与数据的相关性，密码电路工作时的瞬态功耗数据获取通常有两种方式：第一种是采用示波器对实际芯片进行功耗曲线测量[8]；第二种是采用功耗模拟工具，在设计阶段获取加密过程的功耗信息[9-10]. 第一种方法虽然更有说服力，但是不能在芯片设计周期评估密码芯片的功耗分析攻击安全性，为了能够准确、快速的研究 Piccolo 的抗功耗分析攻击特性，本文基于 SMIC 0.18 μm 1P6M Logic18 CMOS 工艺和 PrimeTime PX 功耗模拟工具，获取加密过程的功耗信息，攻击实验中所采用的实验条件与资源如表 1 所示. 功耗曲线获取流程如图 2 所示.

表 1 基于模拟功耗数据的 CPA 攻击实验条件

目 的	软件版本
逻辑级模拟器	ModelSim 6.1f
逻辑综合器	Design Compiler X-2005.09
功耗模拟器	PrimeTime PX C-2009.06
脚本语言	Bash Shell, Perl
攻击平台	MATLAB R2010b

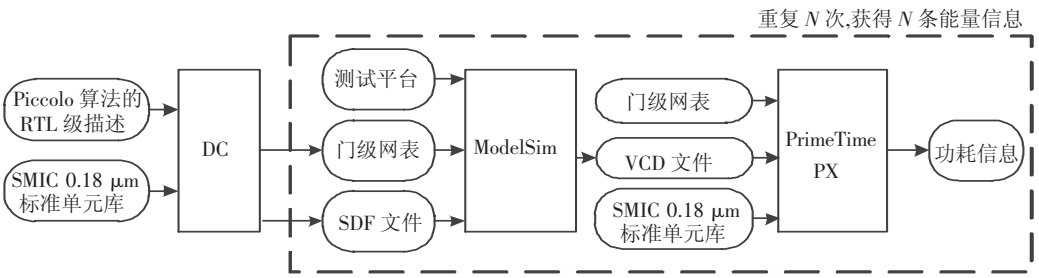


图 2 功耗数据采集模拟平台

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，重复执行图 2 中虚线框中的过程 N 次，则可以得到 $N \times T$ 个采样点，构成 N 行 T 列的实际功耗矩阵 $\boldsymbol{P}_{\text{act}}$.

3.2 攻击结果

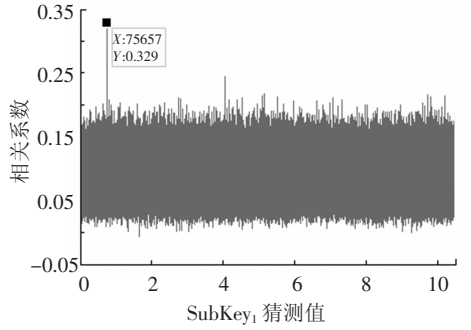
在功耗分析攻击中，功耗样本数量越多，攻击成功率就越高，因此常用 MTD（Measurements To Disclosure）代表为了正确破解密钥至少需要的功耗曲线数量，它经常用来衡量密码算法实现的抗功耗分析攻击的能力[10-11]. 依照第 1 部分中的 CPA 攻击过程，分别完成了对 6 段子密钥的攻击，攻击实验中，Piccolo 加密时的种子密钥 $\text{PK}_{(64)}$ 取 $\{123456\ 789\text{ABCDEF}12345\}_{16}$. 攻击结果显示，500 条功耗曲线足以恢复出白化密钥 WK_0 和 WK_1 以及首轮的轮密钥 $\text{RK}_{0\text{L}}$ 和 $\text{RK}_{0\text{L}}$.

由于对 WK_1 和 $\text{RK}_{0\text{R}}$ 的攻击与 WK_0 和 $\text{RK}_{0\text{L}}$ 的攻击过程基本一致，因此这里仅仅给出对 WK_0 和 $\text{RK}_{0\text{L}}$，即 SubKey_1、SubKey_2 和 SubKey_5 的攻击结果. 图 3 是针对 SubKey_1 的攻击结果. 其中，图 3(a) 表示在 500 条功耗样本下实施 CPA 攻击时，不同 SubKey_1 猜测值所对应的相关系数，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；图 3(b) 则表示在攻击成功时刻点附近，不同的 SubKey_1 猜测值的相关系数随功耗样本数量的变化曲线，它更加形象的表明了 SubKey_1 的抗功耗分析攻击的能力，从 50 条功耗样本开始，通过不断增加样本数量，逐次进行上述 CPA 攻击，直至能够稳定攻击出 SubKey_1，并由此推出 SubKey_1 的 MTD 值.

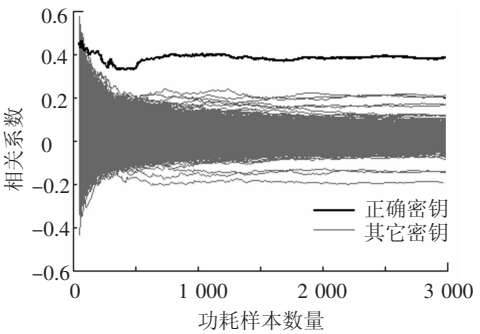
由图 3(a) 可以发现，当 $x = \{75657\}_{10} = \{12789\}_{16}$ 时，相关系数达到最大值 0.329，这说明在本次攻击中 $\{12789\}_{16} = \{0001\_0010\_0111\_1000\_1001\}_2$ 最有可能是 SubKey_1 的真实值，由此可推出 WK_0 的攻击密钥值为 $\{0001\_0010\_0111\_1000\}_2$，而 $\text{RK}_{0\text{L}}(0:3)$ 的攻击密钥值为 $\{1001\}_2$，事实上，WK_0 和 $\text{RK}_{0\text{L}}(0:3)$ 的真实密钥值也的确如此.

随着功耗曲线样本数量的增加，正确密钥与其它密钥的相关系数在总体趋势上都会有所降低，但是，与正确密钥相比，错误密钥的下降速度要来得快些，这一点可以由图 3(b) 可以看出，随着样本量的增加，正确 SubKey_1 猜测值与其它

$SubKey_1$ 猜测值的相关系数的区别不断加大，大约 200 条样本就已经可以成功破解 $SubKey_1$，即 $SubKey_1$ 的 MTD 值约为 200.



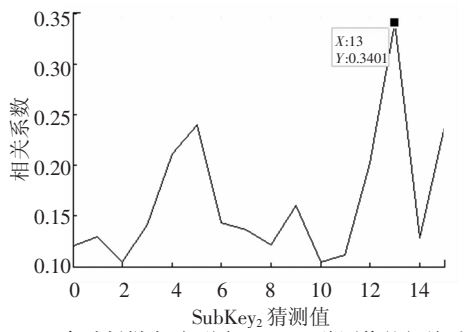
（a）500 条功耗样本时不同 $SubKey_1$ 猜测值的相关系数



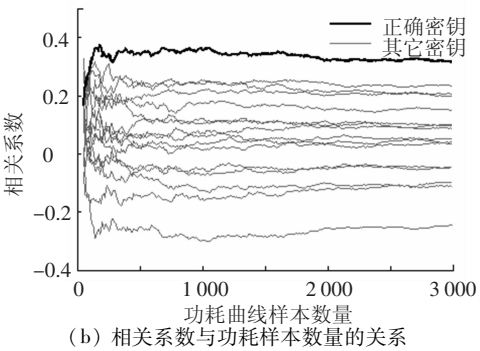
（b）相关系数与功耗样本数量的关系

图 3　针对 $SubKey_1$ 的 CPA 攻击结果

针对 $SubKey_2$ 和 $SubKey_5$ 也可依次完成上述两种实验，得到的 CPA 攻击结果如图 4 和 5 所示. 由图 4(a)，在 $x = \{13\}_{10} = \{1101\}_2$ 时获得了最大的相关系数 0.3401，即 $RK_{0L}(4:7)$ 的攻击密钥值为 $\{1101\}_2 = \{D\}_{16}$. 依据图 4(b)，可以得到 $SubKey_2$ 的 MTD 约为 300.



（a）500 条功耗样本时不同 $SubKey_2$ 猜测值的相关系数

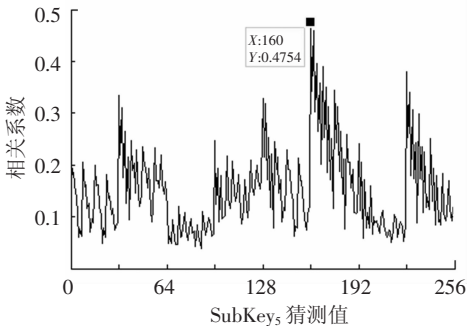


（b）相关系数与功耗样本数量的关系

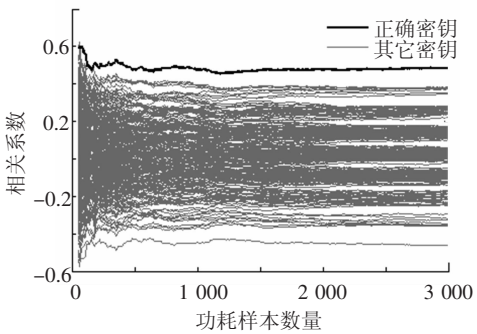
图 4　针对 $SubKey_2$ 的 CPA 攻击结果

对 $SubKey_5$ 的攻击结果如图 5. 从图 5 可得出 $RK_{0L}(8:15)$ 的攻击密钥值为 $\{1010\_0000\}_2 = \{A0\}_{16}$；$SubKey_2$ 的 MTD 值约为 200. 在图 5(a)中主峰与次峰相差较小(分别为 0.475 4 与 0.459 6)，在基于本攻击模型和实测功耗曲线进行 CPA 攻击时可能会被测量噪声淹没，此时可以通过增加样本数量的方法提高攻击成功率.

结合对 $SubKey_{3(20)}$、$SubKey_{4(4)}$ 和 $SubKey_{6(8)}$ 的攻击实验，500 条功耗曲线足以成功攻击上述 6 段子密钥. 综上，对 Piccolo 进行首轮的 CPA 攻击后得到 $RK_{0L} = \{9da0\}_{16}$，$WK_0 = \{1278\}_{16}$，这些结果与预期值相同，表明攻击成功. 同时我们也换取了别的密钥值，虽然所需功耗样本数量(MTD)会稍有不同，但 CPA 攻击同样能够成功，证明了所提出攻击模型的行之有效性.



（a）500 条功耗样本时不同 $SubKey_5$ 猜测值的相关系数



（b）相关系数与功耗样本数量的关系

图 5　针对 $SubKey_5$ 的 CPA 攻击结果

4　讨　论

4.1　种子密钥的获得

在完成对密钥 WK_0、WK_1、RK_{0L} 和 RK_{0R} 的攻击后，能够容易地得到 Piccolo 的 80 位种子密钥中的 64 位，只有 PK(64:79) 是未知的，此时可以基于一对明密文结合穷举的方法获得 PK(64:79)，由此，采用本文上述攻击模型，需要 $(2 \times 2^{20} + 2 \times 2^4 + 2 \times 2^8 + 2^{16})$ 次遍历计算即可获得 Piccolo 的 80 位种子密钥.

4.2　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，$SubKey_1$ 和 $SubKey_2$ 分别依赖于 DFF(0:3) 和 DFF(4:7)，而 $SubKey_5$ 则有赖于 DFF(40:47) 这 8 个触发器，因此攻击 $SubKey_5$ 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 $SubKey_5$ 时的相关系数(0.4754) 比攻击 $SubKey_1$ 和 $SubKey_2$ 时的相关系数(分别是 0.329 和 0.3401) 要高.

4.3 密钥搜索空间

在上述攻击模型中，将 WK_0，WK_1，RK_{0L}，RK_{0R} 重排为 6 段子密钥，由此得到的攻击密钥搜索空间为 $(2\times2^{20}+2\times2^4+2\times2^8)$；需要指出，这种重排方式并不唯一，也能按照如下方式重排为 4 段子密钥建立模型实施攻击：

$$SubKey_{1(20)} = \{WK_0(0:15), RK_{0L}(0:3)\};$$

$$SubKey_{2(20)} = \{WK_1(0:15), RK_{0R}(0:3)\};$$

$$SubKey_{3(12)} = RK_{0L}(4:15);$$

$$SubKey_{4(12)} = RK_{0R}(4:15).$$

这种组合方式造成攻击密钥的搜索空间为 $(2\times2^{20}+2\times2^{12})$，比本文采用的方式要大. 为了获得更小的密钥搜索空间，并降低内存空间复杂度，也可以将 WK_0，WK_1，RK_{0L}，RK_{0R} 重排为如下 8 段子密钥，并基于相应的触发器段进行建模，此时，可以将攻击密钥的搜索空间降低为 (8×2^8)，该攻击模型也已经通过实验证实了其可行性.

$$SubKey_{1(8)} = WK_0(0:7); SubKey_{2(8)} = WK_0(8:15);$$

$$SubKey_{3(8)} = RK_{0L}(0:7); SubKey_{4(8)} = RK_{0L}(8:15);$$

$$SubKey_{5(8)} = WK_1(0:7); SubKey_{6(8)} = WK_1(8:15);$$

$$SubKey_{7(8)} = RK_{0R}(0:7); SubKey_{8(8)} = RK_{0R}(8:15).$$

5 结 论

轻量级密码算法在硬件资源消耗与数学安全方面得到了有机的统一，但是轻量级密码算法同样也受到了功耗分析攻击的威胁，Piccolo 算法在首轮和最后轮中加入了白化密钥，这在一定程度上给功耗分析攻击增加了困难. 由于 Piccolo 属于 GFN 结构型密码算法，与传统密码算法 AES 和 DES 的功耗建模方式有所不同. 本文评估了 Piccolo 面向功耗分析攻击的安全性，提出了一种针对首轮的相关性功耗分析攻击模型，成功地完成了 Piccolo 算法的 80 位种子密钥的攻击. 攻击结果表明，在模拟功耗数据的情况下，只需 500 条功耗曲线即可完全恢复出 Piccolo－80 的种子密钥，而密钥搜索空间也从面向数学分析的 2^{80} 降低为面向实现的功耗分析攻击时的 $(2\times2^{20}+2\times2^4+2\times2^8+2^{16})$，由此可见，轻量级密码算法在面向功耗分析攻击时是脆弱的，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. 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] CLEFIA: a lightweight block cipher with a block size of 128 bits and a key size of 128, 192 or 256 bits[S]. ISO/IEC 29192－2:2012, 2012.

[2] PRESENT: a lightweight block cipher with a block size of 64 bits and a key size of 80 or 128 bits[S]. ISO/IEC 29192－2:2012, 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. Berlin: Springer-Verlag, 2007: 450－466.

[4] SHIBUTANI K. Piccolo: An ultra-lightweight blockcipher[C]//Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer-Verlag, 2011: 342－357.

[5] KOCHER P, JAFFE J, JUN B. Differential power analysis[C]//Proceedings of Advances in Cryptology—CRYPTO'99. Berlin: Springer-Verlag. 1999: 388－397.

[6] BRIER E, CLAVIER C, OLIVIER F. Correlation Power Analysis with a leakage model[C]//Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer-Verlag, 2004: 135－152.

[7] 唐明, 汪波, 杨欣, 等. 分组密码的硬件实现[J]. 哈尔滨工业大学学报, 2006, 38(9): 1558－1562.

[8] 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.

[9] 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): 134－238.

[10] ZHANG J, GU D W, GUO Z, *et al.* Differential power cryptanalysis attacks against PRESENT implementation [C]//Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering. NewYork: IEEE, 2010: V6－61－65.

[11] LIU P C, CHANG H C, LEE C Y. A Low Overhead DPA Countermeasure Circuit Based On Ring Oscillators [J]. IEEE Transactions on Circuits and systems-II, 2010, 57(7):547－550.

(编辑 苗秀芝)