doi: 10.3969/j.issn.1006-1576.2011.02.014

# 针对 RSA 密码芯片的 ZEMD 算法攻击实验

范黎恒，柏代军，张鑫
（重庆军事代表局，重庆 400060）

**摘要**：为了对 RSA 密码芯片的 DPA 攻击进行深入研究，采用相应攻击算法对 AT89C52 单片机上加密程序进行攻击，设计并搭建功耗分析测试平台。利用该平台，对单片机实现的 8 位模拟 RSA 加密算法进行 ZEMD 算法差分功耗分析（Differential Power Analysis，DPA）攻击实验。实验结果表明，由于明文的随机输入，使得模乘运算的时间消耗会有所不同，导致进行差分的功耗轨迹中对应部分无法准确对齐，ZEMD 攻击算法存在失效现象。

**关键词**：RSA；差分功耗分析；模幂算法；ZEMD 算法
**中图分类号**：TP309.7; TP301.6　　**文献标志码**：A

# Experiment of SEMD Algorithm Attack Against RSA Code Chip

Fan Liheng, Bai Daijun, Zhang Xin
(Chongqing Military Representative Bureau, Chongqing 400060, China)

Abstract: In order to research on DPA attack of RSA code chip, adopt corresponding attack method to attack encrypt program on AT89C52 single chip, a power consumption analysis testing platform was designed and constructed. With this platform, SEMD algorithm differential power analysis (DPA) attack experiment against the 8 bit simulated RSA running in a single chip was carried out. In the experiment, ZEMD attack algorithm became workless when the corresponding parts in the different power traces cannot be arranged. The reason of these phenomena was the random input of the plaintext.

Keywords: RSA; DPA; exponentiation algorithm; ZEMD algorithm

## 0 引言

从 Kocher 等人提出针对智能卡微处理器的功耗分析方法[1]以来，观测密码算法运行过程中芯片的功耗变化，从而获取其密钥的方法得以不断深入和发展。密码芯片被广泛应用于电子货币、数字签名和信息安全等领域。在信息社会密码芯片如何抵御功耗分析是十分重要的问题。目前，国内对RSA公钥密码算法的差分功耗分析（Differential Power Analysis，DPA）的研究还主要集中在理论阶段，为了进行更加深入的研究，根据功耗分析攻击原理，设计功耗测试平台，使用DPA[2]攻击方法中的ZEMD算法[3]，对RSA密码芯片进行攻击实验。

## 1 功耗分析攻击

典型的功耗分析攻击方法有2种，即简单功耗分析（SPA）与差分功耗分析（DPA）[2]。SPA是利用加密操作实现细节与功耗之间的关系直接从测量的功耗轨迹获取密钥信息，DPA是通过对大量的密文和功耗轨迹的统计分析获取密钥信息。

DPA 是建立在 SPA 基础之上的更具威胁的功耗分析攻击方法。在 SPA 中，一系列的指令操作会导致容易检测的、易于视觉观察的大规模功耗变化，若被运算的数据间的相互关系由于功耗变化小，检测出错或噪声干扰等原因被掩盖，SPA 所获得的就是毫无意义的平稳波形了。而 DPA 则会使用统计分析方法和纠错技术来提取密钥的相关信息，其分析过程可分为 2 个阶段：功耗轨迹采集和数据分析。在数据分析阶段采用统计学方法，引入一个区分函数，对大量的功耗轨迹点进行均值化和差分化，最后利用区分函数得出新的差分功耗轨迹图，从而得到相关的密钥 bit 位信息。

## 2 RSA 模幂算法

RSA 加密算法中，私钥 $d$ 必须严格保密，否则攻击者就可以使用私钥解密密文。在 RSA 加密算法中，与私钥相关的就是模幂算法。因此，模幂算法的安全性是十分重要的。

在许多公钥密码系统中，模幂运算不仅是最重要的算术运算，同时也是最耗时、最敏感的运算。因此，开发一个既有效又安全的模幂算法对于公钥密码系统是最为重要的。目前，使用最为广泛的模幂算法是“二进制模幂算法”，也称为“平方－乘积”（Square And Multiply）算法[4]，即将模幂分解

收稿日期：2010-09-14；修回日期：2010-11-18
基金项目：国家 863 计划项目（2007AA01Z454）
作者简介：范黎恒（1984—），男，山东人，硕士，从事信息安全研究。

成一系列的平方和乘积运算。对于基本的平方和乘积操作，一般采用 Montgomery 模乘算法来实现。“平方－乘积”算法主要有 2 种不同形式，分别是“从左至右平方－乘积”算法和“从右至左平方－乘积”算法，如表 1。

表 1 “平方－乘积”算法的 2 种形式

| 算法 1：从左至右“平方－乘积”模幂算法 | 算法2：从右至左“平方－乘积”模幂算法 |
|---|---|
| 输入：$m$，正整数 $k=(k_i k_{i-1}\ldots k_1 k_0)_2$<br>输出：$m^k \mod n$ | 输入：$m$，正整数 $k=(k_i k_{i-1}\ldots k_1 k_0)_2$<br>输出：$m^k \mod n$ |
| 步骤：<br>1. $A\leftarrow 1$<br>2. 对于 $t$ 从 $i$到0，执行：<br>2.1 $A\leftarrow A^2 \mod n$<br>2.2 若 $k_t=1$，则 $A\leftarrow A\cdot m \mod n$<br>3. 返回（$A$） | 步骤：<br>1. $A\leftarrow 1$，$S\leftarrow m$<br>2. 对于 $t$ 从0到 $i$，执行：<br>2.1 若 $k_t=1$，则 $A\leftarrow A\cdot S \mod n$<br>2.2 $S\leftarrow S^2 \mod n$<br>3. 返回（$A$） |

## 3 ZEMD 攻击算法

针对模幂算法的 DPA 攻击首先是由 Messerges 等人提出的[3]，他们共提出了 3 种针对“平方－乘积”模幂算法的 DPA 攻击方法，ZEMD 算法是其中之一。ZEMD 算法称为零指数多输入（Zero-Exponent Multiple-Data）攻击算法。

表 2 ZEMD 攻击算法

输入：$L$ 个随机明文 $m$
输出：私钥值 $d$

步骤：

1. 初始化：$d'\leftarrow 0$，$L_1\leftarrow 0$，$L_2\leftarrow 0$。
2. 对$i$=$n$-1到$i$=0，执行循环：
   ①猜想 $d'$ 的第 $i$ bit 位为 1；
   ②对$k$=1到$k$=$L$，执行循环：
   a、输入随机明文 $m$；
   b、模拟 $m^{d'}(\mathrm{mod}\ n)$ 的第 $i$ 步操作后的功耗即计算第 $i$ 步操作后中间值的汉明重量 $W(x_i)$；
   c、if 乘法运算后的中间值具有较大的汉明重量，则采集此输入的真实功耗 $S[j]$，并使 $S_{high}[j]\leftarrow S_{high}[j]+S[j]$，$L_1++$；
   if 乘法运算后的中间值具有较小的汉明重量，则采集此输入的真实功耗 $S[j]$，并使 $S_{low}[j]\leftarrow S_{low}[j]+S[j]$，$L_2++$；
   ③计算 $\overline{S_{high}[j]}=\frac{1}{L_1}S_{high}[j]$ 和 $\overline{S_{low}[j]}=\frac{1}{L_2}S_{low}[j]$；
   ④计算差分功耗 $D[j]=\overline{S_{low}[j]}-\overline{S_{high}[j]}$；
   ⑤if $D[j]\neq 0$，则 $d'_i\leftarrow 1$；
   if $D[j]\approx 0$，则 $d'_i\leftarrow 0$；
   ⑥Update $d'$。
3. return $d\leftarrow d'$。

ZEMD 算法在原理上更多地运用了仿真和统计的办法，在提出 $d'$ 的第 $i$ 位 bit 值 $d'_i$ 猜想后，对随机信息 M 进行指数为 $d'$ 的模幂运算的功率消耗仿真，仿真根据第 $i$ 次平方－乘法运算后的中间值的汉明重量来划分采集的功耗轨迹。如果乘法操作在第 $i$ 次平方－乘法运算中发生，则私钥 $d$ 参与运算的功耗轨迹可依照仿真的汉明重量的高低准确地分为 2 部分，两者之差必然有尖峰值的存在；反之，乘法操作未发生，则私钥 $d$ 参与运算的功耗轨迹不能被假设该操作已发生后的功耗仿真所准确区分，此时的差分值就不会出现尖峰。其具体的算法描述如表 2。

## 4 ZEMD 算法攻击实验

在目标电路板上，由 AT89C52 单片机运行 RSA 加密程序。在目标电路板和稳压电源之间串连一个电阻 R，并由数字存储示波器 Tektronix DPO4032 通过测量电阻 R 上的压降的变化来观测单片机电路板的功耗变化。在 PC 机上用 LabView 编写了控制数字存储示波器的虚拟示波器，并由该虚拟示波器来控制数字存储示波器实时向 PC 机传输功耗波形数据，实现了数据采集的自动化。

依据算法 3，利用功耗测试平台，进行了 ZEMD 攻击实验。其实验操作步骤如下：

1) 对采样控制平台进行配置，采样长度为 10 000，采样时间为4 ms，采样次数为10 000。

2) 向单片机发送10 000个随机明文，采集对应的10 000条功耗轨迹，并存入文件。

3) 在VC++ 6.0编程环境下，仿真RSA加密过程。在仿真过程中，假设猜测的密钥位 $d'_i=1$，计算中间值的汉明重量大小，并根据汉明重量的大小将对应的真实功耗轨迹划分为2组。

4) 利用Matlab 7.0对划分后的两组功耗轨迹分别求平均后再作差。

5) 观察生成的差分功耗轨迹，如果功耗轨迹上 $d_i$ 所在的区域出现尖峰，说明密钥位猜测正确，$d_i=1$；如果没有出现尖峰，说明密钥位猜测错误，$d_i=0$。

6) 所有密钥位的猜测完成后，就重构了密钥，实验结束。

（下转第 61 页）

[8] Marcel Bergerman, Omead Amidi, James Ryan Miller, Nicholas Vallidis, and Todd Dudek. Cascade Position and Heading Control of a Robotic Helicopter[C]. USA: Proc of the 2007 IEEE/ RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, 2007: 135−140.

[9] Michael Trentini, Jeff K. Pieper. Model-Following Control of a Helicopter in Hover[C]. Proc of the 1995 IEEE International Conference on Control Applications, Dearborn, September, 1996: 7−12.

[10] Bilal Ahmed, Hemanshu R. Pota, and Matt Garratt. Flight Control of a Rotary wing UAV-A Practical Approach[C]. Proc of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, 2008: 5042−5047.

[11] 于志，赵佳，申功璋. 基于鲁棒观测器结构的直升机飞控系统设计[J]. 系统仿真学报，2007, 19(8): 1776−1779.

[12] Johnson E N, and Kannan S K. Adaptive trajectory control for autonomous helicopters[J]. AIAA Journal of Guidance, Control and Dynamics, 2005, 28(3): 524−538.

[13] Mettler B, Tischler M B, and Kanade T. Attitude control optimization for a small-scale unmanned helicopter[R]. Washington,D.C.:AIAA Guidance, Navigation and Control Conference, 2000.

[14] Hess R A, Gao C, and Wang S H. Generalized technique for inverse simulation applied to aircraft maneuvers[J]. AIAA Journal of Guidance, Control and Dynamics, 1991, 14(5): 920−926.

[15] 祁飞，刘成国. 无人机航迹跟踪控制与仿真[J]. 计算机仿真, 2006, 23(11): 75−78.

（上接第 48 页）

实验中用 $d=(11011001)_2$ 作为单片机中 RSA 加密算法的私钥，并采集相应的功耗轨迹等待进一步的划分。每次都猜测密钥位 $d_i'$ 为 1，中间值就可以选取每轮循环中完成乘法操作后的 $A$（即算法 1 中的 $A\leftarrow A\cdot m \bmod n$）。之所以选 $A$ 为进行划分的中间值，是因为 $A$ 的值既与密钥相关，又与明文相关，它的汉明重量最有可能与功耗存在一定的关系。

图 1 是猜测密钥第一位 $d_1'=1$ 时，经过仿真分组，再利用 Matlab 7.0 进行计算后得到的差分功耗轨迹图。从图 1 中可以看出在 $d_1'=1$ 所在的区域里出现了一个很明显的尖峰，这就说明猜测密钥第一位 $d_1'=1$ 是正确的 $d_2'=1$。

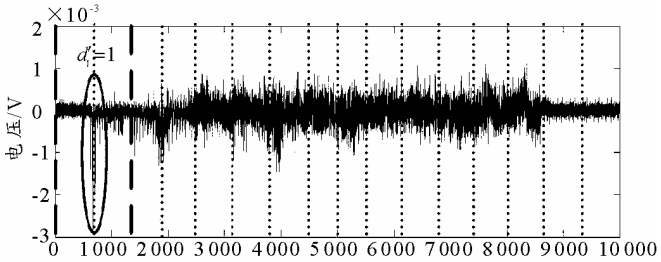


图 1　ZEMD 算法第一位实验结果

图 2 是猜测密钥第二位 $d_2'=1$ 时，经过仿真分组，再利用 Matlab 7.0 进行计算后得到的差分功耗轨迹图。图 2 中可以看出：在 $d_2'=1$ 所在的区域里没有出现明显的尖峰。但事实上 $d_2'$ 的猜测与输入的未知密钥是一致的，却没有获得预期的实验结果。通过分析发现是由于明文的随机输入，使得模乘运算的时间消耗会有所不同，导致功耗轨迹无法对齐，最终导致实验结果无法达到预期效果。由于第一位猜测受到功耗轨迹无法对齐的影响很小，所以第一位猜测的实验结果比较理想。

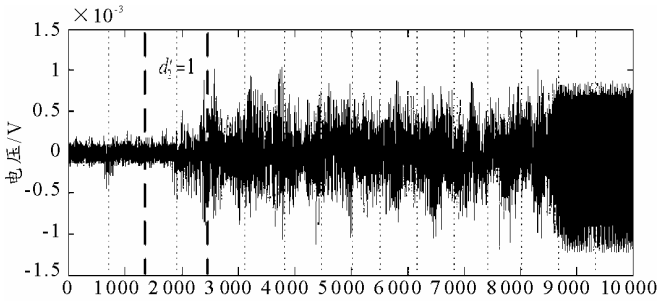


图 2　ZEMD 算法第二位实验结果

## 5　结论

针对单片机实现的 RSA 加密算法的 ZEMD 攻击实验在第一位的结果证明，在功耗轨迹能够对齐时，该攻击方法的可行性。而由于 FPGA 自身的运算特点，在 FPGA 上实现的 RSA 加密算法，不会出现单片机上类似的无法对齐的现象，因此，ZEMD 攻击算法是能取得很好的攻击效果的。但该实验在第二位以后存在失效现象，下一步将进行重点研究。

## 参考文献：

[1] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis[C]// In: M. Wiener, editor. Advances in Cryptology: Proceedings of CRYPTO'99. Volume 1666 in Lecture Notes in Computer Science, Santa Barbara, CA, USA: Springer-Verlag, 1999: 388–397.

[2] 韩军，曾晓洋，汤庭鳌. RSA 密码算法的功耗轨迹分析及其防御措施[J]. 计算机学报, 2006, 29(4): 590−596.

[3] T. S. Messerges, E. A. Dabbish, R. H. Sloan. Investigations of power analysis attacks on smartcards. Proc. USENIX Workshop on Smartcard Technology, 1999.

[4] D. E. Kunch. Seminumerical Algorithm. In the Art of Computer Programming, Vol. 2, Addison-Wesley, 1981.