https://doi.org/10.1051/jnwpu/20244210078

基于门级信息流追踪技术的逻辑混淆攻击方法

刘亮[1], 朱嘉诚[2], 张喆[1], 沈利香[2], 孙玉峰[1], 慕德俊[2]

(1.北京智芯微电子科技有限公司, 北京 100000; 2.西北工业大学深圳研究院, 广东 深圳 518057)

摘 要:逻辑混淆技术是一种实现知识产权保护、防止逆向工程的主流技术手段。提出了基于门级信息流追踪技术的逻辑混淆攻击方法,采用门级抽象层次上的信息流分析方法,建立信息流模型,对输出及其污染标签进行约束,使用 SAT 求解器求解满足约束条件的混淆密钥序列。实验结果表明该攻击方法对 5 种混淆加密算法、2 种面积开销生成的测试基准有很好的破解效果和效率。

关 键 词:逻辑混淆攻击;门级信息流追踪;集成电路;知识产权保护
中图分类号:TP309 **文献标志码**:A **文章编号**:1000-2758(2024)01-0078-06

随着集成电路(integrated circuit,IC)规模和复杂度的快速增长,片上系统通常需要集成来自不同供应商的第三方知识产权块。此外,芯片设计公司与晶圆代工厂往往是分离的。这种商业模式不可避免地将不受信任的各方引入硬件设计和供应链,诸如知识产权(intellectual property,IP)盗用、逆向工程和伪造等硬件攻击行为变得愈加频繁。攻击者很容易就能进行复制 IC、插入硬件木马等攻击行为,从而造成严重的安全隐患。事实上,各种各样的安全威胁已经使得 IC 行业每年损失惨重,对信息技术和集成电路制造业的发展造成了十分严重的阻碍。

面对这一阻碍,研究者提出了各种对策,例如 IP 水印[1]、逻辑混淆[2]和物理不可克隆函数[3]等。在这些方法中,基于密钥的硬件逻辑混淆是一种有效的技术,在学术界和 IC 行业中都引起广泛研究。与源代码混淆或加密相比,它在降低人类可读性的同时不会影响设计黑盒的使用。硬件逻辑混淆通过锁定 IP 或芯片功能主动保护设计,使得硬件设计只有在提供正确混淆密钥的情况下能够正常运行。逻辑混淆能够有效阻止硬件设计的非法生产、使用以及 IP 盗取。

收稿日期:2023-02-20
基金项目:北京智芯微电子科技有限公司实验室开放基金(SGSC0000SJQT2207164)资助
作者简介:刘亮(1982—),研究员
通信作者:朱嘉诚(1996—),博士研究生
e-mail:zhu_jc@mail.nwpu.edu.cn

硬件逻辑混淆主要包括组合逻辑混淆和时序逻辑混淆 2 种技术。组合逻辑混淆[4]通过在门级设计中插入 XOR 门、OR 门或 MUX 实现对硬件电路的混淆,并使用密钥位配置这些逻辑门。错误的密钥会导致电路内部的功能故障。时序逻辑混淆[5]将门级、RTL 级或算法级进行设计以扩展有限状态机状态,只有特定的输入向量序列允许硬件进入正常功能状态空间,否则硬件保持故障状态。逻辑混淆技术发展的同时,也出现了越来越多的针对组合逻辑混淆技术的攻击方法。最具代表性的是基于布尔可满足性(SAT)求解的 SAT 攻击方法[6]。给定一个加密的网表和功能芯片,该芯片能够提供一个正确的输入/输出响应,SAT 攻击通过求解一系列能够有效消除错误密钥的 SAT 问题搜索正确的密钥。但是,SAT 攻击不能直接应用于时序电路。针对时序逻辑混淆电路,基于 SAT 的攻击[7-8]仍可以通过展开时序电路形成更大的组合电路以求解正确密钥。以 SAT 攻击为基础,研究者提出了 AppSAT 攻击[9]、信号概率偏移(signal probability skew,SPS)攻击、基于 AppSAT 的移除(appSAT guided removal,AGR)攻击和基于敏化的 SAT(sensitization guided SAT,SGS)攻击[10]。

门级信息流追踪(gate level information flow tracking,GLIFT)[11]通过对门级网表中的每一个二进制位数据添加 1 bit 的标签,精确地掌握每个二进制位信息的流动,通常用来对信息流进行追踪和分析,验证集成电路设计的安全与可靠性。本文创新

性地探索了门级信息流追踪方法在攻击领域的应用可能性,将 GLIFT 方法引入逻辑混淆攻击之中,结合 SAT 求解器对逻辑混淆电路进行了密钥流分析,破解部分甚至全部密钥。破解部分密钥时可结合 SAT 攻击手段,为 SAT 攻击缩小攻击范围,最终达到破解逻辑混淆密钥这一目的。

本文选取 ISCAS'85 基准电路集中的 8 个电路,使用 5 种逻辑混淆加密算法生成测试电路。通过建立测试电路的 GLIFT 模型,使用 SAT 求解器对 GLIFT 模型进行信息流分析攻击,可以破解逻辑混淆电路的混淆密钥,实验证明该攻击方法具有很好的有效性和可用性。

1　门级信息流追踪

门电路是数字电路的基本组成单元。其中,最基本的门电路即为与门、或门和非门。通过这 3 种基本门电路的不同组合,可以得到与非门、或非门、异或门等更加复杂的门,然后进一步利用复杂门构建更加复杂的电路。GLIFT 方法的本质即为原始电路构建一个相应的 GLIFT 逻辑电路。简而言之,引入 GLIFT 标签就是为原电路增加污染分析节点。不妨假设原电路 C1 中包含输出 Output_0,则其相应 GLIFT 电路中必有一相应的污染标签输出 Output_0_t,该输出就是一个用来表示 Output_0 安全属性的二进制信息。

本文采用大写字母 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{O}$ 等表示信号的逻辑值,其中 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{O} \in \{0,1\}$。$\mathrm{sh}(x)$ 表示相应的逻辑变量 x 或逻辑函数 x 的 GLIFT 逻辑,$\mathrm{sh}(x) \in \{0,1\}$,$\mathrm{sh}(x)=0$ 表示电路信号 x 不受故障影响,$\mathrm{sh}(x)=1$ 表示电路信号 x 受故障影响。

接下来讨论基本逻辑门的 GLIFT 模型建模方法。首先考虑非门的 GLIFT 逻辑。非门的布尔函数为 $\boldsymbol{O}=\overline{\boldsymbol{A}}$,其中 $\boldsymbol{A}$ 为非门的输入,$\boldsymbol{O}$ 为非门的输出。在 GLIFT 逻辑中,如果一个信息单位 $\boldsymbol{A}$ 受到污染,则经过非门的输出亦会受到污染;反之如果一个信息单位 $\boldsymbol{B}$ 不受污染,则经过非门的输出亦不受污染。非门的 GLIFT 逻辑可用(1)式表示。

$$\begin{aligned}\mathrm{sh}(\overline{\boldsymbol{A}}) &= \mathrm{sh}(\boldsymbol{A})\\ \mathrm{sh}(\overline{\boldsymbol{B}}) &= \mathrm{sh}(\boldsymbol{B})\end{aligned} \tag{1}$$

考虑与门的 GLIFT 逻辑,布尔函数为 $\boldsymbol{O}=\boldsymbol{A}\cdot\boldsymbol{B}$,表 1 描述了二输入与门的 GLIFT 逻辑真值表。

表 1　二输入与门的 GLIFT 逻辑真值表

$(\mathrm{sh}(\boldsymbol{A}),\boldsymbol{A})$	$(\mathrm{sh}(\boldsymbol{B}),\boldsymbol{B})$			
	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
(0,1)	(0,0)	(0,1)	(1,0)	(1,1)
(1,0)	(0,0)	(1,0)	(1,0)	(1,0)
(1,1)	(0,0)	(1,1)	(1,0)	(1,1)

由表 1 得二输入与门的 GLIFT 逻辑可用(2)式表示。

$$\mathrm{sh}(\boldsymbol{O}) = \boldsymbol{A}\cdot\mathrm{sh}(\boldsymbol{B}) + \boldsymbol{B}\cdot\mathrm{sh}(\boldsymbol{A}) + \mathrm{sh}(\boldsymbol{A})\cdot\mathrm{sh}(\boldsymbol{B}) \tag{2}$$

根据(2)式,可以得出以下结论:若 $\boldsymbol{A}$ 为 1 且不受污染,则 $\boldsymbol{B}$ 的污染状态对 $\boldsymbol{O}$ 的污染状态起决定性作用;若 $\boldsymbol{B}$ 为 1 且不受污染,则 $\boldsymbol{A}$ 的污染状态对 $\boldsymbol{O}$ 的污染状态起决定性作用;若 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 都受污染,则 $\boldsymbol{O}$ 一定受到污染。

考虑二输入或门的 GLIFT 逻辑。或门的布尔函数为 $\boldsymbol{O}=\boldsymbol{A}+\boldsymbol{B}$,可简化为 $\boldsymbol{O}=\overline{\overline{\boldsymbol{A}}\cdot\overline{\boldsymbol{B}}}$,根据(1) ~ (2)式中非门和与门的 GLIFT 逻辑推演,或门的逻辑可用(3) 式表示。

$$\mathrm{sh}(\boldsymbol{O}) = \overline{\boldsymbol{A}}\cdot\mathrm{sh}(\overline{\boldsymbol{B}}) + \overline{\boldsymbol{B}}\cdot\mathrm{sh}(\boldsymbol{A}) + \mathrm{sh}(\boldsymbol{A})\cdot\mathrm{sh}(\boldsymbol{B}) \tag{3}$$

根据(3)式,可以得出以下结论:若 $\boldsymbol{A}$ 为 0 且不受污染,则 $\boldsymbol{B}$ 的污染状态对 $\boldsymbol{O}$ 的污染状态起决定性作用;若 $\boldsymbol{B}$ 为 0 且不受污染,则 $\boldsymbol{A}$ 的污染状态对 $\boldsymbol{O}$ 的污染状态起决定性作用;若 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 都受污染,则 $\boldsymbol{O}$ 一定受到污染。

类似地,可以得到基本逻辑门如异或门、同或门等基本逻辑单元的 GLIFT 逻辑表达式。根据基本逻辑门的 GLIFT 逻辑,可以为每一个逻辑单元生成 GLIFT 模型,从而为整个电路设计生成 GLIFT 模型。

门级信息流追踪技术广泛应用于硬件设计的安全与可靠性验证。例如,在机密性验证中,定义 $\mathrm{sh}(\boldsymbol{A})=1$ 代表信号 $\boldsymbol{A}$ 具有机密性,若节点 $\boldsymbol{O}$ 为低安全域,则可以通过断言 $\mathrm{sh}(\boldsymbol{O})=0$ 表示机密信息不会流向低安全域 $\boldsymbol{O}$。此时使用 SAT 求解器对断言进行验证,若验证失败,SAT 求解器将返回一个反例,使得 $\mathrm{sh}(\boldsymbol{O})=1$,表示机密信息从信号 $\boldsymbol{A}$ 流向了节点 $\boldsymbol{O}$,违背了安全性原则;反之验证通过,任何情况下信号 $\boldsymbol{A}$ 的信息都不会流向节点 $\boldsymbol{O}$。本文创新性地将门级信息流追踪技术融入安全攻击中,研究了基于门机信息流追踪技术的逻辑混淆攻击方法,可以有

效破解组合逻辑混淆中部分甚至全部密钥。

2 基于门级信息流追踪技术的逻辑混淆攻击方法

2.1 攻击模型

本文提出的攻击方法建立在 Oracle 电路引导攻击之上。本文假设攻击者可以获得混淆电路网表,称之为 Obfuscation Circuit,同时还可以获得与原始电路功能相同的黑盒电路,即 Oracle Circuit。在此攻击模型下,攻击者可以使用不同的输入状态测试混淆电路和黑盒电路,通过对 2 个电路的输出进行比较和观察进行逻辑混淆攻击。

2.2 逻辑混淆攻击方法

本文提出的逻辑混淆攻击方法包含 2 个步骤:①混淆逻辑电路建立 GLIFT 模型,并生成信息流分析电路;②对信息流分析电路进行污染分析,实现混淆密钥破解。

2.2.1 信息流分析电路建模

根据第一节描述的 GLIFT 模型建模方法,将混淆电路网表转换为 GLIFT 电路,本文中将其称为 GLIFT Circuit。然后将 Oracle Circuit 与 Obfuscation Circuit 集成,构建信息流分析电路。

由于 Obfuscation Circuit 和 GLIFT Circuit 都是由 Oracle Circuit 衍生而来,故将 3 个电路中所有的原始电路输入称为 $\boldsymbol{I}$,$\boldsymbol{I}=\{I_1,I_2,\cdots,I_n\}$,输出为 $\boldsymbol{O}$,$\boldsymbol{O}=\{O_1,O_2,\cdots,O_m\}$;相较于 Oracle Circuit,Obfuscation Circuit 和 GLIFT Circuit 增加了混淆密钥输入位,不妨称之为 $\boldsymbol{K}$,$\boldsymbol{K}=\{K_1,K_2,\cdots,K_z\}$,其输出为 $\boldsymbol{O}_G$,$\boldsymbol{O}_G=\{O_{G_1},O_{G_2},\cdots,O_{G_m}\}$;GLIFT Circuit 将污染标签引入电路之中,故为输入或输出后添加后缀“_t”表示对应的污染标签。

信息流分析电路示意图如图 1 所示。

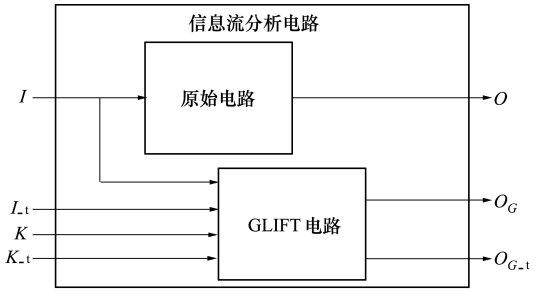

图 1 信息流分析电路示意图

2.2.2 信息流污染分析

首先以一个简单电路为例,对本攻击方法予以说明。以原始电路 $O=A$,混淆电路 $O_G=A\ \&\ K_1|K_2$ 为例,其中正确的混淆密钥应为 $K_1=1,K_2=0$。现考虑破解混淆密钥 K_2 的值,根据 2.2.1 节内容令 $\boldsymbol{A}_{\_t}=0,K_{1\_t}=1,K_{2\_t}=0$,表 2 为 GLIFT 电路的真值表,表内值为$(O_G,O_{G_t})$。

表 2 示例 GLIFT 电路的真值表

$(K_1,K_{1\_t},K_2,K_{2\_t})$	$(\boldsymbol{A},\boldsymbol{A}_{\_t})$	
	(0,0)	(1,0)
(0,1,0,0)	(0,0)	(0,1)③
(1,1,0,0)	(0,0)	(1,1)
(0,1,1,0)	(1,0)①	(1,0)
(1,1,1,0)	(1,0)②	(1,0)

首先寻找原始电路与 GLIFT 电路输出不同的组合,表 2 中上标①~③标明了 3 组输出不同的组合。由于原始电路的输出必然是正确的,所以这 3 组输出 O_G 一定是错误输出,并且是因为 K_1 或 K_2 错误导致的。然后寻找 $O_{G\_t}=0$ 的组合,第一步 3 个组合中①、②符合条件。根据 GLIFT 逻辑理论,输出的污染标签为 0 代表被污染的输入没有对输出产生影响,表明①、②的输入 K_1 未对 O_G 产生影响,结合第一步的结论,得出①、②产生错误输出的原因是 K_2 的错误。组合①、②对应的 $K_2=1$,则正确的 K_2 应取反得到 $K_2=0$,与正确密钥一致。

拓展至普遍情况,本文提出的攻击方法流程如图 2 所示。

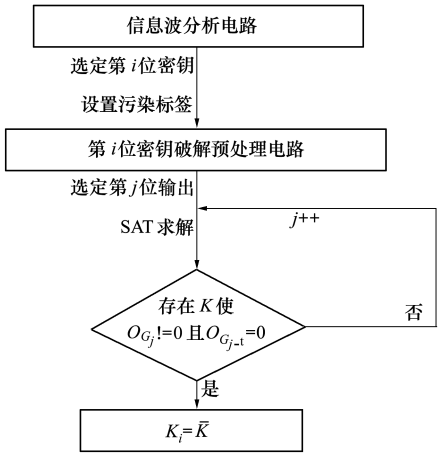

图 2 逻辑混淆攻击流程图

首先选取需要破解的密钥位 i,设置该密钥位标签 K_{i_t} 为零,其他所有密钥位的标签为 1,将所有的原始电路输入污染标签设置为零。然后选取一个输出位 j,调用 SAT 求解器进行判断,求解是否存在特定的输入序列使 $\boldsymbol{O}_{G_j}$ 与 $\boldsymbol{O}_j$ 相异,由此找出所有使输出结果产生错误的密钥序列。最后调用 SAT 求解器进行判断,求解是否存在密钥序列在满足上一步约束的同时使得 $\boldsymbol{O}_{G\_t}$ 为零,由此可以确定除密钥位 K_i 以外的其他密钥位都不会对输出产生污染,确保输出的错误是 K_i 错误导致的。若能够通过 SAT 求解器找到满足条件的密钥序列,则只需将密钥序列中的 K_i 取反即可得到正确的密钥值;否则另取输出位 $j+1$,按上述的 2 个步骤进行迭代,直至遍历所有输出位;若仍无法破解 K_i,则说明 K_i 破解失败,应挑选 K_{i+1} 尝试下一位密钥破解。

3　实验与分析

3.1　实验环境与方案

硬件环境为英特尔 i7 处理器,8 GB 内存。软件环境为 64 位 Ubuntu 14.04操作系统,Yosys,ABC,Lingeling SAT 求解器和 CPLEX ILP 12.5 求解器。

实验选取 ISCAS′85 以及北卡罗莱纳州微电子中心的 8 个基准电路,采取 IOLTS14、ToC13_mux、ToC13_xor、RND 和 DAC12 逻辑混淆加密方法,分别选用 5%和 10%的逻辑加密面积开销,生成 2 组共 80 个逻辑混淆加密电路。

3.2　实验结果分析

实验对生成的 80 个加密电路按照本文所描述的攻击方法进行密钥破解。下文仅展示并讨论 IOLTS14算法实验数据,表 3 和表 4 分别展示了对 IOLTS14 加密算法在 5%和 10%面积开销下生成混淆电路的攻击数据。表中 input 代表加密电路的输入数目;keys 代表加密电路所含的混淆密钥位数;gates 代表加密电路所含逻辑门的数量;crack 代表本文提出的攻击方法破解的混淆密钥位数;time 代表密钥破解所消耗的时间。

由表 3 可以看出在 IOLTS14 加密算法 5%面积开销下,本文提出的攻击方法能够对 c432、c499、c880、c1355、apex2 和 i4 电路中的所有密钥位进行破解,同时能够破解 c1908 电路 46 位密钥中的 37 位,但是不能破解 ex5 电路中的任何密钥。总体上,本文提出的攻击方法对 IOLTS14 算法 5%面积开销下生成的混淆电路有较高的破解成功率。归其原因,一方面 5%面积开销下生成的加密电路复杂度较低,另一方面,IOLTS14 混淆加密算法安全性相对较低。

表 3　IOLTS14 算法 5%面积开销下混淆电路攻击数据

基准电路	input	keys	gates	crack	time/s
c432	36	10	170	10	43
c499	41	12	214	12	146
c880	60	22	405	22	2 346
c1355	41	29	575	29	3 922
c1908	33	46	926	37	5 047
apex2	39	32	642	32	247
i4	192	27	365	27	473
ex5	8	53	1 108	0	9 208

由表 4 可以看出在 IOLTS14 加密算法 10%面积开销下,本文提出的攻击方法能够对 c499、apex2 和 i4 电路中的所有密钥位进行破解,能够破解 c1355、c1908 电路中大部分密钥,但是不能破解 c432、c880 和 ex5 电路中的任何密钥。可以看出在 IOLTS14 算法 10%面积开销下生成的混淆电路有更高的安全性,但本文提出的攻击方法仍能对大多数加密算法实现有效破解。而在 2 种面积开销下,攻击方法都不能对 ex5 电路实现任一位密钥的破解,主要原因是 ex5 基准电路规模较大,对基于该电路进行加密的逻辑混淆电路进行信息流分析所需的存储空间和运行时间也会相对较大,电路密钥位和输出之间很可能有着较为复杂的信息流关系。因此,使用本文所提出的新型攻击方法对该逻辑混淆电路进行攻击所需的运行时间和存储空间也会相对较大。

表 4　IOLTS14 算法 10%面积开销下混淆电路攻击数据

基准电路	input	keys	gates	crack	time/s
c432	36	20	180	0	304
c499	41	24	226	24	1784
c880	60	44	427	0	1 403
c1355	41	59	605	32	8 425
c1908	33	91	971	70	5 646
apex2	39	65	675	65	299
i4	192	53	391	53	420
ex5	8	106	1 161	0	16 546

对 80 个测试电路的攻击结果进行分析,本文提

出的新型攻击算法对电路规模中等、混淆算法简单的测试基准具有较高的破解成功率,能够达到 90% 以上;而对于大规模电路、混淆算法精密的测试基准,密钥搜索过程中容易导致状态空间爆炸问题,破解成功率受限于实验设备的存储空间。综合考虑下,本文提出的逻辑混淆攻击方法能够对实验环节中在多种逻辑加密面积开销下使用多类型加密方法生成的测试电路成功进行密钥破解。该方法可以适用于多种应用场景,能够在 2 种(5% 和 10%)常用逻辑加密面积开销下成功对 5 种主流加密方法进行攻击,这就意味着该方法有较大概率对现实生活中的一般信息系统或一些敏感信息系统中的逻辑混淆硬件成功进行密钥破解。此外,该方法还可以在多种复杂电路结构上进行有效应用,通过在不同具体情况下有机结合信息流分析攻击和 SAT 联合攻击,该攻击方法在基于 ISCAS′85 电路集中多个复杂基准电路的逻辑混淆攻击测试中表现良好。

4 结 论

本文提出了基于门级信息流追踪技术的逻辑混淆攻击方法对逻辑混淆电路进行密钥破解。以 80 个混淆电路为测试基准,该攻击方法为混淆电路生成 GLIFT 模型,为每个模型添加约束,利用 SAT 求解器求解满足所有约束的密钥序列,从而破解逻辑混淆密钥。对 80 个混淆电路的攻击数据进行统计与评估,该方法能够对大多数测试基准实现密钥破解。

参考文献:

[1] ABDEL-HAMID A T, TAHAR S, ABOULHAMID E M. IP watermarking techniques: survey and comparison[C]//The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, 2003: 60-65

[2] YASIN M, SINANOGLU O. Evolution of logic locking[C]//2017 IFIP/IEEE International Conference on Very Large Scale Integration, 2017: 1-6

[3] HERDER C, YU M D, KOUSHANFAR F, et al. Physical unclonable functions and applications: a tutorial[J]. Proceedings of the IEEE, 2014, 102(8): 1126-1141

[4] MENON V V, KOLHE G, SCHMIDT A, et al. System-level framework for logic obfuscation with quantified metrics for evaluation[C]//2019 IEEE Cybersecurity Development, 2019: 89-100

[5] CHAKRABORTY R S, BHUNIA S. HARPOON: an obfuscation-based SoC design methodology for hardware protection[J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2009, 28(10): 1493-1502

[6] SUBRAMANYAN P, RAY S, MALIK S. Evaluating the security of logic encryption algorithms[C]//2015 IEEE International Symposium on Hardware Oriented Security and Trust, 2015: 137-143

[7] SHAMSI K, LI M, PAN D Z, et al. KC2: key-condition crunching for fast sequential circuit deobfuscation[C]//2019 Design, Automation & Test in Europe Conference & Exhibition, 2019: 534-539

[8] EL MASSAD M, GARG S, TRIPUNITARA M. Reverse engineering camouflaged sequential circuits without scan access[C]//2017 IEEE/ACM International Conference on Computer-Aided Design, 2017: 33-40

[9] SHAMSI K, LI M, MEADE T, et al. AppSAT: approximately deobfuscating integrated circuits[C]//2017 IEEE International Symposium on Hardware Oriented Security and Trust, 2017: 95-100

[10] YASIN M, MAZUMDAR B, SINANOGLU O, et al. Removal attacks on logic locking and camouflaging techniques[J]. IEEE Trans on Emerging Topics in Computing, 2017, 8(2): 517-532

[11] TIWARI M, WASSEL H M, MAZLOOM B, et al. Complete information flow tracking from the gates up[C]//Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, 2009

A logic obfuscation attack method based on gate-level information flow tracing techniques

LIU Liang[1], ZHU Jiacheng[2], ZHANG Zhe[1], SHEN Lixiang[2], SUN Yufeng[1], MU Dejun[2]

(1.Beijing Smart-Chip Microelectronics Technology Co.,Ltd., Beijing 100000, China;
2.Shenzhen Research Institute of Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Logical obfuscation technique is a mainstream technical means to achieve intellectual property protection and prevent reverse engineering. In this paper, we propose a logical obfuscation attack method based on the gate-level information flow tracking technique, using the information flow analysis method at the gate-level abstraction level, establishing an information flow model, constraining the output and its contamination labels, and solving the obfuscation key sequence satisfying the constraints using an SAT solver. The experimental results show that the attack method has good cracking effect and efficiency for benchmarks generated by five obfuscated encryption algorithms with two test area costs.

Keywords: logic obfuscation attack; gate level information flow tracking; integrated circuits; intellectual property protection

引用格式: *刘亮, 朱嘉诚, 张喆, 等. 基于门级信息流追踪技术的逻辑混淆攻击方法[J]. 西北工业大学学报,2024,42(1): 78-83*

LIU Liang, ZHU Jiacheng, ZHANG Zhe, et al. A logic obfuscation attack method based on gate-level information flow tracing techniques[J]. Journal of Northwestern Polytechnical University, 2024, 42(1): 78-83 (in Chinese)