

旁路模板在密码芯片指令分析中的应用

邓高明, 吴恒旭, 张 鹏, 邹 程, 陈开颜
(军械工程学院 计算机工程系, 河北 石家庄 050003)

摘 要: 集成电路微控制器芯片在执行不同的指令时会产生不同的功耗旁路泄露特征, 根据这些特征用统计方法构建对应不同指令以及指令执行路径的旁路模板, 并根据极大似然方法匹配目标芯片工作时泄露的功耗旁路信号, 进而推断芯片中程序的执行路径甚至指令代码。针对微控制器(AT89C52)上实现的 RSA 二进制模幂算法的简单旁路模板分析验证了旁路模板分析对于推断程序执行路径的可行性, 对微控制器典型指令的统计旁路模板分析实验说明了旁路模板分析在芯片指令逆向工程中的可行性。

关键词: 逆向工程; 旁路分析; 模板分析; 密码芯片

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2011)02-0140-05

Side Channel Template in Retrieve Cipher Chips' Instructions

DENG Gao-ming, WU Heng-xu, ZHANG Peng, ZOU Cheng, CHEN Kai-yan

(Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: There are different power consumption leakage characters for different instructions executed in IC chips. These side channel characters can be used to construct templates for different instructions and different paths a program took at a certain time. And then the side channel signals obtained from an unknown chip would be matched with these side channel templates, and the according to the maximum likelihood theory the best matched candidate may be the most likely correct instruction or path. The simple side channel template analysis against the binary modular exponentiation implementation of RSA executed in a microcontroller (AT89C52) validated the effective of the side channel template analysis in programs' path detection, and the static side channel template analysis against several typical instructions showed that the side channel template analysis is useful and powerful in reverse engineering for cipher chips.

Key words: reverse engineering; side channel analysis; template analysis; cipher chip

1 引言

密码芯片是实现密码算法的重要载体。当前我国进口先进生产设备及武器装备中大量地应用了以微控制器等集成电路芯片为核心的嵌入式密码系统, 其代码的逆向工程对国家和军事安全有重大意义。但微控制器密码芯片中存储的代码大都是经过加密的, 这是代码逆向工程的一大障碍。近年来新出现的旁路攻击 (Side Channel Attacks, SCA) 方

法^[1], 通过对密码设备工作时泄露的时间、功耗等旁路泄露信息的分析, 获取密码芯片的密钥或相关秘密信息, 成为密码分析方面的热点, 模板分析^[2-3]更是新兴的旁路攻击形式之一。最近, 旁路攻击方法开始运用于逆向工程领域^[4-5]。结合逆向工程的思想, 运用模板分析的旁路攻击方法对微控制器的指令建立模板, 可以获取密码芯片所运行程序的关键指令, 并分析其代码执行路径。

文中在 AT89C52 微控制器上实现密码算法作

收稿日期: 2010-02-09; 修回日期: 2010-04-02

基金项目: 国家自然科学基金项目 (60940019); 国家“八六三”计划项目 (2007AA01Z454); 军械工程学院原始创新基金 (YSCX0903)

为密码芯片实例,以 CMOS 门电路功耗泄露为旁路泄露的代表,在分析功耗泄露机理的基础上,介绍了旁路模板分析方法的原理,将其应用于分析以密码芯片的指令执行路径,并通过旁路泄露推断芯片中执行的指令。

2 密码芯片功耗泄露

文中以 CMOS 反相器为例分析微控制器芯片的功耗泄露机理及模型。

图 1 给出的反相器是 CMOS 芯片的基本单元,其工作时的功耗 P_{tot} 可表示为 $P_{\text{tot}} = P_{\text{dyn}} + P_{\text{dp}} + P_{\text{stat}}$. 其中 P_{dyn} 为负载电容充放电导致的动态功耗, P_{dp} 为直接通路电流引起的功耗, P_{stat} 为静态功耗. 静态 CMOS 在稳态工作状态时几乎没有功耗, P_{stat} 只占 P_{tot} 的 1%^[6], P_{dp} 是 PMOS 和 NMOS 同时导通的瞬间产生的功耗,占 P_{tot} 的 20%,而 P_{dyn} 是倒相器动态工作时功耗的主要部分^[6],也是旁路攻击所关注的与数据处理相关的部分,下面重点考虑起支配作用的 P_{dyn} . 动态功耗为能耗与负载电容充电频率的乘积 $P_{\text{dyn}} = C_L V_{\text{DD}}^2$. 将 CMOS 电路具有的寄生电容看成在器件端点之间观察到的一个集总等效电容^[7],称其为负载电容,如图 1 所示. 负载电容 $C_{V_{\text{DD}}}$ 与 C_{GND} 经历的充放电周期相位相反^[6]. 表 1 给出了 CMOS 门动态功耗的数据相关性,即为不同数据(状态)变化导致不同电能消耗。

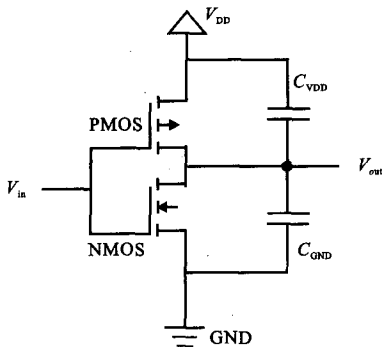


图 1 CMOS 反相器

表 1 CMOS 反相器动态功耗

V_{out}	V_{DD} 和 GND 端	动态功耗
0→1	C_{GND} 充电, 有电流	$P_{0 \rightarrow 1} = C_{\text{GND}} V_{\text{DD}}^2$
1→0	$C_{V_{\text{DD}}}$ 充电, 有电流	$P_{1 \rightarrow 0} = C_{V_{\text{DD}}} V_{\text{DD}}^2$
1→1	无电容充电, 无电流	$E_{1 \rightarrow 1} \approx 0$
0→0	无电容充电, 无电流	$E_{0 \rightarrow 0} \approx 0$

上述分析可知, CMOS 反相器的功耗与其发生翻转的概率有关. CMOS 集成电路微控制器中寄存

器的基本结构单元就是 CMOS 反相器, 因此微控制器芯片在某个时刻总的功耗与该时刻用到的寄存器个数相关, 即与该时刻发生翻转的反相器个数相关^[8]. 微控制器中不同的指令要用到不同的寄存器, 因而会产生不同的功耗泄露特征, 这就为利用功耗泄露反推指令代码提供了可能。

3 简单模板分析获取执行路径

代码执行路径的判断是微控制器指令逆向工程的一种情形, 即在已知代码的情况下, 如何判断运行过程中各个跳转分支的走向. 这在密码分析中有很重大意义, 因为某些密码算法的执行路径是和密钥位密切相关的, 判断代码的执行路径即可推断密钥, RSA 二进制模幂算法就是很好的例子。

3.1 RSA 二进制模幂算法

RSA 密码算法是公钥密码系统中的经典算法, 有多种实现形式, 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, 即将模幂运算分解成一系列的平方与乘积的连立运算. 下面简单介绍该算法的一个代表: “从左至右平方—乘积”算法。

输入: 明文 p , 密钥 $k = (k_i k_{i-1} \dots k_1 k_0)_2, n >$

输出: 密文 $c = p^k \bmod n$

1. $A \leftarrow p$
2. 对于 t 从 $i-1$ 递减到 0, 执行:
 - 2.1 $A \leftarrow A^2 \bmod n$
 - 2.2 若 $k_t = 1$, 则 $A \leftarrow A \cdot m \bmod n$
3. 返回 ($c = A$)

3.2 跳转指令的简单旁路模板分析

从算法可以看出, 在密钥位 $k_i = 0$ 时程序直接进入下一个循环, 而 $k_i = 1$ 时算法需要进行额外的乘法运算(步骤 2.2), 显然密码芯片在执行这两种不同的操作时消耗的功耗是不一样的, 在功耗泄露轨迹上会表现不同的特征, 即算法执行不同的路径会表现不同的功耗特征。

简单旁路模板分析就可以在已知密钥的情况下, 将密钥位为“0”和“1”时不同的代码执行路径对应的功耗特征作为模板, 对未知密钥情况下获取的一条功耗轨迹上不同的功耗特征进行模板匹配, 推断程序的执行路径, 进而推断各个密钥位的值。

为了说明简单旁路模板分析, 在 AT89C52 微控制器芯片上实现算法, 在芯片工作时用示波器采集芯片的功耗轨迹(具体参见文献[7]). 设置密钥为 $k = 1649 = (11001110001)_2$, 进行 RSA 二进制模幂

运算时的功耗轨迹如图 2 所示. if 条件语句跳转与否从功耗轨迹中可以看出明显的差别, 所以这样构建旁路模板可以有效地推断不同分支执行路径.

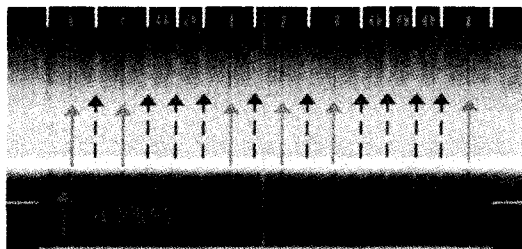


图 2 RSA 二进制模幂算法对应的功耗轨迹

4 统计模板分析获取指令代码

4.1 统计模板分析技术

对于程序执行路径能表现不同功耗特征的情况, 可以用简单旁路模板分析进行推断, 但要通过旁路特征反求密码芯片的指令则需要用到更高级的统计分析技术. 统计模板分析由两个步骤组成^[3]:

(1) 指令模板构建阶段

为了将功耗曲线按不同的指令分类, 需要预先为每个可能的指令都构建一个模板. 该模板反映了和指令对应的功耗曲线的统计特性, 也就是说模板反映了曲线上所有点的概率分布的特性.

由于指令执行的各状态期受到地址、累加器、数据等因素的影响, 造成不同的指令可能表现出相似的功耗特征. 假设密码芯片指令集有 r 条指令, 每条指令由于地址、累加器、数据等因素而造成的影响因子数目为 c , 因此需要为所有可能的指令构建 $r \times c$ 个模板. 用每个指令对同一个影响因子运算 m 次, 并获取对应的 m 条功耗曲线, 每条曲线有 n 个采样点, 即 $t_{i1}, t_{i2}, \dots, t_{in} (1 \leq i \leq m)$. 这样, 功耗曲线样本组成了一个矩阵 $t_{m \times n}$, 对应的均值向量计算如下:

$$\begin{aligned} \bar{t} &= \langle \bar{t}_1, \bar{t}_2, \dots, \bar{t}_n \rangle \\ &= \langle \frac{1}{m} \sum_{i=1}^m t_{i1}, \frac{1}{m} \sum_{i=1}^m t_{i2}, \dots, \frac{1}{m} \sum_{i=1}^m t_{in} \rangle \quad (1) \end{aligned}$$

因为所有的曲线都是从同一个操作获取的, 存在的随机噪声用样本与样本均值作差的方法求得. 对某个可能的子密钥的所有样本曲线, 对应的噪声矩阵 $N_{m \times n}$ 为:

$$N_{m \times n} = \begin{bmatrix} t_{11} - \bar{t}_1 & t_{12} - \bar{t}_2 & \cdots & t_{1n} - \bar{t}_n \\ t_{21} - \bar{t}_1 & t_{22} - \bar{t}_2 & \cdots & t_{2n} - \bar{t}_n \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} - \bar{t}_1 & t_{m2} - \bar{t}_2 & \cdots & t_{mn} - \bar{t}_n \end{bmatrix} \quad (2)$$

矩阵中的每一行组成了一条功耗样本曲线的噪

声向量, 每一列是反映该时间点的噪声的一个随机变量.

用协方差来描述两个随机变量之间的线性相关性, 两个随机变量 N_i 和 N_j 之间的协方差通过下式计算:

$$\text{cov}(N_u, N_v) = \frac{1}{m-1} \sum_{k=1}^m (t_{ku} - \bar{t}_u)(t_{kv} - \bar{t}_v) \quad (3)$$

为了描述两个以上随机变量之间的协方差, 需要一个协方差矩阵. 噪声信号的协方差矩阵定义如下:

$$\begin{aligned} C_{n \times n} &= \\ &= \begin{bmatrix} \text{cov}(N_1, N_1) & \text{cov}(N_1, N_2) & \cdots & \text{cov}(N_1, N_n) \\ \text{cov}(N_2, N_1) & \text{cov}(N_2, N_2) & \cdots & \text{cov}(N_2, N_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(N_n, N_1) & \text{cov}(N_n, N_2) & \cdots & \text{cov}(N_n, N_n) \end{bmatrix} \end{aligned} \quad (4)$$

因此, 一个模板定义为一个 2 元组:

$$T = \langle \bar{t}, C \rangle$$

为了创建这样一个模板, 需要采集大量的功耗曲线样本. 显然, 曲线样本量越大模板就越精确.

(2) 指令模板匹配阶段

通过一个被完全控制的芯片完成了所有指令模板的构建后, 用被攻击的芯片对模板构建阶段中用到的指令进行运算, 并采集对应的芯片运行时的功耗曲线 t' , 与先前构建好的模板进行比较. t' 与模板 $\langle \bar{t}, C \rangle$ 匹配的概率可通过式(5)求得. 根据极大似然法则, 概率最大的就是匹配得最好的模板, 对应的就是最可能的正确指令.

$$\begin{aligned} p(t'; \langle \bar{t}, C \rangle) &= \frac{1}{\sqrt{(2\pi)^m |C|}} \cdot \\ &\exp\left(-\frac{1}{2} (t' - \bar{t})^T C^{-1} (t' - \bar{t})\right) \end{aligned} \quad (5)$$

4.2 AT89C52 芯片指令周期分析

文中以 AT89C52 微控制器芯片为例说明统计旁路模板分析方法. 实验中在 AT89C52 中运行单周期常用指令并对其进行分析. 虽然这 and 实际芯片中运算的各种指令有所差别, 但从指令分析的角度来看, 情况是非常类似的. 单周期指令执行时间更短, 分析起来难度更大, 而单周期指令总数占 AT89C52 指令集指令总数的一半以上, 实验中选用单周期指令来说明指令分析的有效性更有代表性.

AT89C52 芯片一个机器周期包含 6 个状态期 (12 个振荡周期), 每个状态周期划分为 2 个节拍,

分别对应2个节拍时钟振荡期间。

在 AT89C52 芯片指令系统中,根据各种操作的繁简程度其指令可由单字节、双字节和三字节组成;从执行指令的时间上来看可以分为单周期、双周期和四周期,只有乘法指令占四周期。图3显示的是单周期指令的取指和执行时序。由图可知在每个机器周期内,地址锁存控制 ALE 两次有效,第一次出现在 S1P2 和 S2P1 期间,第二次出现在 S4P2 和 S5P1 期间。

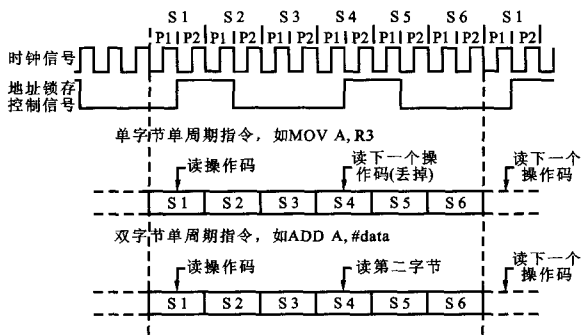
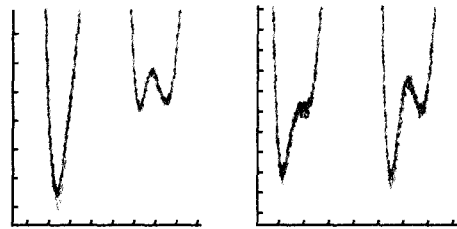


图3 典型单周期指令执行时序

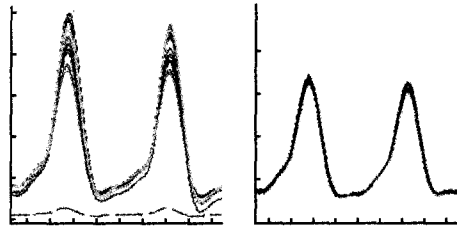
4.3 统计模板分析实验

由于指令在执行过程中不仅仅受到操作特性的影响,还受到许多诸如地址、累加器、寄存器等因素的影响。诸多因素的叠加使得同一指令在不同条件下各状态期呈现出不同的功耗轨迹,若要为每个指令在所有条件下的功耗轨迹都构建一个模板,会造成指令模板库过于庞大,指令模板匹配的正确率低,且工作量巨大。根据不同类型指令之间的区别对指令进行等价类划分,可以有效提高模板匹配的正确率,并降低工作难度。

在实际分析过程中,由于一条功耗曲线中样本点可能较多(n 可达到 10^3 到 10^6),可以从中选取少部分和指令操作相关的有效点(interesting points)组成功耗信号样本。选取指令机器周期 S1, S2 状态期功耗轨迹波谷的部分点作为有效点建立模板,可以分辨单周期与双周期(见图4(a)),双周期指令第二机器周期 S1, S2 状态期功耗轨迹波谷部分有明显尖峰);选取指令机器周期 S4 状态期功耗轨迹波峰的部分点作为有效点建立模板,可以作为单周期单字节指令与双字节指令区分依据(见图4(b)),单字节指令在机器周期 S4 状态期不做读操作,表现在功耗轨迹上明显比双字节指令低;单周期单字节指令周期中只有 S3 状态期受到指令本身的影响表现不同,可以选取其功耗轨迹上 S3 状态期波峰的部分点作为有效点。



(a) 单、双周期指令机器周期中 S1 状态期波谷之间区别



(b) 单周期单、双字节指令中 S4 状态期波谷之间区别

图4 指令功耗轨迹上部分典型特征

文中仅通过对单周期单字节指令的测试实验说明分析方法。选取下列四条指令: NOP; MOV A, R3; ADD A, @R1; DEC A。按照本节所述方法选取 S3 状态期4个点建立模板,根据4.1节介绍的模板创建以及匹配方法各进行500次匹配实验,试验结果如表1所示。

表1 匹配实验结果对比

指令	识别				正确率
	NOP	MOV A, R3	ADD A, @R1	DEC A	
NOP	500	0	0	0	100%
MOV A, R3	0	262	248	0	52.4%
ADD A, @R1	0	257	243	0	48.6%
DEC A	0	0	0	500	100%

通过实验可以看出,指令 NOP 和 DEC A 可以被准确的识别,MOV A, R3 和 ADD A, @R1 两条指令由于都涉及累加器与寄存器之间的操作,很难分辨开。在指令模板库建立的过程中可以将 MOV A, R3 和 ADD A, @R1 两条指令划为等价类相同看待,在具体分析时针对上下文进行推测。

5 结束语

文中结合逆向工程思想和模板攻击原理,利用密码芯片功耗泄露信号为芯片指令以及程序执行路径构建模板,将目标芯片上获取的旁路信号与模板进行匹配以推断芯片中执行的代码以及程序的执行路径。通过对 AT89C52 单片机上运行 RSA 二进制模幂算法的简单旁路模板分析和单片机单周期单字节指令的统计模板分析实验,证明了旁路模板指令

分析的可行性与有效性.

参考文献:

- [1] Paul Kocher. Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems[C]// Proceedings Crypto '96, LNCS. New York: Springer-Verlag, 1996:104-113.
- [2] 邓高明, 赵强, 张鹏, 等. 针对密码芯片的电磁频域模板分析攻击[J]. 计算机学报, 2009, 32(4): 602-610.
- [3] 邓高明, 张鹏, 赵强, 等. 针对密码芯片的电磁模板分析攻击[J]. 微电子学与计算机, 2010, 27(1): 1-4.
- [4] Martin Goldack. Side-channel based reverse engineering for microcontrollers [D]. Bochum: Ruhr-University, 2008.
- [5] Dennis Vermoen. Reverse engineering of Java card applets using power analysis [D]. Netherlands: TU Delft, 2007.
- [6] Jan M Rabaey, Anantha Chandrakasan, Borivoje Nikoli. 数字集成电路—电路、系统与设计[M]. 2 版. 周润德,

译. 北京: 电子工业出版社, 2004: 141-166.

- [7] SungMo Kang, Yusuf Leblebici. CMOS 数字集成电路—分析与设计[M]. 3 版. 王志功, 窦建华, 译. 北京: 电子工业出版社, 2005: 97-98.
- [8] 褚杰, 赵强, 丁国良, 等. 密码系统差分功耗分析攻击及功耗模型[J]. 华中科技大学学报: 自然科学版, 2008, 36(6): 18-21.

作者简介:

- 邓高明** 男, (1983-), 博士研究生. 研究方向为电磁信息检测与主动防护技术.
- 吴恒旭** 男, (1982-), 硕士研究生. 研究方向为信息安全.
- 张 鹏** 男, (1976-), 博士研究生. 研究方向为电磁信息检测与主动防护技术.
- 邹 程** 男, (1982-), 博士研究生. 研究方向为电磁信息检测与主动防护技术.
- 陈开颜** 女, (1970-), 博士, 硕士生导师. 研究方向为信息安全.

(上接第 139 页)

码, 1-D DCT 复用等措施来减少芯片面积, 通过优化电路结构, 加入 8 级流水线, 优化转置矩阵来提高电路速度. 仿真结果验证了算法的正确性. 最高工作频率可以达到 139.43MHz. 能够满足视频图像压缩的实时性要求.

参考文献:

- [1] Ahmed N, Natrajan T, Rao K R. Discrete cosine transform[J]. IEEE Transaction on Computers, 1974(C-23): 90-95.
- [2] Agostini L V, Saraiva I. Pipelined fast 2-D DCT architecture for JPEG image compression[C]// SBCCI'01, the 14th Symposium on Integrated Circuits and Systems Desing. Brazil: Pirenopolis, 2001: 226-231.
- [3] 王继东, 张会生, 范忠亮. 基于 NEDA 算法的二维 DCT

硬件加速器的设计实现[J]. 微电子学与计算机, 2008, 25(5): 166-168.

- [4] Loeffler C, Lightenberg A, Moschytz G. Practical fast 1-D DCT algorithms with 11 multiplications [C]// IC-ASSP'89, Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing. USA: New York, 1989, 2(5): 988-991.
- [5] Uwe Meyer Baese. 数字信号处理的 FPGA 实现[M]. 刘凌, 译. 北京: 清华大学出版社, 2006: 32.
- [6] 陈伟, 卢贵主, 郑灵翔. 二维 DCT 算法及其精简的 VLSI 设计[J]. 微电子学与计算机, 2008, 25(3): 108-110.

作者简介:

- 郭宝增** 男, (1953-), 教授, 硕士生导师. 研究方向为集成电路设计.