https://doi.org/10.1051/jnwpu/20224010076

基于信息流分析的密码核设计安全验证与漏洞检测

马艺新, 唐时博, 谭静, 李雪霏, 胡伟

(西北工业大学 网络空间安全学院, 陕西 西安 710072)

摘 要:密码算法核是保障信息机密性和完整性的关键部件。由于密码算法实现的安全性与算法在数学上的安全性是 2 个完全不同的问题,密码算法核可能隐含设计缺陷和旁路信道等安全隐患。基于功能验证的安全性分析方法严重依赖于测试向量的质量,覆盖率低,难以满足密码算法核这一安全关键部件的安全验证需求。从信息流安全的角度研究密码算法核安全验证与漏洞检测方法。该方法能够对密码算法核中全部逻辑信息流进行精确度量,实现对机密性和完整性等安全属性的形式化验证,可通过捕捉有害信息流动来检测密码算法核设计中潜在的安全隐患。实验结果表明信息流安全验证方法对密码算法核中的设计缺陷和旁路信道导致的敏感信息泄漏有很好的检测效果。

关 键 词:信息流分析;安全验证;漏洞检测;密码算法核

中图分类号:TP309 **文献标志码**:A **文章编号**:1000-2758(2022)01-0076-08

密码技术是网络与信息安全的重要支撑性技术,是用于保障信息机密性和完整性等关键属性的重要手段。密码算法在数学上的安全性与其实现的安全性是 2 个截然不同的问题。虽然密码算法的安全性已经从数学上得到证明,但是,密码算法核实现会由于设计错误、旁路信道[1]、调试接口[2]或后门程序[3]等原因引入安全漏洞。这些安全漏洞一旦被攻击者挖掘与利用,将能够在较低时间和空间复杂度条件下完全恢复算法保密密钥。攻击者可进一步利用密钥窃取更多有价值的信息,使得密码算法核以及计算设备的安全性受到严重威胁。

在密码核硬件电路设计过程中,验证的时间开销往往达到总工作量的 40%~70%[4]。功能验证的局限性在于:覆盖率依赖于测试激励信号,导致验证覆盖率低、验证不充分;设计规模很大或要模拟的情况很多时通常会导致模拟时间会很长,验证效率低。密码算法核中的高隐蔽性安全漏洞,单纯依靠有限次的功能测试很难保证可以覆盖。由此可见,功能验证难以满足密码算法核的安全验证需求。

信息流跟踪(information flow tracking,IFT)方法通过为存储对象添加安全标签,并通过计算与跟踪存储对象的标签执行信息流安全策略去验证、控制整个硬件系统的信息流动[5]。信息流跟踪方法具有良好的信息流监控能力,可以防止有害的信息流动。Tiwari 等[6]提出了门级信息流跟踪(gate level information flow tracking,GLIFT)方法。GLIFT 方法通过为每个二进制位数据分配一比特的标签精确地监控每个二进制位信息的流动,是一种细粒度的信息流控制方法,能够通过捕捉有害的信息流动来检测和消除设计中潜在的安全漏洞。Ferraiuolo 等[7]设计了一种带类型系统的安全 Verilog 语言 SecVerilog,通过在设计编译阶段静态检查不同安全级别数据之间的信息流安全属性,给出了一种验证处理器架构中安全域和非安全域之间隔离特性的方法。Jin 等提出了一种基于携带证明代码的硬件安全验证框架,硬件知识产权(intellectual property,IP)核携带了安全属性相关的定理,可利用 Coq 进行安全定理证明,从而检测第三方 IP 核中潜在的木马设计[8-9]。Ardeshiricham 等[10]提出了寄存器传输级信息流跟踪(register transfer level information flow

收稿日期:2021-05-25 **基金项目**:国家自然科学基金(62074131)与陕西省自然科学基金(2021JQ-123)资助

作者简介:马艺新(1997—),女,西北工业大学硕士研究生,主要从事硬件信息流安全验证研究。

通信作者:胡伟(1982—),西北工业大学副教授、博士生导师,主要从事硬件安全、可重构计算以及嵌入式系统等研究。

e-mail:weihu@ nwpu.edu.cn

tracking,RTLIFT)方法,它能够直接在寄存器传输级(register transfer level,RTL)代码的基础上生成精确的 IFT 逻辑,进行安全验证。

现有的安全验证方法需要设计者用新的语言重新编写硬件模型,或者需要设计新的验证框架,或者由于抽象层次低不能扩展到大的设计。本文从信息流安全的角度研究密码算法核安全验证与漏洞检测方法,基于信息流分析的安全验证方法只需要掌握密码算法核的 RTL 级源码或门级网表,在 RTL 级源码或者门级网表的基础上描述信息流和标签的传播,不需要设计者学习新的硬件安全语言。本文利用可实现形式化验证功能的综合工具 Yosys,它支持自定义信息流安全属性,在定义不同信息流关系时更加灵活,例如,显式流还是隐式流,并且在较高的抽象层次上工作会提高验证速度。

本文通过对 Aoki Laboratory[11] 发布的密码算法核中全部逻辑信息流进行精确度量,对机密性、完整性等安全属性进行形式化验证,可捕捉到电路中的有害信息流,对密码算法核中的安全漏洞以及时间侧信道有很好的检测效果。

1　背景

1.1　硬件信息流分析

信息流是指信息的流动和传播,在本文中信息流均代表与硬件设计系统逻辑状态密切相关的逻辑信息流。硬件信息流分析中,数据通常被赋予一个标签,该标签表征数据的安全属性,如可信/不可信或保密/非保密。在数据运算过程中标签随之在电路中传播,硬件信息流分析中标签传播策略是根据当前操作类型、输入以及输入标签来确定输出的标签,通过静态分析或动态检查输出的标签,可以有效防止违反信息流安全策略的运算操作,防止有害信息流的流动,例如:保密信息流向了电路中非保密区域数据寄存器,可信区域程序计数器中接收到了来自以太网的不可信数据。

如表 1 所示,以密码算法核为例详细介绍标签传播规则。密码算法核中密钥与输入进行与操作,密钥与输入均为一比特,密钥为 K,输入为 M,输出为 $O = K \& M$,k_t,m_t 和 o_t 为密钥、输入和输出的标签。$K,M \in \{0,1\}$,$k_t,m_t,o_t \in \{\text{LOW},\text{HIGH}\}$,其中标签“LOW”代表密码算法核中低安全级别属性非保密或者可信,标签“HIGH”代表密码算法核中高安全级别安全属性保密或者不可信。

假定 $L(a)$ 为信号 a 的信息流跟踪逻辑,当 $a_t = 1$ 时,$L(a) = \text{HIGH}$,当 $a_t = 0$ 时,$L(a) = \text{LOW}$。在运算操作中对输出起关键决策作用的输入的标签决定了输出的标签,例如输入 A 的标签为 HIGH,$A = 0$,输入 B 的标签为 LOW,$B = 1$,A 为 0 决定了输出为 0,所以输出标签由 A 安全级别决定,输出标签为 HIGH。由表 1 可以推导出密码算法核中二输入与门的信息流跟踪逻辑为(1)式。

$$o_t = Mk_t + Km_t + k_t m_t \tag{1}$$

表 1　密码算法核中与操作的标签传播规则

(k_t,K)	(m_t,M)			
	(LOW,0)	(LOW,1)	(HIGH,0)	(HIGH,1)
(LOW,0)	(LOW,0)	(LOW,0)	(LOW,0)	(LOW,0)
(LOW,1)	(LOW,0)	(LOW,1)	(HIGH,0)	(HIGH,1)
(HIGH,0)	(LOW,0)	(HIGH,0)	(HIGH,0)	(HIGH,0)
(HIGH,1)	(LOW,0)	(HIGH,1)	(HIGH,0)	(HIGH,1)

类似地,可以得到密码算法核中其他基本逻辑门如非门、或门和异或门等基本逻辑单元的信息流模型,从而可以构建信息流模型库。根据信息流模型库可以为密码算法核的每一个逻辑单元离散式地生成信息流模型,从而为整个密码算法核设计生成信息流模型。

1.2　信息流安全验证

安全验证是一种在流片前对硬件设计进行的基于属性的逻辑验证,安全验证能够证明一些被定义好的安全属性在硬件设计中是否一直成立。根据具体方法的不同,进行安全验证的硬件设计可能是 RTL 级代码或者逻辑门级网表。

基于硬件信息流分析的安全验证是检测密码算法核安全漏洞的一种手段。假设密码算法核中包含安全漏洞,密码算法核正常运行产生正确的运算结果,并不会观察到安全漏洞,只有在罕见的情况下,例如违反电路机密性(如密钥泄露至输出)时,安全漏洞才会被观察到。安全漏洞难以通过有限次的功能验证被覆盖,信息流分析方法则可取得更好的检测效果。

在安全属性的定义中,机密性属性主要是保证高安全级别敏感信息不会流向密码算法核中可公开观测的区域。标准硬件描述语言 Verilog 和 VHDL 只能描述功能属性,不能用于描述安全属性。信息

流分析中关注信息的流动,更好地描述了机密性、完整性等安全属性,安全属性可采用断言语言描述。

图 1 描述了信息流安全验证方法。密码算法核可以作为安全验证的检测对象,该方法分析了密码算法核中需满足的通用安全属性,如机密性、完整性、隔离特性以及一些模式相关的属性等。将安全属性映射至密码算法核的安全验证模型上,主要进行属性相关信号安全级别的划分和安全属性标签的实例化,以及安全属性中断言语句向功能性验证语言的转化。完成安全属性的映射之后,即可结合密码算法核的信息流模型,采用形式化验证工具实现安全属性的验证,若验证失败则表明存在安全漏洞,同时会生成一个反例,可用于漏洞复现。

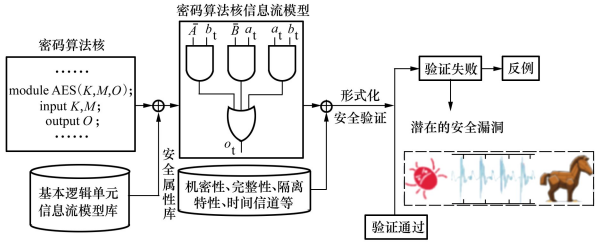

图 1 密码算法核安全验证方法

2 密码核安全验证与漏洞检测方法

2.1 密码算法核信息流模型的生成

在 1.1 节为密码算法核中基本逻辑门构造信息流模型的基础上,可以构造一个功能完备的信息流模型库,通过为密码算法核中更复杂的功能单元生成信息流模型,最终生成整个密码算法核的信息流模型。

如图 2 所示,生成密码算法核的信息流模型首先需要构建一个包含基本逻辑单元如与门、或门、非门等的信息模型库。给定一个逻辑函数,首先使用逻辑综合工具将其综合为由基本逻辑单元描述的门级网表,例如图 2 中将二输入选择器逻辑函数转化为由 2 个与门和 1 个或门描述的门级网表;然后将门级网表映射至基本逻辑单元信息流模型库中,为门级网表中的各个逻辑单元实例化信息流模型,生成了逻辑函数的信息流模型。图 2 中生成了二输入选择器的信息流模型;当使用信息流模型生成算法为逻辑函数生成信息流模型后,将逻辑函数的信息流模型集成至基本逻辑单元信息流模型库中,将基本信息流模型库扩充至复杂信息流模型库;对密码算法核而言,可使用同样的方法,经过逻辑综合生成门级网表,将门级网表中各个逻辑单元映射至复杂硬件信息流模型库,为密码算法核中各个逻辑单元实例化地附加信息流模型,由此生成整个密码算法核的信息流模型。

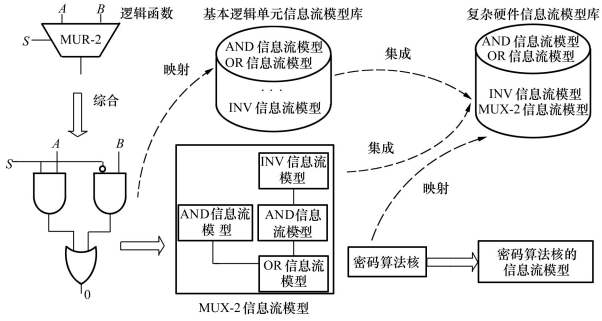

图 2 密码算法核信息流模型的生成

2.2 安全属性

对密码算法核进行安全验证的前提是分析电路所需遵循的安全属性,如机密性、完整性、隔离特性以及时间信道等。首先需要描述所需验证的安全属性,通常借鉴 SystemVerilog Assertion 等断言描述语言的语义,并拓展安全相关的特性。安全属性描述语言常用的关键字,主要包含安全级别的设置、安全属性的断言、断言条件的设置以及许可路径的设定。关键字 set 用来设置信号安全级别,如保密、公开、可信、不可信等,关键字 assert 用来断言信号安全级别。

在安全属性描述语言的基础上,可以对密码算法核中通常需要满足的安全属性进行描述。完整性属性保证高安全级别的不可信信号不会流向低安全级别的可信区域;机密性属性保证高安全级别的保密信号不会流向可公开观测的区域。在机密性验证中将保密信号设置为高安全级别 HIGH,并断言其不会流向低安全级别 LOW 的输出。例如,断言密钥不应该流向中间数据寄存器,描述如下:

```
set    key = HIGH
assert    data_reg == LOW
```

将安全属性描述语言转化为形式化验证工具 Yosys 可以识别的安全约束,即进行安全标签的设置,以及属性相关信号安全级别的划分与设置,具体如下所示:

```
assign    key_t =    1'b1
assert    data_reg_t ==  1'b0
```

2.3　安全验证与漏洞检测

Yosys 是种基于 Verilog 的开源综合与验证工具,该工具在命令行窗口下启动和运行。它可以将 Verilog 代码综合为逻辑门级网表,内置用于检查安全属性和功能的布尔可满足性(boolean satisfiability problem,SAT)求解器。

图 3 描述了密码算法核安全验证与漏洞检测方案。首先将密码算法核通过 Yosys 综合生成逻辑门级网表,然后根据 2.1 节的信息流模型生成方法,对门级网表进行分析,对门级网表中的逻辑单元进行处理,包括信号端口定义、基本逻辑门、寄存器、触发器以及赋值语句等,为门级网表中的每个逻辑单元实例化信息流模型,最终生成整个密码算法核的信息流模型。

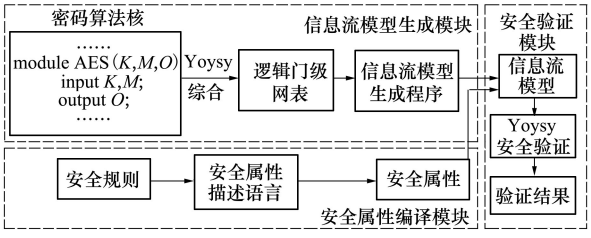

图 3　密码算法核安全验证与漏洞检测方案

密码算法核信息流模型是安全验证的基础。通过对密码算法核进行安全需求分析,建立密码算法核安全属性库,将密码算法核需满足的安全规范使用安全属性描述语言进行描述,然后对密码算法核中属性相关信号进行安全等级的划分,其次通过信号安全标签的设置,指定允许或禁止的信息流,通常采用断言语句进行信息流的禁止或许可。这些安全属性断言被写入密码算法核信息流模型中,用于验证信息流安全流动。例如,机密性验证中需要将保密信号设置为高安全级别 HIGH,并禁止它流向低安全级别 LOW 的输出。假设密钥不应该被泄露,将密钥安全属性标记为 HIGH,其他信号安全属性标记为 LOW,因为这是一个关于机密性的属性,需要检查密文有效信号是否始终为 LOW,这种安全属性的验证描述如下:

```
set    key_t    HIGH
set    DEAFAULT_LABEL    LOW
assert    cipher_ready_t    LOW
```

将安全属性转化为 Yosys 可识别的安全约束脚本或文件,结合写入安全属性断言的信息流模型,调用 Yosys 验证工具进行安全验证,分析密码算法核安全验证模型中安全属性以及敏感信息的流动,观察待验证的安全属性标签信号值。根据 Yosys 验证的结果判断是否存在违反信息流安全规范的信息流动,若验证成功则不存在安全漏洞,若验证失败则通过分析验证失败时相关信号的值,对安全漏洞特征进行分析以及定位,结合验证工具给出的反例对信息流模型进行功能测试,可获得密码算法核中安全漏洞的泄露途径以及漏洞行为。

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,16 GB 内存。软件环境为 64 位 Ubuntu 20.04.1 操作系统,编译器为 gcc 9.3.0。实验如图 4 所示,包含 3 个部分:设计输入、安全验证和验证输出分析。实验验证的对象是加密算法 IP 核,来自 Aoki Laboratory[11]。

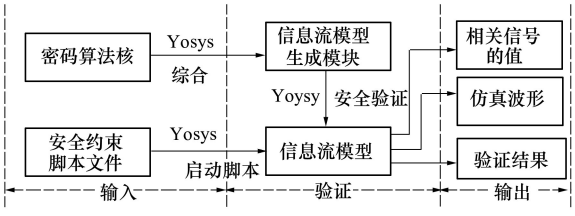

图 4　实验方案

3.2　AES 密码算法核安全验证

1) 对下载的密码算法核使用 Yosys 综合工具进行综合,具体是使用综合脚本和 script 命令,例如对于 AES 密码算法核,对其进行综合,产生门级网表。

2) 对综合后产生的逻辑网表进行处理,对逻辑网表中的每一个逻辑单元进行分析,包括端口定义、基本逻辑门、寄存器、触发器以及赋值语句等,实例化每一个逻辑单元的信息流模型,最后生成密码算法核的信息流模型。

3) 对密码算法核的信息流模型进行分析,根据密码算法核中的信号与安全标签的传播以及电路安全规范对需要验证的安全属性进行分析,对属性相关信号进行安全级别划分,以及安全标签的设置,实现信息流规则描述。例如在 AES_Comp IP 核中对密钥信号进行分析,根据机密性安全属性中禁止高安全级别的保密信号流向可公开观测的区域,使用

安全属性描述语言进行描述,并断言密钥信号不会直接流向可公开观测的密文输出,具体描述如下:

```
set key_t HIGH
set DEFAULT_LABEL LOW
assert Dout_t LOW
```

将安全属性编译为 Yosys 可识别的安全约束,即安全验证脚本。在安全验证中,不对明文、密文、密钥和其他状态信号的安全级别进行细致的划分,将除密文外所有信号的安全标签置为 0,重点关注密钥泄露问题。为了在输出结果中更直观地体现出密钥的流动,得到更直观的验证结果,将部分密钥位的安全标签设置为 1,部分位的污染标签设置为 0。例如将密钥污染标签最低位设置为 1,剩余位设置为 0,即 Kin_t = 128 'h01。为了捕捉密钥泄露安全漏洞,需要设置反例,向信息流模型中加入安全断言,设置一条赋值语句,当输出信号安全标签等于密钥安全标签时,即输出信号为高安全级别时,将证明信号 proof-f 的安全标签赋值为 1,具体如下:

```
assign proof_t = (Dout_t == 128 'h01) ? 1 : 0;
```

Yosys 中集成了 SAT 求解器,在安全脚本中添加 SAT 形式化验证语句,从而形式化地验证证明信号的值是否始终为 0,是则验证成功,否则验证失败。可分析安全漏洞的存在,使用 SAT 形式化验证进行安全验证的脚本为 sat -prove proof_t 1'b0。得到的验证结果如图 5 所示,可以看出寄存器 AES_Comp_ENC.Drg_t 的值等于 Kin_t 的值。

```
Signal Name                    Dec       Hex
                Bin
-------------------- ---------- --------- ----------------------------
--------------------------------------------------------------------------------
----------
  \AES_Comp_ENC.Drg_t              --        --    00000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000001
  \Kin_t                           --        --    00000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000001
```

图 5 AES_Comp 密码算法核形式化验证结果

根据加密算法扩散和混淆的特点,密钥的任何一位都应该对密文的多位存在影响,尽管只有最低位密钥安全标签为 1,加密结果中至少有一半位的安全标签为 1。通过 Mentor Graphics ModelSim 工具进行仿真,得到加密结果中仅有最低位安全标签为 1,功能仿真中未捕捉到密钥和加密结果的值相等的波形。

图 6 显示了 AES 密码算法核加密模块的硬件结构,如被标记为虚线所示,执行第一轮密钥加时,明文与密钥异或的结果直接被赋值给密文输出端口。如图 7 所示,将仿真 testbench 中明文设置为全 0,则在输出端口上可直接观测到密钥值。因为只有最低位密钥受到污染,所以密文输出中也只有最低位受到污染。经过信息流安全验证与功能验证结合分析得到,在特定明文输入条件下即明文为全 0 时密码核直接泄露了全部的密钥,此种安全漏洞属于由中间加密结果导致的关键信息泄露。

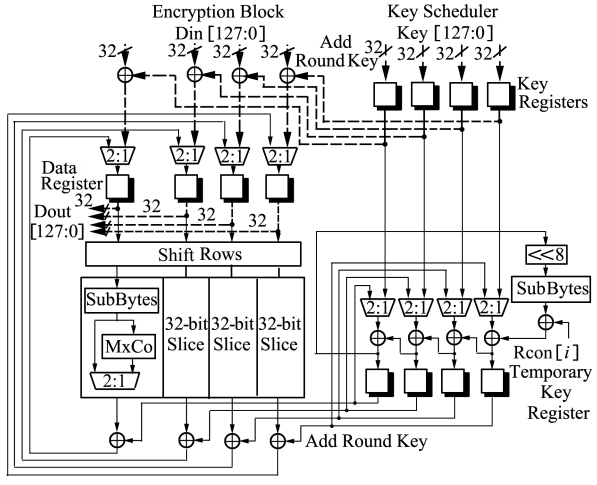

图 6 AES_Comp 密码算法核加密模块的硬件结构[11]

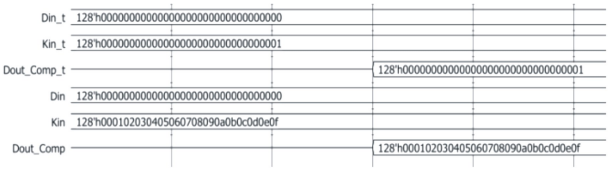

图 7 AES 密码算法核密钥泄露仿真波形

3.3 RSA 密码算法核安全验证

本实验采用 Aoki Laboratory[11] 提供的 RSA 密码算法核进行安全验证和漏洞检测,在安全验证过程中,不对明文、模数、密钥和其他状态信号的安全级别进行细致的划分,将密钥的安全标签全部置为 1,其他信号的标签置为 0。按照 3.2 节中的方法对 RSA 密码算法核进行逻辑综合,并产生信息流模型。

对生成的信息流模型进行安全验证,使用 Mentor Graphics ModelSim 工具进行时序仿真,通过分析仿真产生波形的变化来验证相关信号安全标签是否会产生变化,检验是否存在安全漏洞。设置密钥信号 32'hFFFFFFFF,经过一段时间的仿真后,Dout_t 信号变为 32'hFFFFFFFF,BSY_t 信号也变为

1。Dout 输出的标签信号受到污染是毋庸置疑的,因为密钥必然会影响输出。BSY 信号是操作完成的标志,当 IP 核进行加密、解密或者导入数据之后,BSY 信号变为 1。通过对 RSA 密码算法核进行分析,该密码算法利用密钥位进行算法流程控制。密钥位取 1 或者取 0 时会导致条件分支语句执行,需要执行不同的操作,会导致 RSA 模幂运算时间出现差异[12],操作时间上的差异导致 RSA 密码算法核中存在时间侧信道。

信息流安全仿真的结果表明:受污染的密钥不仅导致密文输出受到污染,还导致了操作完成信号(BSY)受到了污染,这表明了密钥也流向了 BSY 信号。密钥流向密文信号是毋容置疑的,因为密钥影响着加密结果。在本次实验中,密钥对 BSY 信号是没有直接影响的,当加密完成后,BSY 信号总会由 0 变为 1。事实上,由 BSY 信号的标签 BSY_t 从 0 变为 1 可以看到密钥确实影响了 BSY 信号。这是由于密钥值的变化导致操作时间存在差异,从而 BSY 信号由 0 置 1 的时间也发生了变化,这属于时间信息流。

通过信息流安全验证的方法与工具,可以捕捉到由密钥信号到加密完成信号的时间隐通道。

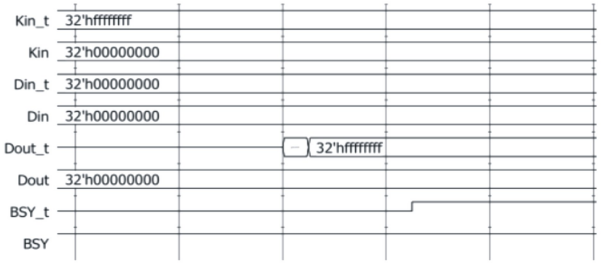

图 8　RSA 密码算法核时间隐通道仿真结果

3.4　IP 核安全验证总结

对 Aoki Laboratory[11] 中的 6 个密码算法核进行安全验证,总结如表 2 所示。经过信息流安全验证和信息流仿真,DES、AES、Camellia、MISTY1 和 CAST-128 密码核经过安全验证以后均发现有明文或者密钥通过中间寄存器泄露至输出的漏洞行为。这是由于上述几种密码算法核的设计实现方式导致的,中间寄存器的存在使得关键信息存在泄露风险。RSA 密码核中发现了由密钥到操作完成信号的时间侧信道。对密码算法核进行形式化安全验证和功能仿真,形式化验证在 Ubuntu 平台下 Yosys 中完成;功能仿真在 Windows 系统下 Mentor Graphics ModelSim 中完成。对验证时间和资源消耗进行统计,结果列于表 3。可以看出,DES 和 MISTY1 密码算法核在功能仿真中没有捕捉到密钥或明文泄露。

表 2　密码核安全验证与漏洞检测结果

IP 核	安全属性	漏洞行为
RSA	set Kin_t : =HIGH assert BSY_t==LOW	时间侧信道 导致密钥泄露
DES	set Kin_t : =HIGH assert Dout_t==LOW	密钥泄露至输出
AES	set Kin_t : =HIGH set Din_t : =HIGH assert Dout_t==LOW	明文和密钥 泄露至输出
Camellia	set Kin_t : =HIGH assert Dout_t==LOW	密钥泄露至输出
MISTY1	set Din_t : =HIGH assert Dout_t==LOW	明文泄露至输出
CAST-128	set Din_t : =HIGH assert Dout_t==LOW	明文泄露至输出

表 3　密码核验证性能

IP 核	泄露对象	形式化安全验证		功能仿真	
		验证时间/s	资源消耗/MB	时钟周期/ns	资源消耗/MB
RSA	密钥			4 979 450	247.3
DES	密钥	4	197.25		
AES	明文	25	847.4	265	24.2
	密钥	22	847.5	30	112.5
Camellia	密钥	31	1 325.85	95	8
MISTY1	明文	48	1 578.16		
CAST-128	明文	61	2 202.95	1 305	17.5

4 结 论

由于传统功能验证方法已经无法满足密码算法核安全验证的需求,本文提出了利用信息流安全验证的方法进行密码算法核验证。以 Aoki Laboratory 中的密码算法核作为测试基准,信息流安全验证方法通过为每个信号添加安全标签的方式为整个密码算法核设计产生安全验证模型,利用形式化验证工具 Yosys 进行输出信号安全级别的检测,从而验证是否存在违反信息流安全属性的有害信息流流动。辅助使用仿真工具 Mentor Graphics ModelSim,进行信息流仿真,对敏感信息的泄露行为有了直观展现,并且通过相关信号安全标签的变化分析出 RSA 密码算法核中时间侧信道的存在。无论是对中间寄存器导致的敏感信息泄露还是时间信息流,信息流安全验证方法都有较好的检测效果。需要进一步完善的是如何自动化书写高质量安全属性。

参考文献:

[1] MAO B, HU W, ALTHOFF A, et al. Quantitative analysis of timing channel security in cryptographic hardware design[J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2017, 37(9): 1719-1732

[2] ZHANG F, LOU X, ZHAO X, et al. Persistent fault analysis on block ciphers[J]. IACR Trans on Cryptographic Hardware and Embedded Systems, 2018, 20(3): 150-172

[3] SKOROBOGATOV S, WOODS C. Breakthrough silicon scanning discovers backdoor in military chip[C]//Proceedings of the 14th International Conference on Cryptographic Hardware and Embedded Systems, 2012

[4] GAO Y, LIU L, DU H, et al. Software and hardware co-verification technology based on virtual prototyping of RF SOC[C]// 2018 IEEE International Conference on Computer and Communication Engineering Technology, 2018

[5] TAI Y, HU W, ZHANG L, et al. A multi-flow information flow tracking approach for proving quantitative hardware security properties[J]. Tsinghua Science and Technology, 2020, 26(1): 62-71

[6] 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

[7] FERRAIUOLO A, XU R, ZHANG D, et al. Verification of a practical hardware security architecture through static information flow analysis[C]//Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems, 2017

[8] JIN Y, GUO X, DUTTA R G, et al. Data secrecy protection through information flow tracking in proof-carrying hardware IP-part I: framework fundamentals[J]. IEEE Trans on Information Forensics and Security, 2017, 12(10): 2416-2429

[9] BIDMESHKI M M, GUO X, DUTTA R G, et al. Data secrecy protection through information flow tracking in proof-carrying hardware IP-part Ⅱ: framework automation[J]. IEEE Trans on Information Forensics and Security, 2017, 12(10): 2430-2443

[10] ARDESHIRICHAM A, HU W, MARXEN J, et al. Register transfer level information flow tracking for provably secure hardware design[C]//Design, Automation & Test in Europe Conference & Exhibition, 2017: 1691-1696

[11] Aoki Laboratory. Cryptographic hardware project: IP cores[EB/OL]. (2008-04-01)[2021-03-01]. http://www.aoki.ecei.tohoku.ac.jp/crypto/web/cores.html

[12] 毛保磊, 胡伟, 慕德俊, 等. 基于信息熵的 RSA 硬件时间隐通道信息泄露量化研究[J]. 计算机学报, 2018, 41(2): 426-438

MAO Baolei, HU Wei, MU Dejun, et al. Information entropy-based quantification of information leakage in RSA hardware time-hidden channels[J]. Journal of Computer Science, 2018, 41(2): 426-438 (in Chinese)

Cryptographic core design security verification and vulnerability detection based on information flow analysis

MA Yixin, TANG Shibo, TAN Jing, LI Xuefei, HU Wei

(School of Cyberspace Security, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Cryptographic cores are the key components to enforce information security properties related to confidentiality and integrity. Since the security of a cryptographic core implementation and the security of the cryptographic algorithm itself from the mathematical perspective are two different problems, cryptographic cores may contain hidden security vulnerabilities such as design flaws and side channels. Security analysis methods based on functional verification rely heavily on the quality of test vectors and usually have low test coverage, which is difficult to meet the security verification requirements of security-critical components like cryptographic cores. This work proposes a cryptographic core security verification and vulnerability detection method from the perspective of information flow security. The proposed method can accurately measure all logical information flows in cryptographic core designs to formally verify security properties such as confidentiality and integrity. It can detect potential security vulnerabilities in cryptographic implementations by capturing harmful information flows. Experimental results show that our information flow security verification method is effective in detecting sensitive information leakage caused by the design vulnerabilities and side channels in cryptographic cores.

Keywords: information flow analysis; security verification; vulnerability detection; cryptographic core

引用格式:马艺新, 唐时博, 谭静, 等. 基于信息流分析的密码核设计安全验证与漏洞检测[J]. 西北工业大学学报,2022,40(1): 76-83

MA Yixin, TANG Shibo, TAN Jing, et al. Cryptographic core design security verification and vulnerability detection based on information flow analysis[J]. Journal of Northwestern Polytechnical University, 2022, 40(1): 76-83 (in Chinese)