# 面向硬件安全验证的信息流覆盖度量

Andres Meza 加
州大学圣迭戈分校
anmeza@ucsd.edu

Ryan Kastner 加
州大学圣迭戈分校
kastner@ucsd.edu

arXiv:2304.08263v1 [cs.CR] 12 Apr 2023

摘要—**安全图用于对计算机网络和系统中的攻击、防御措施、缓解措施及漏洞进行建模。通过添加适当的属性，它们可以利用标准图算法提供安全度量。超流图是一种寄存器传输级（RTL）硬件安全图，可用于促进安全验证。超流图对信息流进行建模，并附加属性，以便安全度量能够测量流路径、流条件和流速率。超流图有助于理解与机密性、完整性和可用性相关的硬件漏洞，如在多种威胁模型下对 OpenTitan 硬件信任根的分析所示。**

## I. 引言

安全图在对计算机和网络系统中的攻击与防御措施进行建模方面已有悠久历史 [1]–[4]。安全图可以根据规范和漏洞自动生成 [5]。将与安全相关的属性添加到节点和边后，便可利用安全度量来理解和推理攻击的严重程度以及漏洞的有效性。可以使用高效的图算法分析该图，例如最短路径、广度优先搜索和深度优先搜索、割集以及最小/最大流。

Hyperflow 图将安全图的核心思想应用于 RTL 硬件安全。Hyperflow 图使用与硬件信息流跟踪（信息流跟踪）相关的属性，对机密性、完整性和可用性方面的安全漏洞进行建模。Hyperflow 图通过图算法实现硬件安全度量指标。Hyperflow 图使用开源硬件编译器 [6] 自动构建，并可通过开源工具 [7]进行可视化。

Hyperflow 分析评估硬件中的信息流，这对于理解安全行为至关重要。尽管现有的编译器依赖分析能够提供有关数据依赖和控制依赖的信息，但它无法高效地表达并简洁地概述信息流关系。Hyperflow 分析扩展了传统分析，将非干扰和其他超属性纳入考量。其关键创新在于扩展分析，使其能够考虑硬件信息流跟踪所提供的安全标签。这样，分析便从轨迹属性扩展到了超属性 [8]。

超流图使用基于仿真的属性进行注释，以从功能测试平台中提取与安全相关的信息。因此，超流图包含与功能仿真相关的信息，以及来自信息流跟踪安全标签的数据。安全标签值概括了与设计中资产相关的非干扰特性，这对于验证涉及机密性、完整性和完整性的设计属性至关重要。仿真数据提供了关键属性，例如语句的执行次数，以及语句在特定安全标签下的执行次数。

超流图能够自然地实现信息流的可视化，帮助安全验证工程师更直观地理解设计的安全态势。属性驱动的硬件安全验证 [9] 会标记设计资产，旨在了解与该资产相关的信息可能流向何处，以及在何种条件下流动。超流图通过总结与信息流相关的条件及其他相关信息，对信息流进行可视化。这为设计的硬件安全提供了独特的可视化洞察，并帮助验证工程师更好地凭直觉评估弱点、理解漏洞并推导安全属性。本工作的贡献包括以下内容：

- 介绍超流图——一种用于寄存器传输级安全分析与可视化的模型。
- 定义超流图上的属性和度量指标，从而深入了解硬件安全漏洞。
- 展示超流图在 Open-Titan 硬件信任根中的价值。

## II. 背景

### A. 基于图的安全建模与度量

图是一种重要的抽象工具，可用于建模和理解安全攻击、防御措施、漏洞及缓解措施。安全图使我们能够衡量系统的漏洞，并可基于成熟高效的图算法推导缓解措施的有效性。因此，许多安全研究工作都采用图来建模攻击和防御措施。

特权图是最早运用图来理解系统安全的研究成果之一 [1]。它对系统访问控制进行建模，其中节点表示用户（组）的权限，边表示一个用户将其权限扩展至另一个用户（组）的能力。特权图有助于理解访问控制策略、通过路径搜索定义漏洞，并利用属性和图搜索算法评估攻击成功的概率。

攻击树将如何攻击某项资产形式化表示 [2], [3]。树根是攻击目标，叶节点则定义了实现该目标的方法。边带有逻辑与／或标签，用于关联子节点与其父节点。节点具有布尔值或连续值属性。可以在树上定义度量指标，以确定最佳攻击计划、比较攻击成本、了解潜在漏洞，并决定在哪里采取缓解措施。攻击防御树通过加入对策并描述其与攻击之间的相互作用，更全面地对系统安全进行建模 [4]。

攻击图广泛用于分析网络安全问题，其中节点表示网络配置的逻辑陈述，边表示网络配置与攻击者权限 [5], [10], [11]之间的因果关系。边上标注了与攻击成功概率、攻击执行时间或成本相关的属性。最短路径算法可识别成功概率最高的攻击。

攻击图相关文献总体上聚焦于网络安全问题。不过，近期有一些研究将这些思路应用于与硬件安全联系更为紧密的问题。

Koteshwara 将基于图的度量指标扩展到 PCB 级服务器架构 [12]。组件和接口分别作为节点和边，

分别为节点分配基础分数——根据通用漏洞评分系统（CVSS）计算的风险严重性指标，分数越高表示越容易受到攻击。基础分数在一定程度上是凭经验设定的，例如，SPI、LPC、处理器、以太网套接字、VGA、USB、PCIe、UART 和 I2C 组件都容易受到攻击（0.95），因为 TrustHub 中列出了针对它们的硬件木马。信任根的脆弱性要低得多（0.05），因为它采用了缓解措施和保护机制。而闪存则不易受到攻击（0.0），因为信任根会保护它。Saha 等人 [13] 初步尝试将这一方法扩展到片上系统架构。他们使用预定义影响概率，对两个小型（$< 30$ 个顶点）SoC 级图进行了分析。

Hyperflow 图将安全图分析转化为 RTL 硬件。Hyperflow 图使用硬件信息流跟踪对与机密性、完整性和可用性相关的重要信息进行建模。这使得我们能够采用简洁的度量指标，帮助理解潜在弱点并确定漏洞。

### *B.* 硬件信息流跟踪

硬件信息流跟踪（信息流跟踪）使验证工程师能够跟踪资产（例如寄存器、内存位置和信号）中存储的信息，以及这些信息在硬件中的传播情况。硬件信息流跟踪能够以极少的工作量自动执行安全分析。借助属性驱动设计流程，验证工程师可以识别资产，并描述其信息可以或不可以传播到何处 [9]。硬件信息流跟踪有助于验证与机密性、完整性、可用性、侧信道及其他威胁模型相关的硬件漏洞 [14]。

硬件信息流跟踪使用安全标签自动跟踪资产。每个 RTL 变量都关联一个安全标签，用于标示相关信息流的存在。显式流由直接的数据移动导致，即 RTL 赋值。隐式流取决于上下文，即从条件语句中的变量流向赋值语句中变量的信息流，这种流可能由该条件语句导致。

Hyperflow 图能够独特地对 RTL 设计中的信息流进行建模，从而简洁地概述潜在的安全漏洞。Hyperflow 图根据 IFT 增强仿真添加属性标注。对超流图应用图算法，可以揭示弱点，并提供与不同漏洞相关的度量指标。我们将在下一节对 Hyperflow 图进行形式化定义，介绍如何从 RTL 硬件设计中自动生成超流图，并定义其标注过程。

## III. HYPERFLOW 图

超流图是一种有助于安全分析和可视化的中间表示（IR）。Hyperflow 图描绘了硬件设计变量之间的关系，包括直接流、条件流、与安全验证相关的信息（资产、边界）、功能仿真相关的信息（执行的赋值语句），以及硬件安全分析所需的其他重要信息。

### *A.* 形式化定义

超流图 $G$ 是一个有向图 $G = (V,\ E)$，其中：

- $V$ 是一个带有属性标注的顶点集合；
- $E$ 是一个由带有属性的有向边组成的集合。

每个顶点 $v \in V$ 都是一个有序对（$\mathrm{s},\ \mathrm{M}^v$），其中 s 是待分析 RTL设计中的单个设计信号，而 $\mathrm{M}^v$ 是一个集合，包含与信号 s 相关联的仿真元数据。

$$\begin{aligned} V &= \{\ v_1,\ v_2,\ \ldots,\ v_{n-1},\ v_n\ \} \\ &= \{(\mathtt{s}_1, \mathtt{M}_1^v),\ (\mathtt{s}_2, \mathtt{M}_2^v),\ \ldots,\ (\mathtt{s}_{n-1}, \mathtt{M}_{n-1}^v),\ (\mathtt{s}_n, \mathtt{M}_n^v)\} \end{aligned}$$

顶点-元数据集合 $\mathrm{M}^v$ 中的每个元素都是形如（t、val、$\mathrm{val}^\tau$）的有序三元组，其中

- t 是仿真中的特定时刻
- val 是信号 s 在时刻 t 的功能值；• $\mathrm{val}^\tau$ 是信号 s 在时刻 t 的污点值（即安全标签值）。

每条边 $e \in E$ 都是一个有序三元组 ($\mathrm{s}^t$，$\mathrm{s}^h$，$\mathrm{M}^e$)，其中 $\mathrm{s}^t$ 和 $\mathrm{s}^h$ 分别表示边 $e$的尾部和头部信号。$\mathrm{M}^e$ 是一个集合，包含与从信号 $\mathrm{s}^t$ 到信号 $\mathrm{s}^h$的信息流相关的仿真元数据。超流图可以对隐式、显式、控制、数据和时序信息流进行建模。

$$\begin{aligned} E &= \{\ e_1,\ \ldots,\ e_m\ \} \\ &= \{(\mathtt{s}_1^t, \mathtt{s}_1^h, \mathtt{M}_1^e),\ \ldots,\ (\mathtt{s}_m^t, \mathtt{s}_m^h, \mathtt{M}_m^e)\} \end{aligned}$$

边元数据集合 $\mathrm{M}^e$ 中的每个元素都是形如 ($\mathrm{t}^s$，$\mathrm{t}^e$) 的有序对，其中

- $\mathrm{t}^s$ 是一个仿真时刻，此时发生该边所建模的信息流所需的所有条件均已满足
- $\mathrm{t}^e$ 是一个仿真时间（晚于 $\mathrm{t}^s$），在该时刻，该边所建模的信息流发生所需的一个或多个条件不再满足。

### *B.* 超流边与流的特征

超流图的边表示潜在的信息流。信息流源自 RTL设计中的赋值语句。更准确地说，一条边（$v_X$，$v_Y$）对应 RTL代码中的一条赋值语句，其中信号 Y 位于赋值语句的左侧，而信号 X 是（1）赋值语句右侧的一个操作数，（2）赋值语句守卫条件中的一个操作数，或（3）同时满足（1）和（2）。[1] 情况（1）是从 X 到 Y 的显式流，情况（2）则是隐式流。

信息流还可以根据其条件性质进行分类。条件流具有必须满足才能发生的谓词（或守卫条件）。相反，无条件流没有谓词，会持续发生。条件流的谓词是累积的谓词，即包含流起源构造周围的所有嵌套守卫条件。

### *C.* 超流图自动生成

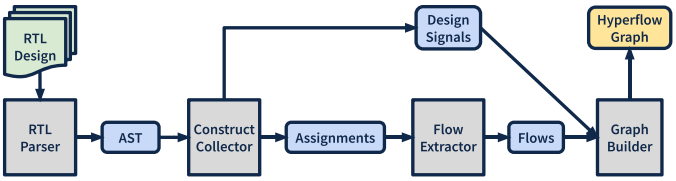



图 1：从硬件设计的源 RTL 代码构建超流图的自动化流程。

超流图通过图 1 所示的自动化流程构建。输入是 RTL 代码（例如 Verilog 文件），输出是该硬件设计的超流图。

图构建流程分为四个主要步骤。第一步解析输入的 RTL 代码，生成抽象语法树（AST）。我们使用 SystemVerilog 作为硬件描述语言，并使用 slang [6] 作为解析器。一般而言，这些方法也适用于其他硬件描述语言和解析器。第二步遍历 AST，以识别对应于信号或信号间信息流的 AST 节点/子树。信号是

[1]当信号 X 是赋值语句右侧的操作数，且出现在该赋值语句的守卫条件中时，每条流都表示为一条唯一的边。

从 SystemVerilog 声明结构中提取，包括 reg、wire 和 logic。同样，也从 SystemVerilog 赋值结构中提取流，包括连续_assign、阻塞_assignment 和非阻塞_赋值语句；还从 SystemVerilog 约束/条件结构中提取流，包括 if、else 或 case。第三步从上一步收集的结构中提取信息流。如第 III-B 节所述，流是从赋值语句及其守卫条件中提取的。最后一步将提取出的设计信号和信息流分别转换为超流图中的顶点和边，该图的定义见第 III 节。

### *D.* 超流图属性与标注过程

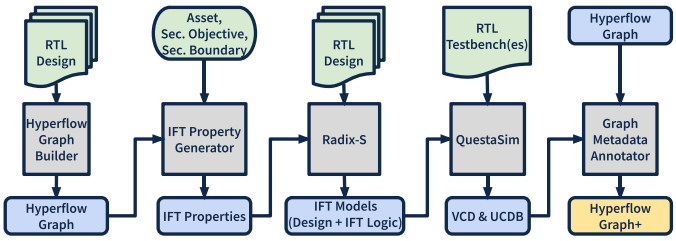



图2：通过自动化流程为硬件设计的超流图添加仿真元数据，包括但不限于设计信号的功能值和 IFT安全标签。

超流图可以添加各种形式的元数据，以便获得仅凭未添加注释的超流图无法得出的见解。本文通过图2所示的自动化流程为超流图添加仿真元数据。该自动化流程的输入包括：(1) 特定硬件设计的 RTL代码；(2) 设计资产列表，以及这些资产的安全目标和安全边界；(3) 至少一个用于该硬件设计的 RTL测试平台。标注过程的输出是一个超流图，其顶点添加了相应设计信号的功能值和污点值注释，边则添加了相应流的激活时间注释。图标注过程可分为 5 个主要步骤。

标注过程的第一步会创建一个未标注的超流图（见第 III-C 节）。仅当当前硬件设计尚不存在超流图时，才需要执行此步骤。

此过程的第二步会利用超流图以及设计资产列表（连同各资产的安全目标和安全边界），以生成一组 IFT 属性。这些 IFT 属性将用于生成 Radix 安全监视器（即由 Radix-S 生成的 IFT 模型），并在后续步骤中与原始设计一同进行仿真。对于给定的三元组（资产、目标、边界），IFT 属性的形式如下所示。

```
断言 iflow(
  资产  =/=>{安全边界之外的信号}
);

断言 iflow(
  {安全边界之外的信号}  =/=>资产
);
```

第一个 IFT 属性通过规定资产中的信息不应泄漏到其安全边界之外的信号集合，来验证资产的机密性。第二个属性通过规定来自资产安全边界之外的信号的信息不应泄漏到资产中，来验证资产的完整性。

第三步使用 Cycuity 的 Radix-S 工具，自动生成 Radix 安全监视器（即 IFT 模型），以便在仿真期间生成/跟踪安全标签。Radix-S 会为每个 IFT 属性构建一个安全监视器。每个安全监视器都会跟踪从所有源信号（即无流算子左侧的信号）到所有其他设计信号的信息泄漏。

第四步使用商业仿真器（即 Questa Advanced Si mulator），通过提供的测试平台对原始设计和所有安全监视器进行仿真。仿真期间，原始设计和安全监视器的轨迹会被保存，以供后续使用。原始设计的轨迹包含所有设计信号的功能值，而安全监视器的轨迹则包含所有设计信号的污点值。

流程的最后一步会解析存储的轨迹（即 VCD 文件），并将仿真元数据映射到相应的顶点和边上。

## IV. 信息流度量指标

本节介绍一组可从超流图中推导出的定量信息流覆盖度量。对于每种覆盖度量，我们都会说明其推导方法及其所提供的见解。需要指出的是，为简洁清晰起见，本节未列出其他可从超流图中推导出的有用度量指标。

### *A.* 度量指标 *#1*：信号连通性

信号连通性度量（SCM）是一种布尔度量，用于指示在设计正常执行期间，信息是否可能从某个信号 A 流向另一个信号 B。如果超流图中不存在从信号 A 到信号 B 的任何路径，那么信息就不可能从信号 A 流向信号 B。然而，如果至少存在一条从信号 A 到信号 B 的路径，则需要进一步分析以确定该路径是否可实现。如果从信号 A 到信号 B 的路径仅使用无条件流/边，那么信息就可能经由这条路径从 A 泄漏到 B。但是，如果该路径使用了任何条件边，就必须进行可达性分析，以确定激活从 A 到 B 的每条边/流所需的设计状态序列，能否根据设计其余部分的逻辑/FS M 实际实现。进行这种形式化可达性分析的另一种方法是对设计进行 IFT 仿真，并将信号 A 设为 IFT 源（即其信息被标记安全标签的信号）。如果在仿真过程中 B 的安全标签在任何时刻从 0 变为 1，那么信息就可能发生流动。

### *B.* 指标 *#2*：路径激活

路径激活指标（PAM）用于量化在特定仿真期间，从信号 A 到信号 B 的路径中有多少条被激活。当信息按照路径的顶点和边所描述的方式从信号 A 流向信号 B 时，该路径即被视为已激活。

$$\mathrm{PAM}(G, A, B) = \frac{\#\text{ of Activated Paths}}{\text{Total \# of Paths from } A \text{ to } B}$$

从带注释的超流图推导 PAM 分为两个主要步骤。第一步是收集信号 A 中的信息到达信号 B 的时间点。

这通过检查存储在信号 B 对应的顶点元数据集中的安全标签来完成。第二步是针对上一步收集的每个时间点，确定是哪条路径使信号 A 中的信息到达信号 B。这通过反转超流图中的边，然后从信号 B 到信号 A 执行约束路径搜索来完成。搜索过程中可以经过哪些边受到约束。具体而言，只有当某条边所代表的流正是使信号 A 中的信息在特定时间 $t$到达当前顶点/信号的流时，才能经过该边。还可以通过找出在 $t$ 满足约束，且其源顶点在时间 $t$ 也具有相应非零安全标签的边来确定这一点。

### C. 指标 #3：信号接近度

信号接近度指标（SPM）表示来自某个信号 A 的信息距离到达某个信号 B 还有多远。“接近”或距离的概念以超流图中的边数来衡量。对于给定的仿真时间 t，SPM 将返回信号 A 的信息与信号 B 之间相隔的最少边数。

$$\text{SPM}(G,\ A,\ B) = \text{信号 A 的信息与 } B \text{ 之间的最少边数}$$

第一步检查信号 B 的安全标签元数据，以确定来自信号 A 的信息是否已经到达信号 B。如果信息已经到达信号 B，则 SPM 为 0。否则，继续执行。第二步收集在时间 t 包含来自信号 A 信息的顶点／信号集合。可以通过从信号 A 开始执行广度优先搜索，直到到达信号 B 来完成此操作。执行广度优先搜索时，会检查每个顶点／信号的安全标签元数据，以查看其中是否包含来自信号 A 的信息。最后一步计算所有候选信号与信号 B 之间的最短路径，然后返回该路径的长度（即边数）。

### D. 指标 #4：局部信息流速率

局部信息流速率（LIFR）指标用于量化在给定时间窗口内，从信号 A 流向信号 B 的信息量。信息流量以比特为单位进行测量。

$$\text{LIFR}(G, B, t_1, t_2) = \frac{\text{SLBT}(B, t_1, t_2)}{t_2 - t_1}$$

SLBT($B,\ t_1,\ t_2$) 项返回给定时间窗口内，B 的安全标签中任意比特从 0 转换为 1 的次数。

### E. 指标 #5：全局信息流速率

全局信息流速率（GIFR）指标用于确定在给定时间窗口内，从信号 A 流经所有路径到达信号 B 的信息量。

## V. 实验

本节将简要介绍 OpenTitan 硬件信任根，然后说明如何通过带注释的超流图，推导其一次性可编程（OTP）存储器控制器的信息流度量指标。

### A. OpenTitan

OpenTitan 是一款商用级开源硬件信任根 [15]。在半导体行业中，硬件信任根用于执行各种安全关键任务，例如安全启动、运行模式（如正常模式与调试模式）的管理和配置，以及敏感数据（如密码密钥）的管理。OpenTitan 旨在支持广泛的此类硬件安全应用场景，具备经过安全增强的 RV32IMCB RISC-V Ibex 核心、各种安全外设（如高级加密标准、KMAC、HMAC）、多种存储器（如只读存储器、嵌入式闪存、静态随机存取存储器、一次性可编程存储器），并配备专用控制器以实现访问控制和扰码功能，同时还提供各种输入输出外设。在本实验中，我们展示了超流图及其相关度量指标在一次性可编程存储器控制器上的实用价值。

OTP 控制器是一种非易失性存储器，用于存储密钥和其他安全资产。因此，其安全验证对于 OpenTitan 的正确、安全运行至关重要。OTP 控制器中包含 RndCnstKey 资产——一种网表常量，用作 OTP 存储器中安全资产的密钥加密密钥。存储在 OTP 存储器中的闪存和静态随机存取存储器（SRAM）扰码密钥、设备根秘密、解锁令牌及其他资产，均使用 RndCnstKey 加密，因此这些资产不会以明文形式存储在 OTP 中。因此，必须确保 RndCnstKey 的机密性；一旦攻击者访问 RndCnstKey，就能解密 OTP 存储器中的安全关键数据。来自 RndCnstKey 的任何信息都应仅留在 OTP 控制器内部，不得有关于 RndCnstKey 的信息泄漏到控制器之外。为验证这一点，我们对 OpenTitan 进行 IFT 插桩仿真，追踪从 RndCnstKey 流向 OpenTitan 中任意信号的信息。

### B. 超流图

图 3 展示了 OTP 控制器超流图的多种可视化形式。超流图总共有 11,579 个顶点（即寄存器）和 32,084 条边（即潜在的信息流）。图 3a 展示了超流图中的所有显式边和隐式边。图 3b 仅展示显式边，而图 3c 仅展示隐式边。最后，图 3d 展示了一条路径，该路径使信息能够从 RndCnstKey 流出控制器。超流图顶点和边所附带的元数据/属性使得创建此类可视化成为可能。该超流图用于推导表 I 中报告的度量指标。

### C. IFT 覆盖率指标

在对 OpenTitan 进行 IFT 插桩仿真后，我们按照第 III-D 节所述，使用安全标签和其他仿真元数据标注 OTP 控制器的超流图。随后，我们为 OTP 控制器的所有输出推导出所提出的四项安全度量。表 I 报告了推导得到的度量指标。

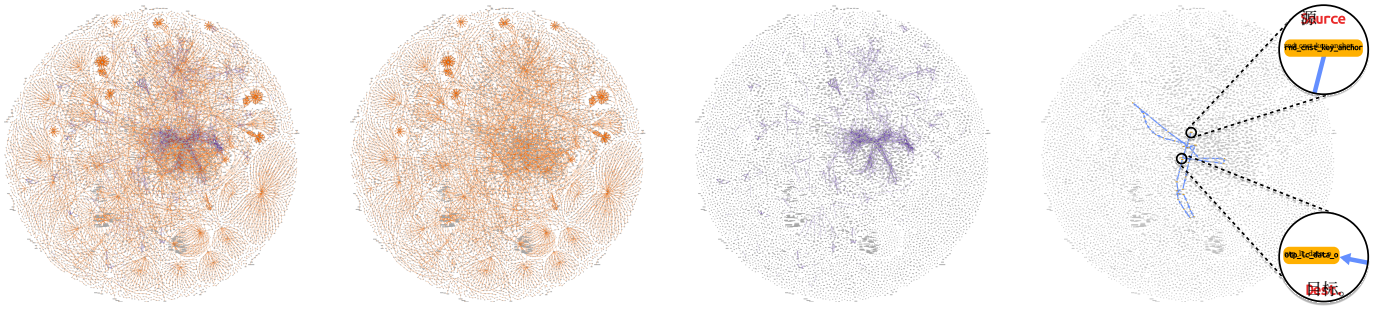
## VI. 相关工作

### A. 硬件安全度量

功耗侧信道采用了一些硬件安全领域中最有效、应用最广泛的度量指标。它们使用统计度量指标 [16]–[19], , 与该威胁模型有直接对应关系。遗憾的是，这些度量指标无法有效对应我们所考虑的、更通用的硬件安全威胁模型，这些模型涉及机密性、完整性和可用性。

Rostami 等人总结了针对硬件木马、知识产权盗用、逆向工程、侧信道和仿冒的度量指标 [20], , 这些指标所对应的威胁模型与本文提出的不同。DSeRC 有两个与故障注入和硬件木马相关的度量指标 [21]。

各种硬件信息流跟踪验证工具会针对属性 [22]–[24]是否被违反给出二元判定（二值逻辑度量）。这有助于检测潜在漏洞是否存在，但对于理解漏洞的价值较低。




**a** **b** **c** **d**

图 3：a) 部分展示了 OTP 控制器的超流图，其中包含 11,579 个顶点（即寄存器）和 32,084 条边（即潜在的信息流）。在这 32,084 条边中，27,218 条为显式流（见 b) 部分），4,866 条为隐式流（见 c) 部分）。d) 部分展示了在一次信息流跟踪仿真期间，信息从 OTP 扰码器资产 RndCnstKey 泄漏到 OTP 控制器输出端（otp_lc_data_o.test_unlock_token）的路径。这些超流图的交互式版本可在 hyperflowgraphs.com 上查看，便于更清晰地观察和探索。

表 I：本表报告了 OpenTitan 一次性可编程存储器控制器各输出端的信息流覆盖度量，考察的信息源为 RndCnstKey。“SCM”表示从 RndCnstKey 到所有列出输出端的信号连通性度量。“PAM*”表示从 RndCnstKey 到所有列出输出端的路径激活指标。“SPM”表示 RndCnstKey 的信息与所有输出端之间的信号接近度指标。“LIFR (x%)”表示 IFT 插桩仿真分别完成 25%、50%、75% 和 100% 后所有输出端的局部信息流速率。仿真由 OpenTitan 的“chip sw otp ctrl smoketest”测试平台驱动。

| No. | OTP 存储器控制器的输出端口 | SCM | PAM* | SPM | LIFR（25%） | LIFR（50%） | LIFR（75%） | LIFR（100%） |
|---|---|---|---|---|---|---|---|---|
| 1 | /ot top/u 一次性可编程 控制/cio test en o | 假 | 0 | ∞ | 0 | 0 | 0 | 0 |
| 2 | /ot top/u 一次性可编程 控制/cio 测试 o | 假 | 0 | ∞ | 0 | 0 | 0 | 0 |
| 3 | /ot top/u 一次性可编程 控制/intr 一次性可编程 错误 o | TRUE | 0 | 26 | 0 | 0 | 0 | 0 |
| 4 | /ot top/u 一次性可编程 控制/intr 一次性可编程 操作完成 o | TRUE | 0 | 26 | 0 | 0 | 0 | 0 |
| 5 | /ot top/u 一次性可编程 控制/lc 一次性可编程 供应商测试 o | 假 | 0 | ∞ | 0 | 0 | 0 | 0 |
| 6 | /ot top/u otp 控制/otp AST 电源时序 o | 假 | 0 | ∞ | 0 | 0 | 0 | 0 |
| 7 | /ot top/u otp 控制/otp 观测 o | 假 | 0 | ∞ | 0 | 0 | 0 | 0 |
| 8 | /ot top/u otp 控制/原语传输层 o | 假 | 0 | ∞ | 0 | 0 | 0 | 0 |
| 9 | /ot top/u 一次性可编程 控制/熵分发网络 o | TRUE | 1 | 0 | 0 | 1 | 1 | 1 |
| 10 | /ot top/u 一次性可编程 控制/lc 一次性可编程存储器编程 o | TRUE | 1 | 0 | 0 | 2 | 2 | 2 |
| 11 | /ot top/u 一次性可编程 控制/电源 一次性可编程 o | TRUE | 1 | 0 | 0 | 2 | 2 | 2 |
| 12 | /ot top/u 一次性可编程 控制/警报 发送 o | TRUE | 1 | 0 | 0 | 4 | 4 | 6 |
| 13 | /ot top/u 一次性可编程 控制/核心 TL o | TRUE | 1 | 0 | 0 | 0 | 0 | 135 |
| 14 | /ot top/u 一次性可编程 控制/OTBN 一次性可编程 密钥 o | TRUE | 1 | 0 | 0 | 128 | 128 | 193 |
| 15 | /ot top/u otp 控制/flash otp key o | TRUE | 1 | 0 | 0 | 128 | 128 | 259 |
| 16 | /ot 顶层/u 一次性可编程 控制/一次性可编程 密钥管理器 密钥 o | TRUE | 1 | 0 | 0 | 513 | 513 | 513 |
| 17 | /ot top/u otp 控制/otp 硬件配置 o | TRUE | 1 | 0 | 0 | 644 | 644 | 644 |
| 18 | /ot 上层/ u 一次性可编程 控制/静态随机存取存储器 一次性可编程 密钥 o | TRUE | 1 | 0 | 0 | 384 | 384 | 771 |
| 19 | /ot top/u 一次性可编程 控制/一次性可编程 l c data o | TRUE | 1 | 0 | 0 | 1102 | 1102 | 1102 |

Contreras 等人针对测试结构开展了漏洞分析，并进行了与密码核心、硬件木马和访问控制相关的实验 [25]。他们的分析在门级进行路径遍历，并计算距离——即从资产到控制点或观测点之间的门数。他们认为路径越长，安全性越高。Hyperflow 图则在 RTL 层级运行，能够提供与控制结构相关的额外信息，并利用非干扰来更准确地理解信息流。

信任覆盖率是用于量化第三方 IP [26] 信任水平的度量指标。它包含三种与功能、结构和资产相关的基于向量的覆盖率指标。资产覆盖率是其中最相关的指标，可基于超流图计算。资产覆盖率衡量指定资产能够到达的可观测输出数量。他们使用这些度量指标评估硬件木马。

定量信息流用于表征信息泄漏量级 [27]。目前已有研究尝试将其应用于硬件 [28]–[30]，并聚焦于密码核心和硬件木马。然而，要将定量信息流跟踪应用于这些威胁模型之外的场景仍颇具挑战。

SCRIPT 是一个利用基于信息流跟踪的漏洞度量指标来评估侧信道泄漏的框架 [31]。SCRIPT 使用一种基于门级路径敏化的形式化方法，来识别信息可能流经的寄存器。形式化方法难以扩展；相关技术不得不采用启发式方法，例如避免全时序自动测试向量生成，并称其“效率低下且耗时” [25]。此外，这些方法针对的是功耗侧信道。通过采用基于仿真的信息流跟踪分析，我们的技术在适用范围和可扩展性方面涵盖了这项工作。我们的技术还可以应用于其基于信息流跟踪的分析，从而更高效地找出目标寄存器。

HW2VEC 是一种自动化工具，可将硬件设计转换为欧几里得图嵌入，随后将其用作图神经网络的特征输入 [32]。他们展示了如何利用这种图变换训练二值神经网络，以判断设计是否含有硬件木马。另一个示例则开发了一种神经网络，用于检测一个设计是否盗用了另一个设计。它们解决的问题与我们提出的问题不同。

据我们所知，目前还没有研究采用基于仿真的信息流跟踪度量指标。我们这项技术的优势包括：1)

采用非干扰，而不只是直接影响性；2) 了解测试平台效应对安全性的影响，反之亦然；3) 提供针对隐式流和显式流的流速度量。

## VII. 结论

本文介绍了超流图这一寄存器传输级安全分析与可视化模型，并定义了超流图上的属性和度量指标，以帮助理解硬件安全漏洞。我们在 OpenTitan 硬件信任根上展示了超流图的价值：运用所提出的超流图和度量指标，可视化并理解信息从 OpenTitan 的 OTP 存储器控制器中的安全关键资产流出的过程。

## 参考文献


[1] M. Dacier 和 Y. Deswarte，“特权图：类型化访问矩阵模型的扩展”，载于欧洲计算机安全研究研讨会。施普林格，1994 年，第 319–334 页。 [2] B. Schneier，“攻击树”，《Dr. Dobb's》杂志，第 24 卷，第 12 期，第 21–29 页，1999 年。[3] S. Mauw 和 M. Oostdijk，“攻击树基础”，载于国际信息安全与密码学会议。施普林格，2005 年，第 186–198 页。[4] B. Kordy、S. Mauw、S. Radomirovi´c 和 P. Schweitzer，“攻击–防御树基础”，载于安全与信任形式化方面国际研讨会。施普林格，2010 年，第 80–95 页。[5] O. Sheyner、J. Haines、S. Jha、R. Lippmann 和 J. M. Wing，“攻击图的自动生成与分析”，载于*2002*年*IEEE*安全与隐私研讨会论文集。IEEE，2002 年，第 273–284 页。[6] “slang - SystemVerilog语言服务”。 [在线]。网址：https://github.com/MikePopoloski/slang [7] M. Franz、C. T. Lopes、G. Huck、Y. Dong、O. Sumer 和 G. D. Bader，“Cytoscape.js：用于可视化和分析的图论库”，《生物信息学》，第 32 卷，第 2 期，第 309–311 页，2016 年。[8] M. R. Clarkson 和 F. B. Schneider，“超属性”，《计算机安全杂志》，第 18 卷，第 6 期，第 1157–1210 页，2010 年。 [9] W. Hu、A. Althoff、A. Ardeshiricham 和 R. Kastner，“迈向属性驱动的硬件安全”，载于*2016*年第*17*届微处理器与*SoC*测试和验证国际研讨会（*MTV*）。IEEE，2016 年，第 51–56 页。[10] C. Phillips 和 L. P. Swiler，“用于网络漏洞分析的基于图的系统”，载于*1998*年新安全范式研讨会论文集，1998 年，第 71–79 页。[11] X. Ou、S. Govindavajhala 和 A. W. Appel，“Mulval：基于逻辑的网络安全分析器”，载于*USENIX*安全研讨会，第 8 卷。马里兰州巴尔的摩，2005 年，第 113–128 页。[12] S. Koteshwara，“利用图分析评估服务器硬件架构的安全风险”，载于*2021*年亚洲硬件安全与信任研讨会（*AsianHOST*）。IEEE，2021 年，第 1–4 页。[13] S. K. Saha、J. M. Mbongue 和 C. Bobda，“评估片上系统安全性的度量指标”，载于*2022*年*IEEE*硬件安全与信任国际研讨会（*HOST*）。IEEE，2022 年，第 113–116 页。[14] W. Hu、A. Ardeshiricham 和 R. Kastner，“硬件信息流跟踪”，《ACM计算调查》（CSUR），第 54 卷，第 39 期，2021 年，第 1–4 页。[15] “OpenTitan”，https://opentitan.org/，访问日期：2022-07-26。[16] “ISO/IEC 17825:2016 信息技术——安全技术——用于缓解针对密码模块的非侵入式攻击类别的测试方法”，https://www.iso.org/standard/60612.html，2016 年。[17] N. Veyrat-Charvillon 和 F.-X. Standaert，“互信息分析：如何、何时以及为何？”，载于密码硬件与嵌入式系统国际研讨会。施普林格，2009 年，第 429–443 页。[18] A. Moradi、B. Richter、T. Schneider 和 F.-X. Standaert，“使用 $\chi^2$-检验进行泄漏检测”，《IACR密码硬件与嵌入式系统汇刊》，第 209–237 页，2018 年。[19] A. Althoff、J. Blackstone 和 R. Kastner，“整体功耗侧信道泄漏评估：迈向稳健的多维度量指标”，载于*2019*年*IEEE/ACM*计算机辅助设计国际会议（*ICCAD*）。IEEE，2019 年，第 1–8 页。[20] M. Rostami、F. Koushanfar 和 R. Karri，“硬件安全入门：模型、方法与度量指标”，《IEEE会刊》，第 102 卷，第 8 期，第 1283–1295 页，2014 年。[21] K. Xiao、A. Nahiyan 和 M. Tehranipoor，“集成电路设计中的安全规则检查”，《Computer》，第 49 卷，第 8 期，第 54–61 页，2016 年。[22] A. Ardeshiricham、W. Hu、J. Marxen 和 R. Kastner，“用于可证明安全硬件设计的寄存器传输级信息流跟踪”，载于*2017*年欧洲设计、自动化与测试会议及展览（*DATE*）。IEEE，2017 年，第 1691–1696 页。[23] W. Hu、J. Oberg、A. Irturk、M. Tiwari、T. Sherwood、D. Mu 和 R. Kastner，“门级信息流跟踪的理论基础”，《IEEE集成电路与系统计算机辅助设计汇刊》，第 30 卷，第 8 期，第 1128–1140 页，2011 年。[24] D. Zhang、Y. Wang、G. E. Suh 和 A. C. Myers，“用于时序敏感信息流安全的硬件描述语言”，ACM SIGPLAN Notices，第 50 卷，第 4 期，第 503–516 页，2015 年。[25] G. K. Contreras、A. Nahiyan、S. Bhunia、D. Forte 和 M. Tehranipoor，“ 针对片上系统资产保护的面向测试设计漏洞利用安全漏洞分析”，载于*2017*年第*22*届亚洲及南太平洋设计自动化会议（*ASP-DAC*）。IEEE，2017 年，第 617–622 页。[26] J. Cruz、P. Mishra 和 S. Bhunia，“度量指标至关重要：衡量电子产品信任度的艺术”，载于*2019*年第*56*届*ACM/IEEE*设计自动化会议（*DAC*）。IEEE，2019 年，第 1–4 页。[27] G. Smith，“定量信息流的基础”，载于软件科学与计算结构基础国际会议。施普林格，2009 年，第 288–302 页。[28] B. Mao、W. Hu、A. Althoff、J. Matai、Y. Tai、D. Mu、T. Sherwood 和 R. Kastner，“密码硬件设计中时序信道安全的定量分析”，《IEEE集成电路与系统计算机辅助设计汇刊》，第 37 卷，第 9 期，第 1719–1732 页，2017 年。[29] X. Guo、R. G. Dutta、J. He、M. M. Tehranipoor 和 Y. Jin，“QIF-Verilog：基于定量信息流、用于硅前安全评估的硬件描述语言”，载于*2019*年*IEEE*硬件安全与信任国际研讨会（*HOST*）。IEEE，2019 年，第 91–100 页。[30] L. M. Reimann、L. Hanel、D. Sisejkovic、F. Merchant 和 R. Leupers，“QFlow：Verilog中用于安全感知硬件设计的定量信息流”，载于*2021*年第*39*届*IEEE*计算机设计国际会议（*ICCD*）。IEEE，2021 年，第 603–607 页。[31] A. Nahiyan、J. Park、M. He、Y. Iskander、F. Farahmandi、D. Forte 和 M. Tehranipoor，“SCRIPT：利用信息流跟踪和模式生成进行功耗侧信道漏洞评估的计算机辅助设计框架”，《ACM电子系统设计自动化汇刊》（TODAES），第 25 卷，第 3 期，第 1–27 页，2020 年。[32] S.-Y. Yu、R. Yasaei、Q. Zhou、T. Nguyen 和 M. A. Al Faruque，“HW2VEC：用于自动化硬件安全的图学习工具”，载于*2021*年*IEEE*硬件安全与信任国际研讨会（*HOST*）。IEEE，2021 年，第 13–23 页。