

基于 HMM 的微控制器旁路模板指令序列恢复

余 浩¹, 陈开颜¹, 张 阳¹, 邓高明², 吴恒旭³

(1 军械工程学院 计算机工程系, 河北 石家庄 050003; 2 北方电子设备研究所, 北京 100191;

3 成都军区 78086 部队, 四川 成都 610000)

摘 要: 针对微控制器代码旁路逆向恢复的问题, 采用逆向工程思想与旁路攻击方法, 依据不同的指令在芯片内执行时, 会产生不同的功耗旁路泄漏信号这一特点, 在已实现的单条指令旁路模板恢复的基础上, 综合考虑程序的“上下文”信息, 运用隐马尔可夫模型(HMM)对该问题进行建模描述与求解. 对 AT89C52 微控制器中运行的数据加密标准(DES)加密算法的部分指令序列的恢复实验表明, 该方法能够有效的恢复出微控制器芯片中运行的指令序列.

关键词: 旁路模板; 隐马尔可夫模型; 微控制器; 指令序列恢复

中图分类号: TN391

文献标识码: A

文章编号: 1000-7180(2012)08-0042-05

Retrieve Instruction Sequences with Side Channel Template Based on Hidden Markov Model in Microcontroller

YU Hao¹, CHEN Kai-yan¹, ZHANG Yang¹, DENG Gao-ming², WU Heng-xu³

(1 Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China;

2 Institute of North Electronic Equipment, Beijing 100191, China;

3 Chengdu Military Area Commands Unit 78086, Chengdu 610000, China)

Abstract: In order to exploit prior knowledge about the program code, a math model, Hidden Markov Model (HMM), is applied to describe and resolve a problem of recovering the program code of a microcontroller. With the idea of reverse engineering and the method of Side-Channel Attacks (SCA), and according to the property that different power consumption leakage characters for different instructions executed in chips, a practical experiment based on previous side channel templates built by making use of power consumption leakages for each instruction has shown that instruction sequences executed in a Data Encryption Standard (DES) microcontroller (AT89C52) cipher chip circuit can be reconstructed and has confirmed the effectiveness of method mentioned above.

Key words: side channel templates; HMM; microcontrollers; instruction sequences reconstructed

1 引言

由于微控制器芯片内部的指令代码都是安全地存储在芯片内存中的, 并且绝大多数芯片公司为了保证其知识产权, 防止产品被仿造, 甚至对指令代码进行了加密操作, 即使读取后也无法进行分析. 因此, 针对微控制器代码的逆向工程是困难的, 但在某些情况下却很有必要. 近年来, 旁路攻击 (Side

Channel Attacks, SCA) 领域中的旁路模板分析^[1]方法开始被运用于逆向工程领域^[2], 结合逆向工程的思想, 运用模板分析的方法对密码芯片指令执行时的旁路泄漏信号建立模板, 逆向密码芯片中所运行的秘密算法, 可以恢复芯片所运行的程序指令.

Goldack^[3]利用功耗模板分析对微控制器中密码算法代码的逆向工程和执行路径判别进行了研究. Vermoen^[4]介绍了如何通过旁路分析得到 Java

收稿日期: 2012-01-09; 修回日期: 2012-03-06

基金项目: 国家自然科学基金(60940019); 军械工程学院原始创新基金(YSCX0903)

卡上运行的指令代码,其用的方法是将未知指令对应的旁路轨迹与已知指令的比较来推断指令。国内方面,文献[5]验证了旁路模板分析在芯片指令逆向工程中的可行性,文献[6]对旁路模板指令恢复进行改进,提高了指令匹配的正确率。然而,这些方法关注的都是单个指令对应的旁路信号,实际上芯片指令运行在一个特定的程序中,各指令并不是单独存在的,而是会受到程序“上下文”的影响,有些指令前后紧邻的指令是有特殊要求的,这些程序“上下文”信息将有助于芯片指令序列的恢复。

本文以 AT89C52 微控制器为实际对象,在分析微控制器功耗泄漏与指令相关性基础上,对运行于其中的一段 DES 加密程序作为指令序列分析实例,利用对该微控制器已建立的单条指令模板库,考虑程序“上下文”信息的作用,将隐马尔可夫数学模型应用于微控制器指令序列的恢复。

2 微控制器功耗泄漏与指令的相关性

微控制器中寄存器的基本结构单元是互补金属氧化物半导体(Complementary Metal-Oxide-Semiconductor, CMOS)反相器,根据微控制器芯片功耗泄漏机理,CMOS 反相器的功耗与其发生翻转的概率有关,因此微控制器芯片在某个时刻总的功耗与该时刻发生翻转的反相器个数相关,即微控制器中不同指令执行时所对应的发生翻转的反相器个数不同,因而会产生不同的旁路功耗泄漏特征。

在 AT89C52 微控制器芯片指令系统中,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,只有乘除法指令占四周期。

根据表 1 所示,在相同条件下对表中指令序列重复执行 100 次并求出功耗轨迹平均值,得到平均轨迹如图 1 所示。可以看出单周期指令,双周期指令和四周期指令的执行时长不同,在功耗轨迹长度上有所反映,并且不同指令有着不同的轨迹形状,这就

表 1 不同指令组成的序列

指令序列	指令周期数
MOV R3,40H;	双周期
MOV R3,A;	单周期
ADD A,R3;	单周期
XRL 40H, #F0H;	双周期
MOV B,R3;	单周期
MUL AB;	四周期

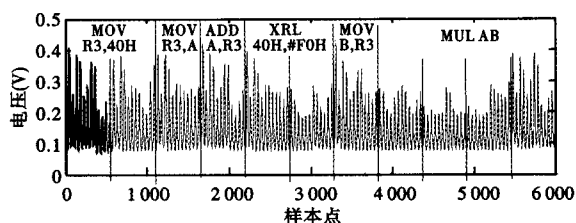


图 1 表 1 中指令序列所对应的功耗轨迹

为利用旁路功耗泄漏反推指令提供了可能。

3 指令序列分析

3.1 微控制器指令序列恢复问题定义

文献[5-6]针对单条指令的模板构建与匹配进行了详细阐述,得到了针对 AT89C52 微控制器指令集中各条指令的匹配结果,即不同指令根据旁路模板被匹配为各个指令的概率,也称单条指令的识别率。在对指令序列的恢复的过程中,由于各条指令并不是单独存在的,程序中前后执行顺序与相邻指令之间存在某些可以利用的规律,比如函数调用指令 LCALL 之前通常需要先程序现场压栈,LCALL 之前通常有 POP 指令,此外在给累加器 A 赋值(MOV A,X)之后一般不会紧接着做清空 A 的操作(CLR A)。这些程序“上下文”信息将有助于指令序列的恢复。

通过用大量随机明文和随机密钥进行微控制器仿真调试的方法,得到 DES 密码程序中各指令后续指令出现概率,表 2 给出了部分结果。

表 2 AT89C52 中 DES 密码程序部分
指令后续指令出现概率

当前指令	后继指令及对应的出现概率			
MOV Rn,A	ADD A, #data	MOV A,@Rn	DEC Rn	CLR A
	8.72%	6.36%	3.84%	1.28%
INC Rn	INC Rn	INC @Ri	LJMP	AJMP
	43.25%	33.76%	11.50%	11.49%
POP direct	LCALL addr16	POP direct	RETI	—
	42.85%	42.85%	14.3%	—
MOV B, #data	MUL AB	—	—	—
	100%	—	—	—
SWAP A	ANL A, #data	—	—	—
	100%	—	—	—
...	...	...	...	...

上述分析过程可以得到相邻两个指令的关系,为了分析程序中更长的指令序列,可以将微控制器指令执行过程建模成一个状态机,微控制器的指令

集就是状态机的状态集,指令的执行序列就是状态机的状态转移过程.指令序列恢复的目的就是要确定该状态机的状态转移过程.分析者无法直接获取状态机的状态转移过程,仅能通过指令执行时的旁路信号推测状态转移过程,根据上述描述将基于旁路模板的指令序列恢复问题形式化定义如下:

对于微控制器 C 有指令集 $S = \{s_1, s_2, \dots, s_M\}$, 已知指令 S_i 的后续指令为 S_j 的概率为 a_{ij} , 通过旁路模板分析将指令 S_i 判别为指令 S_j 的概率为 b_{ij} ($1 \leq i, j \leq M$). 在 C 上运行程序 P_x 的指令序列为 $(O_1, O_2, \dots, O_N)$, 用测量装置 M 采集到运行 P_x 对应的旁路泄漏信号轨迹 $T^{(x)} = (r_1^{(x)}, r_2^{(x)}, \dots, r_m^{(x)})$ (m 是旁路信号轨迹上的采样点数). 问如何根据 S, a_{ij}, b_{ij} 以及 $T^{(x)}$ 推断 P_x 中的指令序列 $(O_1, O_2, \dots, O_N)$.

上述定义描述的问题中包含两个随机过程, 一个是指令序列的执行顺序(这实际上是一个马尔可夫链), 另一个是对每个指令的概率性旁路观测, 这种问题可以用隐马尔可夫模型来描述.

3.2 隐马尔可夫模型及具体参数确定

隐马尔可夫模型^[7]以马尔可夫链为基础, 由于实际问题往往比马尔可夫链模型所描述的更为复杂, 观察到的事件并不与状态一一对应, 而是与一组概率分布相联系, 这样就产生了“隐马尔可夫模型”. HMM 是一个双重随机过程, 即状态到状态之间的转移是一个随机过程, 而每个状态的观测结果是另一个随机过程. 因此 HMM 可以看作由以下两个部分组成: (1) 马尔可夫链: 一个观测不到的有限状态链, 用来描述状态的转移, 这一部分由隐含状态组成, 其输出为状态序列; (2) 一般随机过程: 描述状态与观测序列之间的关系, 是与状态链相关的可观测到的随机变量, 这一部分由观察值序列组成.

这样, 站在观察者的角度只能看到观察值, 不能直接看到状态, 而是通过一个随机过程去感知状态的存在及其特性, 故称之为“隐”马尔可夫模型. 其结构如图 2 所示.

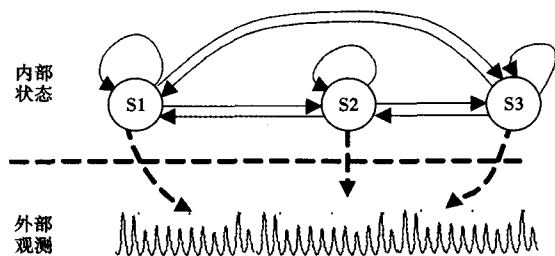


图2 指令执行序列旁路观测的隐马尔可夫模型

此外, 隐马尔可夫模型也具有马尔可夫链的遗忘功能, 当前状态只和前一个状态相关. HMM 模型主要由以下几个参数来描述:

(1) 模型中的状态数 N , 状态集 $S = \{S_1, S_2, \dots, S_N\}$, t 时刻的状态用 q_t 表示, 且 $q_t = S_i$, ($1 \leq t \leq T, 1 \leq i \leq N$); 相应的状态序列记为 $Q = \{q_1, q_2, \dots, q_T\}$;

(2) 模型中每个状态所对应的不同观察数 M , 记为 $V = \{v_1, v_2, \dots, v_M\}$; t 时刻的观察值用 O_t 表示, 且 $O_t = v_k$, ($1 \leq t \leq T, 1 \leq k \leq M$), 相应的观察符号序列记为 $O = \{O_1, O_2, \dots, O_T\}$;

(3) 状态转移概率矩阵 $A, A = (a_{ij})_{N \times N}$, 表示状态 S_i 的后继状态为 S_j 的概率, 即: $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$, ($1 \leq i, j \leq N$);

(4) 观察值概率矩阵 $B, B = (b_{jk})_{N \times M}$, 表示状态机处于状态 S_j 时观察值的概率分布, 其中 $b_{jk} = P(O_t = v_k | q_t = S_j)$, ($1 \leq j \leq N, 1 \leq k \leq M$), M 为每个状态对应的可能观测值数目;

(5) 初始状态概率分布向量 $\pi, \pi = (\pi_1, \pi_2, \dots, \pi_N)$, 其中 $\pi_i = P(q_1 = S_i)$, ($1 \leq i \leq N$).

因此, 可以记 HMM 为: $\lambda = (N, M, \pi, A, B)$. HMM 的应用可以归为下面三个基本问题:

(1) 认证问题: 给定 HMM 模型 $\lambda = (\pi, A, B)$ 和观测序列 $O = (O_1, O_2, \dots, O_T)$, 如何快速有效地计算条件概率 $P(O | \lambda)$, 衡量了给定模型随时间演变, 得到观测序列 O 的可能性;

(2) 解码问题: 给定 HMM 模型 $\lambda = (\pi, A, B)$ 和观测序列 $O = (O_1, O_2, \dots, O_T)$, 如何找到与之对应的最可能的状态序列 $q = (q_1, q_2, \dots, q_T)$;

(3) 训练(学习)问题: 给定观测序列 $O = (O_1, O_2, \dots, O_T)$, 如何调整模型参数 $\lambda = (\pi, A, B)$ 使概率 $P(O | \lambda)$ 最大. 该问题可以使得模型更好地适应于具体的观测序列, 从而达到使用于具体问题的目标.

在用 HMM 分析 AT89C52 指令序列恢复问题时, 需要确定 HMM 各参数, 对于模型中的状态数, 不能简单地以指令个数表示. 由于指令执行的最小公共时间单位为 1 个指令周期, 可以把 1 个指令周期看作 1 个状态, 单周期指令只有 1 个状态, 多周期指令有多个状态, 而且多周期指令前后指令周期的状态转移概率为 1. 如图 1 所示的 6 条指令对应的旁路信号轨迹中包含 11 个状态. 同样地, 需要为每个指令的每个指令周期构建 1 个指令模板. 据此,

HMM的各个参数确定如下:

(1) 模型中的状态数 N 即为微控制器的所有指令的指令周期数之和, 状态集 $S = \{S_1, S_2, \dots, S_N\}$, 即为所有指令的各个指令周期的执行动作的集合, t 时刻的状态 $q_t = S_i (1 \leq t \leq T, 1 \leq i \leq N)$ 表示 t 时刻执行的动作为某个指令的一个指令周期 S_i , 相应的状态序列 $Q = \{q_1, q_2, \dots, q_T\}$ 即为指令的执行序列;

(2) 模型中每个指令周期所对应的不同观察数 M 即为指令模板的个数, 在指令序列恢复问题中 $M = N$;

(3) 状态转移概率矩阵 $A = (a_{ij})_{N \times N}$ 是各个指令周期前后出现的概率, 对于不同指令, 该概率即为表2所示的各指令后续指令出现概率, 对于多周期指令的多个指令周期, 其第1个指令周期后出现其第2个指令周期的概率为1, 出现其它指令周期为0, 其余以此类推;

(4) 观察值概率矩阵 $B = (b_{jk})_{N \times M}$ 中的元素 b_{jk} 为某条指令根据旁路模板被匹配为指令集中各个指令的概率, 对于多周期指令, 需要为每个指令周期构建模板;

(5) 初始状态概率分布向量 $\pi = (\pi_1, \pi_2, \dots, \pi_N)$ 中的元素 $\pi_i = P(q_1 = S_i)$, $1 \leq i \leq N$ 是指状态机初始处于某个状态的概率, 这个值需要根据实际情况考虑. 在一般的C51程序中, 通常需要进行初始化操作, 在本文的实验中都是调用初始化函数来实现的, 因而 π 中仅有LCALL操作的概率为1, 其余操作的概率均为0.

3.3 微控制器指令序列恢复问题的求解

基于旁路模板的指令序列恢复是要通过观测值推断指令的执行序列, 该问题属于HMM的第二个问题——解码问题, 对该问题求解一般用Viterbi解码算法^[8].

定义 $\delta_t(i)$ 为时刻 t 时沿着一条路径 $q_1, q_2, \dots, q_t$, 且 $q_t = S_i$ 产生的观察值序列 $\{O_1 O_2 \dots O_t\}$ 的最大概率, 有:

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} P(q_1, q_2, \dots, q_t = S_i, O_1 O_2 \dots O_t | \lambda) \quad (1)$$

通过迭代有:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(O_{t+1}) \quad (2)$$

为了能够正确地回溯状态序列, 需要追踪那些能够使式(2)最大化的参数, 对每一个 t 和 j , 将其表示为序列 $\psi_t(j)$, 那么求解最优序列的Viterbi算法

描述如下:

① 初始化:

$$\delta_1(i) = \pi_i b_i(O_1), 1 \leq i \leq N$$

$$\Psi_1(i) = 0, 1 \leq i \leq N$$

其中 $\delta_1(i)$ 是 $t=1$ 时刻处于状态 S_i 时产生观察符号 O_1 的概率. 另一个变量 $\Psi_t(j)$ 用于存储每一时刻的最佳状态.

② 迭代:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)$$

$$\Psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

$$1 \leq j \leq N, 2 \leq t \leq T$$

③ 结束:

$$P(O, Q^* | \lambda) = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

④ 回溯, 即解码过程:

$$q_t^* = \Psi_{t+1}(q_{t+1}^*), t = T-1, T-2, \dots, 1$$

4 实验及结果分析

针对AT89C52上执行的DES加密程序, 选取其包含密钥信息的第一个S盒代换操作的部分程序段进行分析, 如表3所示为选取的部分指令序列. 图3所示为指令序列对应的功耗轨迹.

表3 DES加密程序中部分指令序列

序号	指令(指令周期)	序号	指令(指令周期)
1	MOV A, @R0	14	ADDC A, B(0xF0)
2	MOV R7, A	15	MOV DPH(0x83), A
3	DEC R0	16	MOV A, DPL(0x82)
4	MOV A, @R0	17	ADD A, #0x5E
5	MOV B(0xF0), #0x10(Q1)	18	MOV DPL(0x82), A
6	MOV B(0xF0), #0x10(Q2)	19	MOV A, DPH(0x83)
7	MUL AB(Q1)	20	ADDC A, #0x01
8	MUL AB(Q2)	21	MOV DPH(0x83), A
9	MUL AB(Q3)	22	CLR A
10	MUL AB(Q4)	23	MOVC A, @A+DPTR(Q1)
11	ADD A, R7	24	MOVC A, @A+DPTR(Q2)
12	MOV DPL(0x82), A	25	MOV R0, #0xD5
13	CLR A	26	MOV @R0, A

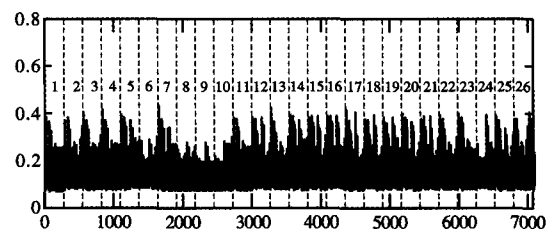


图3 表3中指令序列对应的功耗轨迹

按照文献[5-6]所述,对表 3 序列中的各条指令分别单独进行 100 次模板匹配,其实验结果如表 4 所示. 根据 HMM 模型及 Viterbi 解码算法,进行 100 次的目标指令序列恢复实验,实验结果如表 5 所示.

表 4 表 3 中各指令单独匹配结果

序号	识别率	序号	识别率	平均
1	62%	14	59%	61.4%
2	58%	15	62%	
3	58%	16	62%	
4	62%	17	59%	
5	62%	18	62%	
6	62%	19	62%	
7	64%	20	59%	
8	64%	21	62%	
9	64%	22	62%	
10	64%	23	61%	
11	59%	24	61%	
12	62%	25	60%	
13	62%	26	62%	

表 5 HMM 解码后指令序列恢复结果

次数	正确个数	错误个数	识别率	平均
1	22	4	84.6%	82.5%
2	20	6	76.9%	
3	22	4	84.6%	
...	...	...	...	
99	23	3	88.5%	
100	19	7	73.1%	

从实验结果可以看出,若利用文献[5-6]的单条指令模板匹配的方法对指令序列进行恢复,其平均正确率为 61.4%,而用本文的方法对以上指令序列进行恢复,其平均正确率有显著的提高,证明了该方法的有效性. 实验中,程序“上下文”信息先验概率的利用有时极大的提高了指令的识别率,比如在识别序号 7 乘法指令时,若 Q_1 状态期识别成功,则 Q_2 、 Q_3 、 Q_4 状态期识别的正确率将为 100%,但值得注意的是,指令序列恢复过程还是出现了少数判断错误的状态,分析发现这些出错的情况大多是非常类似的指令,如果要保证较高的正确率,可采用多次重复实验来剔除错误判断的情况. 因此,借助隐马尔可夫模型来利用指令的先验概率,通过模板匹配的方

法能完成指令序列的恢复.

5 结束语

根据微控制器旁路信号与指令之间的相关性,旁路模板分析技术可用作对微控制器指令进行逆向工程. 此外,把指令模板分析与指令间的先验概率知识相结合,可对指令序列的正确恢复起到积极的作用. 对 AT89C52 微控制器中运行的 DES 加密算法的部分指令序列的实验表明,在指令序列旁路观测的基础上,利用程序“上下文”信息,采用 HMM 建模,用 Viterbi 解码算法求解,能较好的恢复出微控制器中执行的指令序列.

参考文献:

- [1] Chari S, Rao J R, Rohatgi P. Template attacks[C]// Proceeding of the workshop on cryptographic hardware and embedded systems (CHES'03). Cologne, Germany, Springer, 2003:13-28.
- [2] Roman Novak. Side-channel based reverse engineering of secret algorithms[C]//ERK 2003. Germany, 2003. 445-448.
- [3] Martin Goldack. Side-channel based reverse engineering for microcontrollers [D]. Germany, Bochum, Ruhr-University, 2008.
- [4] Dennis Vermoen. Reverse engineering of Java Card applets using power analysis[D]. Netherlands, Master's thesis, Delft University of Technology, 2006.
- [5] 邓高明,吴恒旭,张鹏,等. 旁路模板在密码芯片指令分析中的应用[J]. 微电子学与计算机, 2011,28(2):140-144.
- [6] 吴恒旭,陈开颜,邓高明,等. 基于主成分分析的指令旁路模板分析研究[J]. 微电子学与计算机, 2010, 27(10):156-158.
- [7] 李杰. 隐马尔可夫模型的研究及其在图像识别中的应用[D]. 北京:清华大学, 20046.
- [8] Forney G D. The viterbi algorithm[J]. the IEEE, 1973, 61(3): 268-278.

作者简介:

- 余 浩 男,(1987—),硕士研究生. 研究方向为信息安全.
 陈开颜 女,(1970—),博士,副教授,硕士生导师. 研究方向为信息安全.
 张 阳 男,(1984—),硕士,讲师. 研究方向为信息安全.
 邓高明 男,(1983—),博士. 研究方向为信息安全.
 吴恒旭 男,(1983—),硕士. 研究方向为信息安全.