获 IEEE硬件安全与可信国际研讨会 (HOST) 2026年在美国华盛顿哥伦比亚特区接收的论文。

InterPUF：通过物理不可克隆函数和多方计算实现面向可重构中介层的分布式认证

Ishraq Tashdid[1],Tasnuva Farheen[2],Sazadur Rahman[1]美国佛罗里达州奥兰多市中佛罗里达大学， [2]美国路易斯安那州巴吞鲁日市路易斯安那州立大学{ishraq.tashdid, sazadur.rahman}@ucf.edu[1],tfarheen@lsu.edu[2]。

arXiv:2601.11368v1 [cs.CR] 16 Jan 2026

摘要—现代系统级封装（SiP）平台正迅速采用可重构中介层，以支持异构多供应商生态系统中的即插即用芯粒集成。然而，这种灵活性带来了严峻的信任挑战，因为传统安全对策难以在这些去中心化、制造后可编程的环境中实现扩展或适应。本文提出 INTERPUF，这是一种紧凑、可扩展的认证框架，可将中介层转变为分布式信任根。在其核心，INTERPUF 在可重构互连中嵌入基于路径的差分延迟物理不可克隆函数（PUF），并利用多方计算（MPC）保护其评估过程。所提出的架构在各种芯粒中仅引入 0.23% 的面积开销和 0.072%的功耗开销，同时将身份验证延迟保持在几十纳秒以内。使用 PYPUF 进行的仿真结果表明，即使在工艺、电压和温度变化下，该方案仍具有很强的唯一性、可靠性和建模抗性。通过将基于硬件的物理不可克隆函数原语与密码学哈希和协同验证相结合，INTERPUF 在不依赖任何集中式信任锚的情况下实现了最小信任模型。

索引词—**安全异构集成、物理不可克隆函数。**

I. 引言

集成电路制造工艺成本的上升、技术节点的不断缩小以及集成电路日益增加的复杂性，迫使原始设备制造商 (OEM) 将其设计、集成和制造的部分工作外包给第三方供应商、合同晶圆厂和封装厂。相比之下，全定制 ASIC 允许单个设计公司控制规格制定、设计和制造，从而最大限度地提高灵活性和可信度 [1] ，但代价是开发周期长且非重复性工程 (NRE) 费用高昂。晶圆厂提供的工艺—

签核套件（工艺设计套件）可降低这些成本并提高良率，但会降低设计灵活性和安全性。如今，片上系统（SoC）设计高度依赖可复用的 IP 模块，使完全定制设计变得不切实际。这种由复用驱动的范式加速了创新，同时降低了成本并缩短了上市时间。如今，微电子行业正通过异构集成，将这种模块化方法扩展到芯粒、中介层和先进的 2.5D/3D封装 [2]。在这些系统级封装（SiP）架构中，来自不同供应商的预制逻辑电路、存储器、模拟电路和加速器芯粒被组装在共享的中介层平台上。无源和有源中介层 [3]–[8] 可实现高密度、高带宽的芯粒通信，而新兴的可重新配置中介层——

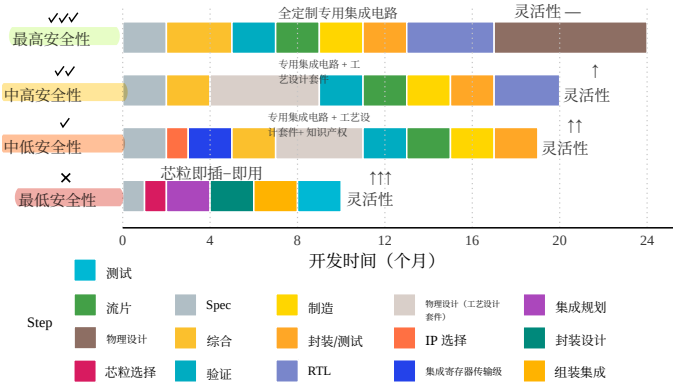

图 1：不同设计流程的开发时间对比。每个柱状图展示了设计各阶段的顺序和持续时间。随着设计流程从全定制逐步演变为基于芯粒的集成，寄存器传输级（RTL）设计或物理实现等冗余阶段被缩减或消除，从而缩短了产品上市时间。

则进一步实现即插即用集成 [9]–[11]。借助这些可编程互连结构，设计人员可以像拼搭乐高积木一样灵活组合芯粒，通过重新配置延长产品生命周期，并快速制作新设计的原型。

向涉及多个全球实体的水平分布式集成电路供应链转型，引入了一系列安全威胁，包括知识产权盗用、硬件木马植入、仿冒、逆向工程和过量生产 [12]。相比之下，早期的全定制设计流程将规格制定、设计和制造保留在内部，从而维护了机密性并最大限度地降低了外部风险 [13]。随着设计和制造职责转移至外包晶圆厂、知识产权供应商和小芯片供应商，攻击面大幅扩大 [14]。设计流程各阶段的威胁模型总结于表 I。知识产权供应商可能隐藏恶意逻辑 [13]；晶圆厂可能进行过量生产、逆向工程或篡改布局 [15], [16]；集成商则必须防范仿冒或重复使用的芯粒 [17]。此外，许多片上系统级防御措施，例如逻辑锁定 [13]、集成电路计量 [1]，和扫描链保护 [18]，并不直接适用于基于芯粒的系统级封装，因为集成商无法修改第三方组件。因此，每个外包环节都以安全保障换取更快的设计周期，从而形成碎片化的最小信任供应链。

基于可重构中介层的系统级封装最鲜明地暴露了速度与信任之间的权衡：可编程性支持自适应

表 I：常见设计流程中的威胁模型。[1]

设计流程	涉及实体	威胁模型
全定制 ASIC	设计公司	内部威胁
专用集成电路+工艺设计套件	+ 晶圆代工厂	+不可信晶圆代工厂
专用集成电路+工艺设计套件+知识产权	+ 供应商	+ 知识产权盗用
芯粒即插即用	+ 封装	+ 芯粒代工厂

路由和互操作性，却也集中了攻击面 [19]。因此，中介层必须充当可验证的信任根，否则模块化将成为系统性风险。现有的系统级封装信任方案依赖集中式信任锚（可信芯粒/端口）[20], [21]，从而造成瓶颈和单点故障；密码学握手会增加延迟 [22]；许多多方计算方案仍假设存在中心验证器 [23]；而以中介层为中心的信任根仍然固定，无法跨可重构的多供应商网格进行扩展 [24]。因此，我们需要分布式、可扩展的信任机制。多方计算支持在不泄露秘密的情况下进行联合验证 [23]，[25]–[27]，并且已在软件中得到大规模验证 [28]–[30]。我们提出 INTERPUF：一种驻留于中介层的去中心化信任互连结构，通过多方计算验证真实性，同时绝不导出原始物理不可克隆函数输出。我们的主要贡献如下。

- 我们提出了 INTERPUF，它在可重构中介层中嵌入一种差分布线路径延迟物理不可克隆函数，并将其用作整个系统级封装的信任根。
- 我们设计了一种协议，仅对稳定后的令牌进行秘密共享，并采用芯粒加密，从而实现分布式接受/拒绝决策，无需中心验证器；原始挑战-响应对（CRPs）绝不会越过边界。
- 我们的硬件评估表明，INTERPUF 实现了极低的中介层开销（低于 0。23%）、低至 0。48% 的每个芯粒加密开销，以及低于 0。072% 的功耗，同时保持较快的身份验证延迟。
- 使用 PYPUF 的仿真结果证实了其出色的物理不可克隆函数质量（均匀性为 $\sim$ 0.5、可靠性为 98%、唯一性为 $\sim$ 0.46）；机器学习攻击只能达到随机猜测的准确率（$\sim$ 47%），验证了其韧性。
- 我们分析了在审慎信任供应链下系统抵御建模、重放、探测和拒绝服务（DoS）攻击的能力。
- 为加速后续研究并促进工业界采用，我们将在 GitHub 上以开源形式发布 Verilog 实现、评估脚本和基准数据。

本文组织如下：第 II 节回顾相关工作，第 III 节定义威胁模型，第 IV 节介绍 INTERPUF 的架构与运行方式。第 V 节概述仿真流程，第 VI 节报告评估结果，第 VII 节分析系统韧性。第 VIII 节讨论局限性，第 IX 节总结全文。

II. 背景与相关工作

异构多芯粒系统中的认证需要架构支持和可扩展的信任。本节从三个方面回顾了相关工作：1 异构集成领域的进展以及支持即插即用组装的可重构中介层；2 现有的系统级封装安全技术，这些技术往往依赖集中式信任锚或复杂的密码学；以及3 基于延迟的物理不可克隆函数，这是一种用于生成独特硬件特征签名的紧凑原语，并由此引出 INTERPUF。

[1]此处的“+”表示在前面各项的基础上进行相加。例如，ASIC+PDK+知识产权设计流程涉及设计公司、晶圆代工厂和供应商，而威胁模型则包括内部人员、不可信晶圆代工厂和知识产权盗用。常见威胁包括硬件木马、侧信道泄漏、故障注入、测试/可测试性设计滥用及伪造文档，因篇幅所限，此处不再赘述。

表 II：与先前的身份验证方法的对比。

方法	缺点	INTERPUF 的优势
GATE-SiP [20]	基于测试访问端口；易受中间人攻击	无需更改测试访问端口；可抵御中间人攻击
PQC-HI [22]	高计算成本；探测下的信息泄漏	低开销，分布式具有强大功能的检查签名哈希
SECT-HI [21]	侧重于测试加密；限制供应商灵活性	同时支持供应商和集成商安全
SAFE-SiP [23]	依赖中央枢纽；	分布式验证；不存在单点故障；可扩展且高效

TAP：测试访问端口。MITM：中间人攻击 le.

A. 异构集成与可重构中介层

异构集成（HI）是后摩尔时代微电子技术的基石，它不再将中介层视为被动载体，而是将其作为一种主动设计资源，使大规模模块化成为可能 [31]。这种模块化方法通过在多个产品之间复用经过验证的优质裸片，降低了成本，提高了良率，并缩短了设计周期。然而，如今的中介层大多针对每个系统进行定制设计，限制了复用能力，也制约了敏捷性 [10]。实际上，大多数系统级封装（SiP）中介层仍采用固定的金属布线；真正的可重构布线网格仍是一个活跃的研究方向，而非商品化实践 [9]–[11], [32], [33]。通过将芯粒功能与布线解耦，这些技术实现了即插即用组合（延迟绑定、分档和现场修复），但也带来了新的安全挑战，下一节将对此进行介绍。

B. 可重构中介层面临的挑战

硅中介层虽然具备现场可编程性，但也扩大了攻击面。与静态中介层不同，其动态重新编程能力可能为攻击者在部署后利用引入新的运行时攻击途径 [34], [35]。1 攻击者可能重新配置中介层以植入硬件木马或改变互连拓扑，从而破坏系统行为 [35], [36]。2 此外，未经授权地访问通信信道，可能劫持或重路由芯粒间的数据流 [36]。而且，3 通过动态重新配置信号路径来观测或调制敏感数据，可能造成隐蔽的侧信道泄漏 [37]。因此，要确保不断变化的配置具有可追溯来源，就必须对配置状态进行证明，为配置控制器提供安全固件，并实施生命周期监控。

C. 现有工作的不足

既有研究（表 II）主要针对静态中介层，采用制造阶段对策、硬件原语和密码协议。分割制造（SM）和互连网络（NoI）混淆通过隐藏设计信息来阻碍逆向工程 [38]。SM 对可信后端制造的依赖

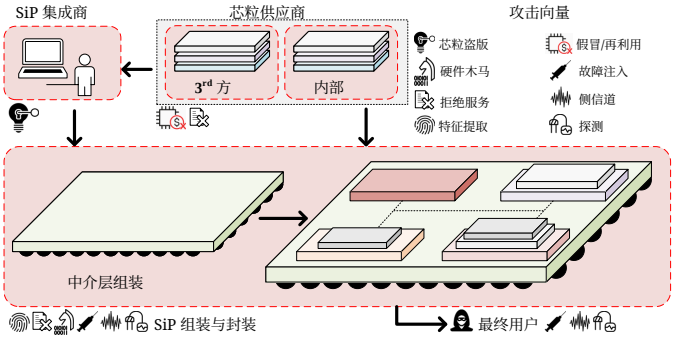

图2：SiP设计流程和生命周期。未受信任方和受信任方分别用红色和绿色方框标示。

限制了可扩展性并损害良率，而 NoI 混淆则增加了集成和验证的复杂性 [16]。模块级方案（例如芯粒硬件安全模块（CHSM）和芯粒安全IP（CSIP））能够提供认证，但代价是增加面积和设计复杂性，并且会（重新）引入单点故障 [14]。研究人员提出了后量子密码认证和加密测试协议 [21], [22]，但这些方案依赖集中式信任锚点，且会产生不可忽视的延迟开销。尽管如此，MPC辅助框架 [23] 通常仍基于协调器运行。此外，这些方法均未明确针对可重构中介层，而此类中介层的路由状态会随时间变化（如第 II-A、II-B 节所示）。总而言之，这些局限促使我们需要一种高效、可扩展、最大限度减少信任依赖且无需协调器，并专为可重构中介层互连结构量身定制的认证原语。

D. 基于延迟的物理不可克隆函数

物理不可克隆函数（PUF）包括多个系列：a 仲裁器 PUF，其中挑战比特控制两条匹配的延迟路径，由仲裁器记录竞赛结果——可产生大量挑战-响应对（CRP），但当CRP暴露时，容易受到建模攻击和侧信道攻击 [39], [40]；b 环形振荡器（RO）PUF，通过比较振荡器频率进行工作，非常适用于FPGA [41]；c 芯片间延迟PUF，通过凸点、通孔和封装走线将延迟竞赛扩展到芯片之间 [42]；以及d 互连PUF，将竞赛嵌入中介层或路由互连结构中，用于SiP认证 [43]。仲裁器PUF通常串联64–512 级以构建熵。基于延迟的物理不可克隆函数对工艺、电压、温度和老化变化较为敏感，可通过多数表决、比特预选择和轻量级纠错予以缓解。随着CRP暴露量增加，建模攻击会变得更加有效，而侧信道（功耗、时序、电磁辐射）可能泄露响应；对策包括限制CRP数量、对挑战进行哈希处理，以及将原始响应保留在芯片上。

III. 威胁模型与假设

我们的框架采用最小信任假设，即异构集成流程中的任何单一参与方都不完全可信 [44]。相关参与方包括 a 芯粒供应商（提供假冒或重复使用的芯粒），b 中介层/封装代工厂（篡改路由互连结构、未经授权的重配置、互连探测），c 系统集成商（半诚实地或恶意地滥用认证凭据），以及 d 最终用户/现场攻击者（部署后在运行时进行重新配置）。芯粒被视为具有身份的组件，其内部结构不透明；我们不验证其内部功能。此外，恶意设计且带有功能性硬件木马的芯粒不在本文讨论范围内；此类情况需要采取超出本文关注范围的额外供应链防御措施。

参与验证的各方至少是半诚实的（会遵循协议，但可能出于好奇而探查信息）。注册在可信环境或可信流程中进行，以获取参考数据（例如黄金挑战—响应信息）。运行期间不会导出原始 PUF 响应，使用参考数据可避免暴露原始挑战-响应对。考虑的攻击能力包括i 克隆/重复使用和替换，ii 旨在对 PUF 建模的重放和选择挑战尝试，iii 探测或重新路由中介层上的信号以提取信号或使信号产生偏差，以及iv 滥用可重新编程性进行配置篡改和拒绝服务（DoS）。我们承认存在侧信道攻击（功耗、电磁辐射、时序），但全面的侧信道防护超出本文范围，并将作为一项局限性进行讨论。根据对手模型，我们的安全目标包括：

- 在不暴露原始挑战-响应对的情况下验证芯粒。
- 即使供应链部分失陷，也能抵御建模、重放和克隆攻击。
- 保持对中等噪声、环境变化和老化效应的鲁棒性。
- 在大型 SiP组件中，随着网格规模和芯粒数量的增加实现高效扩展。

IV. 系统概述

本节介绍 INTERPUF 架构。该架构利用中介层集成路由、紧凑型延迟链和分布式芯粒封装器，实现多芯粒系统中的可扩展认证。第 IV-A 节详细介绍系统的高层概览，如图 3 所示；第 IV-B 节则讨论系统级操作，如图4 所示。

A. 架构

本小节对我们的架构进行高层概述，该架构将在论文发表后公开于 GitHub。虽然 PUF 本质上是一种无法通过逻辑代码完全捕获的物理现象，但我们在此对其行为进行近似，并将在第 V 节进一步介绍一个详细的基于仿真的模型。

1. 中介层网格： 系统的核心是曼哈顿网格互连，这是 [9] 为基于2.5D中介层的集成提出的即插即用基础架构。该网格由路由器瓦片构成，每个瓦片都支持与相邻瓦片之间的双向链路。这种拓扑结构兼具成本效益和可重构性，在保持大规模集成所需灵活性的同时，最大限度地降低了布线开销。每个路由器瓦片都包含一个紧凑的交换盒，用于确定信号流的方向。这种设计简单，不仅降低了面积和功耗，还能以低开销在路由器边界集成物理不可克隆函数。因此，互连结构本身

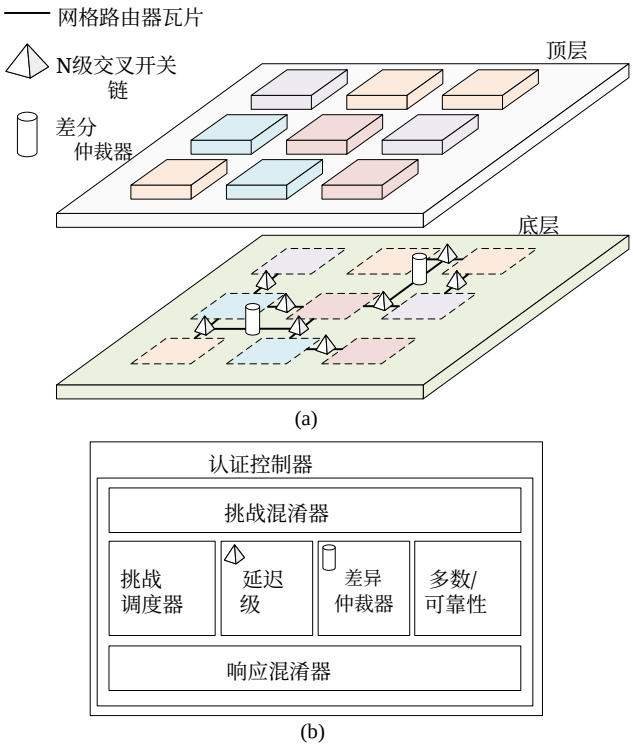

图 3：INTERPUF 的架构。(a) 系统的物理设计，其中底层集成了互连网格，并嵌入了 N 级交叉开关链和差分仲裁器；顶层承载通过互连结构相连的芯粒。(b) 认证控制器，负责管理挑战哈希、调度、差分仲裁、多数表决/可靠性表决以及响应哈希。

成为系统级认证的积极参与者，而非被动的布线结构。

1.工艺：在 INTERPUF 中，每个路由器瓦片都嵌入一条小型“N级”基于交叉开关的延迟链，作为 PUF 的基础。每个挑战会选取一对相邻路由，在网格中形成一条“A路径”和一条“B路径”。接收端处的差分仲裁器会比较这两条路径上的延迟。第 IV-B-2 节展示了这类路径对的一个示例。由于工艺偏差会使每个设备具有独特的延迟特性，仲裁器的判定结果可作为设备特定响应。对各个质询重复这一过程，可得到丰富的响应空间，并使其与互连的物理结构紧密绑定。因此，我们会实例化许多长度相近或相同的源端—接收端对；对于每一对，每次稳定竞赛贡献 1 位，跨多个路径对和置换进行汇总后，即可生成 256 位或 512 位的路由摘要，而无需在单条路径中设置 $256/512$ 专用级。仅采用在整个 PVT 范围内通过稳定性阈值的挑战。所有 PUF 级（延迟元件和交换盒）均位于中介层上；芯粒仅充当源端/接收端，仲裁器则在中介层边缘进行时分复用。以规模为例，一个 16×16网格通常具有 20–30 跳的源端—接收端距离和 40–80 对路径，每对路径有 8 种路径排列，可产生 320–640 个候选位；经过稳定性筛选（~60–80%）后，无需片上级即可轻松实现 256 位摘要。

1.B 黄金-无需中介层 自检-查 无需任何外部黄金参考即可检测路由篡改。中介层利用了这样一个事实：交叉交换盒设置比直通设置引入更大的延迟 [45]。考虑连接相邻芯粒（见图 4-2.1）的两条路径（A 和 B）之间进行的 N 级差分路径竞赛。我们从对称基线开始，将两条路径均配置为全交叉（图 4-2.2）。假设 A 更早到达（图 4-2.3），表明它赢得了竞赛。然后，我们将路径 B 的前 Z 级从交叉切换为直通（同时保持路径 A 的所有级均为交叉），并在多数表决下再次评估仲裁器结果（图 4-2.4）。对于该源端与接收端对，记录 B 能够稳定获胜时的最小 Z，作为阈值 $Z^\star$ {} 。

直观而言，$Z^\star$ 反映了两条名义上相同的路径之间有多少延迟“余量”。诚实且未被篡改的互连结构，其不同配对的 $Z^\star$ 值会紧密聚集。因此，沿某条路径出现的任何隐藏延迟（例如隐蔽插入、意外的额外缓冲或重新路由）都会使其阈值发生偏移，让其 $Z^\star$明显偏离总体均值 Z_{avg}。注册期间，控制器会收集许多本地配对的阈值向量 $Z^\star$，并将其存储在设计者安全存储库中（详见第 IV-B.2 节）。在实际运行中，定期复查会重新计算 $Z^\star$，并标记阈值偏移超出允许范围的路径，从而无需黄金参考即可进行完整性监控。如果某个休眠木马在注册时处于非激活状态，[46]，那么之后定期进行的检查（在启动时或低负载时段 opportunistically 执行）仍会在其激活时揭示其影响，因为相应的 $Z^\star$会明显偏离 Z_{avg}。

2. 认证控制器： 在INTERPUF中，认证并未通过专用封装层下放至芯粒，从而确保即使第三方供应商实施拒绝服务攻击，该方案也无法被绕过，因为认证仍是由中介层级强制执行的集中式任务。为确保这一点，嵌入互连结构中的认证控制器由多个开销低但彼此协同的组件构成，如图 3 所示。

- 一个挑战调度器，负责在网格互连结构中协调并分发挑战。
- 一个挑战混淆器，用于转换挑战以隐藏其中的任何相关性，从而增强抵御建模攻击的能力。
- 分布式延迟级和差分仲裁器，用于评估由挑战引发的时序竞争。
- 一个多数判定／可靠性单元，用于重复评估、抑制噪声并筛除不稳定挑战。这能提高在工艺、电压和温度变化下的有效性，并增强对抗对抗性扰动的能力，例如硬件木马可能会在单次实例中选择性地触发故障 [47]。
- 一种响应混淆器，它会在对外暴露最终响应位之前，对其进行哈希或编码，生成简短的认证令牌，从而确保原始延迟信息绝不会被直接观测到。

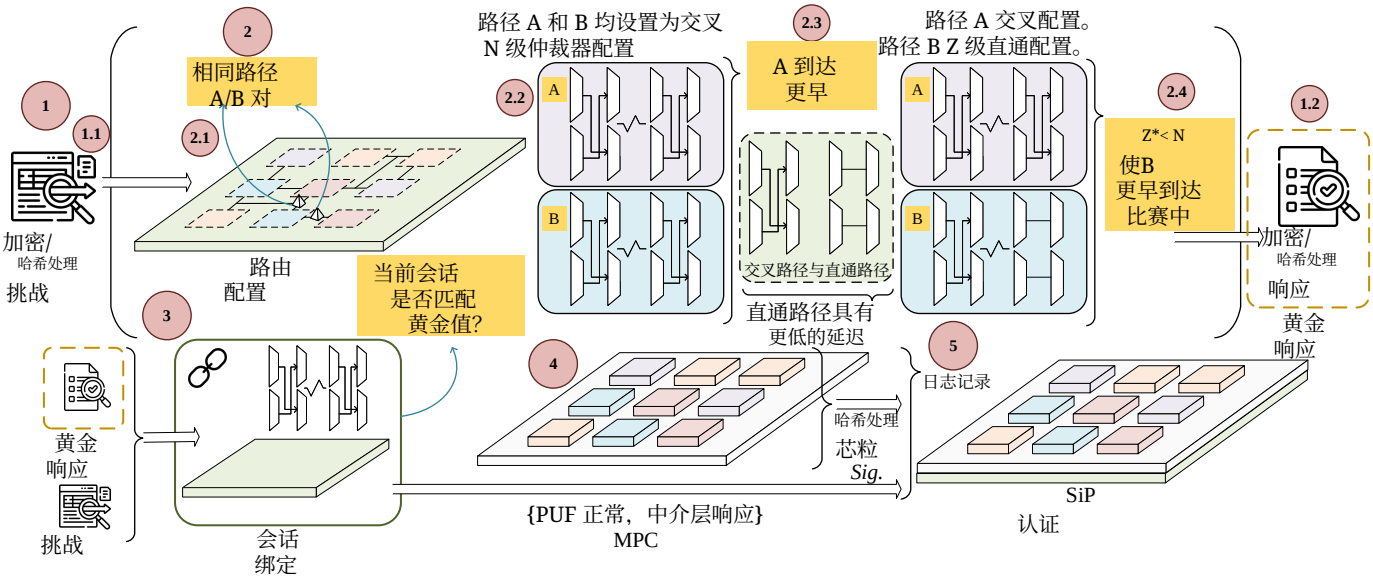

图4：INTERPUF的运行流程。

该互连级控制器建立了可扩展的双层信任层级。中介层互连结构本身成为第一道防线，用于验证路由完整性并保障通信骨干的可信性。在此基础上，各个芯粒提供自己的加密签名，这些签名只有在互连得到保护后才会进行验证。通过明确区分轻量级的互连认证与计算开销更大的芯粒级加密（见第IV-B节），该系统在实现高效性和稳健性的同时，无需在每个芯粒中重复部署逻辑电路。*B*

. $O_{\text{运行}}$ *ion*

我们在图4中提出了一种以互连中介层为中心的流程，其中：(i) 中介层仅公开由其差分延迟物理不可克隆函数生成的黄金路由摘要；(ii) 每个芯粒在注册时将其身份与该摘要绑定。在运行时，认证通过中介层（验证者）与芯粒（证明者）之间的常数轮双方多方安全计算实现。

0. 注释： 中介层 $\mathcal{I}$；芯粒 $\mathcal{C} = \{C_1, \ldots, C_N\}$。令**ch** 表示路由挑战，EPOCH 表示会话标识符，NONCE 表示每会话新值。SHA256 表示 SHA-256；HKDF 是基于HMAC的密钥派生函数（密钥派生函数）。串联表示 ‖。PRK 是伪随机密钥。

1.挑战-响应： 在注册期间，对于经过筛选的挑战集和 PVT 角点，$\mathcal{I}$ 会通过 K 次重复和多数表决来评估差分路径（图 4-1.1）。保留翻转率低于 τ 的响应。借助设计方芯粒（图 4-1.2），将接受的比特串哈希处理为定长的黄金路由摘要 $R^{\star}$。

$$R^{\star} = \text{SHA256（稳定的 PUF比特）。}$$

原始挑战-响应对不会离开 $\mathcal{I}$；只有 $R^{\star}$ 会被导出用于绑定。

2. 注册： 注册在外包半导体组装与测试（OSAT）的标准自动测试设备（ATE）上进行。ATE运行签名测试程序，测试机主机与硬件安全模块（HSM）配对，该模块可能是一个随机设计的芯粒，用于证明测试机身份、控制密码操作，并生成签名日志。但双方都需要交换哈希，以确认流程的安全性。对于精心挑选的挑战集和具有代表性的PVT角点，中介层会评估多组竞争结果，并对获准的比特进行哈希处理，生成黄金路由摘要 $R^{\star}$。可选的辅助数据（如有使用）会作为非敏感元数据存储在设备上的非易失性存储器（NVM）中。每个摘要都会与可追溯字段（设备标识符、晶圆、批次）绑定，而HSM会签署一份按设备生成的注册清单（设备ID、摘要集/版本、追溯元数据、测试机证明），并将其上传至设计方的安全存储库。OSAT仅保留通过/失败结果和非敏感日志。此外，每个芯粒 C_i 都会通过一次性承诺，将其本地身份绑定到中介层的物理不可克隆函数（PUF）

$$G_i = \text{SHA256}\big(\text{ID}_i \,\|\, \text{SIG}_i \,\|\, R^{\star} \,\|\, \text{EnrollTag}\big),$$

其中，ID_i 是设备本地标识符，SIG_i 是用于证明 ID_i的供应商签名或硅上驻留密钥。仅向集成商提供 G_i 和追溯元数据；(ID_i，SIG_i)不会离开 C_i。中介层会将 G_i （以及签名清单）存储在不安全的 NVM 中。

3.会话绑定： 中介层根据黄金摘要 $R^{\star}$ 和当前上下文（**挑**战，训练轮次）推导出每会话盐值 s，并采样一个新的随机数；这两个值在会话内公开，用于将所有证明绑定到此中介层实例，并防止重放。只有当当前挑战通过物理不可克隆函数稳定性检查（PUF 正常=1）时，才会进行认证（见图 4-3）。

4. 多方计算： 目标是判断芯粒是否持有能够重现 G_i 且绑定到 $R^{\star}$的身份材料，并生成一个与 (s,随机数) 关联的逐会话令牌，同时不泄露任一方的秘密。图 4-3 显示，在每次运行中，中介层提供其私有输入 ($G_i, R^{\star}$)，芯粒提供其私有输入 (ID_i,SIG_i)，双方则使用公开输入 (s,随机数, PUF OK)。双方使用带有不经意传输的姚氏混淆电路执行两方计算，以评估一个受 [23] 启发的固定电路 f_i。

• 该电路首先重新计算 $G_i' = \text{SHA256}(\text{ID}_i \parallel \text{SIG}_i \parallel R^\star \parallel \text{ENROLLTAG})$。
• 随后，它仅当 $G_i' = G_i$ 且物理不可克隆函数验证通过时，将 b_i 设为 true = 1。
• 最后，它派生出会话令牌 $T_i' = \text{SHA256}(G_i' \parallel s \parallel \text{随机数})$。只有（$b_i$，$T_i'$）会被揭示给中介层。
• 具体来说，中介层对 f_i 进行混淆，并将混淆表发送给芯粒，为（G_i，$R^\star$）提供输入标签；芯粒则通过不经意传输获得（ID_i，SIG_i）的标签。

• 双方将（s，随机数，物理不可克隆函数验证通过）编码为公开输入。
•芯粒计算混淆电路并返回混淆输出，中介层对其解码，获得（b_i，T_i'）；只有在 $b_i = 1$时才接受。

该两方计算协议在固定轮数内完成，与电路深度无关。

5. 日志记录： 接受后（b_i=1），中介层可记录（i，**ch**，训练轮次，随机数 T_i'）以供审计和检测重放，同时避免存储任何秘密输入。由于$T_i' = \text{SHA256}(G_i \| s \| \text{随机数})$以及盐值 s均在每次会话中重新生成，因此令牌在不同会话间无法关联。

缺少与 G_i 和 $R^\star$绑定的（ID_i，SIG_i）的冒名者，无法在两方计算检查中满足 G_i'=G_i ，因此无法生成有效的 T_i'。令牌重放无效，因为（s，随机数）每次会话都会变化。仅复制 G_i是不够的，还需要相应的身份材料。在整个协议过程中，既不会泄露 G_i ，也不会泄露（ID_i，SIG_i）。对于任意子集 $\mathcal{S} \subseteq \{1, \ldots, N\}$，中介层都会针对每个 $i \in \mathcal{S}$ 独立运行 f_i （可用时并行执行），随后对所得比特 $\{b_i\}_{i \in \mathcal{S}}$ 应用本地策略谓词（例如法定人数），以得出最终决策。

V. 建模与仿真方法

本节介绍一套完整、感知中介层的仿真流程，旨在映照寄存器传输级（RTL）实现。我们采用 PYPUF [48] 来复现参数化物理不可克隆函数（PUF）的行为，集成标准机器学习框架以实施建模攻击，并引入前处理和后处理层，真实模拟 RTL 中观察到的互连变换与哈希封装器。

A. 忠实于设计的系统抽象

该硬件架构基于曼哈顿式网格，其中每个路由器和互连都会影响物理不可克隆函数的有效级深度。在仿真中，这被抽象为网格单元位置数与水平/垂直链路数之和，与寄存器传输级逻辑保持一致。我们的默认网格大小为 4，因此得到的级数与可综合的网格式设计相符。每个仿真芯片都会实例化一个异或型-K 仲裁器系列（其中$K \geq 4$），以反映寄存器传输级中由多条链和并行仲裁器引入的非线性聚合。当寄存器传输级使用抽头反馈来引入前馈复杂度时，仿真器则通过提升的挑战特征（所选抽头索引之间的成对交互）来模拟这一过程。这确保了在硅中捕获的分离性和非线性效应能够在软件模型中得到忠实近似。

互连封装器分为三个级进行建模，与 RTL 对挑战比特进行路由和扰动的方式相一致：(i) 置换级索引，以模拟路由器重新排序；(ii) 稀疏极性翻转，以反映布线反相和模式切换；(iii) 稀疏、可逆的二进制混合矩阵，以捕捉相邻链路之间的弱耦合。每种路由器配置都会确定性地定义这些变换，确保每种互连设置都产生一致且特定于设备的挑战映射。该映射在构造上保持可逆，确保熵不会被人为增加或损失，从而映照 RTL 保持结构的路由逻辑。

B. 测量流程与 *PVT* 真实性

通过在多个模拟设备上批量生成挑战-响应对，以体现芯片间差异。在硬件中，芯片间差异源于不可控的工艺偏差；在仿真中，则为每个芯片独立设置种子，以产生不同的延迟偏置。同样，每个挑战批次采用随机路由器配置，以引入互连多样性。为真实反映运行条件，我们将工艺、电压和温度（PVT）效应建模为低概率的响应翻转。这模拟了亚稳态、电源噪声以及受温度影响的延迟偏移。每个挑战都会评估多次，并使用与寄存器传输级（RTL）相同重复深度的多数表决组合器生成稳定比特。根据挑战的稳定性对其评分，并仅保留最可靠的子集。这与实际部署中的注册过程相呼应：决策边界附近不稳定的挑战-响应对会被舍弃，从而确保传递给攻击者的数据集符合已部署硬件中实际可能暴露的数据。

C. 受控接口与摘要

与我们的寄存器传输级设计一致，质询不会直接施加到延迟链上，而是先经过二元有限域上的可逆线性变换。该哈希层在保留熵的同时，消除了外部质询与内部级基之间的固定对齐关系，从而防止直接的建模攻击。在仿真中，我们采用相同的原理：使用随机但可逆的二进制矩阵，在评估前变换质询。

未经保护的原始物理不可克隆函数比特绝不会离开系统。在寄存器传输级设计中，低开销哈希阶段会将稳定后的响应转换为认证令牌；在仿真中，我们使用一个输出 ± 1 个符号的32位扩散型组合器。只有经过可靠性筛选后，才会应用会话绑定摘要，确保不稳定性不会被掩盖。因此，外部观察者或攻击者只能看到经过混淆的令牌，而非原始竞赛结果，这忠实反映了已部署硬件的信息流。

D. 攻击者模型与训练协议

为评估系统的韧性，我们同时考察预言机模式和部署模式。在预言机模式下，所有保护措施均被禁用，暴露

表 III：INTERPUF 中互连路由和物理不可克隆函数的设计面积开销。

设计	设计面积 (μm^2)	互连路由面积 (μm^2)	开销 (%)
CVA6 [49]	345,755	812.0	0.23
NVDLA [50]	541,552	835.5	0.15
RISC-V [51]	1,309,680	790.1	0.06
Ariane [52]	1,431,536	868.3	0.06
OR1200 [53]	1,488,384	805.2	0.05

表 IV：芯粒加密的设计面积开销（每个芯粒，包含拟议的改动）。

设计	设计面积 (μm^2)	芯粒加密面积 (μm^2)	开销 (%)
CVA6 [49]	345,755	6,988.2	2.02
NVDLA [50]	541,552	7,192.1	1.33
RISC-V [51]	1,309,680	7,063.0	0.54
Ariane [52]	1,431,536	7,000.5	0.49
OR1200 [53]	1,488,384	7,108.0	0.48

原始仲裁器响应。这提供了一个保守的下限，使我们能够对 ϕ 映射特征应用标准逻辑回归，以对简单的仲裁器结构进行建模。在更贴近现实的配置中（并行异或链、特征提升挑战、互连混合），我们训练了多层感知机，其双曲正切激活的规模与挑战维度相匹配。为确保实验可复现，所有实验均使用固定的训练集和测试集。此外，启用响应令牌化后，根据接口设计，无法直接进行 CRP 建模。在这些情况下，不报告攻击准确率并非因为建模能力受限，而是因为系统接口不会提供原始数据。这反映了实际部署情况：外部实体无法访问未受保护的比特流。因此，预言机模式下报告的所有攻击结果都代表严格的最坏情况上限，而部署模式则忠实反映了经加固的寄存器传输级接口。

E. 指标计算

我们评估了三项典型的物理不可克隆函数（PUF）指标：唯一性（设备之间的平均汉明距离）、均匀性（每台设备输出 1 的平均比例）和可靠性（多次读取结果的一致性）。值得注意的是，这些指标是在令牌化之前测量的，这与寄存器传输级多数表决器稳定响应的方式相一致。使用令牌化输出时，我们采用代理测量方法（计算各令牌之间的逐位差异）以保持可比性，所有结果均在第 VI-E 节报告。

VI. 评估

本节通过分析 INTERPUF 的面积、时序和功耗开销，并将其与其他相关方案进行比较，同时结合上一节给出的仿真结果，评估 INTERPUF 的实用性。

A. 实验设置

我们使用 Verilog 实现了 INTERPUF，并采用 Synopsys Design Compiler 和 SAED 14 纳米标准单元库对设计进行综合。我们利用综合后网表提取功耗、面积和时序指标。所有实验均在一台配备 32 个核心和 190 GB 内存的双路英特尔至强（Skylake）服务器上进行。综合后的设计通过了功能验证，并在典型 PVT 条件下实现了 3 GHz 的时序收敛。

B. 面积效率与可扩展性

面积效率是异构多芯粒系统中安全原语最关键的推动因素之一。当安全机制消耗过多硅资源时，在规模化应用中很快就会变得不切实际，因为单个封装中可能集成数十个芯粒。INTERPUF 的关键优势在于其架构：它没有在每个芯粒中复制成本高昂的加密引擎，而是将认证直接嵌入互连结构中。这种设计确保大部分安全成本是在 SiP 级别的一次性开销，而每个芯粒的边际开销仍可忽略不计。

表 V：与近期研究的面积比较。

Work	查找表资源	FF	面积 (mm^2)
SECT-HI [21]	–	–	5.11
PQC-HI（Kyber+Dilithium） [22]	76，999	49，993	–
PQC-HI（仅 Kyber） [22]	1，842	1，634	–
INTERPUF（1 个芯粒）	**1,915**	**1,160**	**0.0078**
INTERPUF（32 个芯粒）	**61,280**	**37,120**	**0.2501**

表 III 展示了这一特性，因此面积大幅减少。路由结构与嵌入式物理不可克隆函数合计占用不到 $900\ \mu m^2$，相较于片上系统基准，其开销仅为 0.05–0.23%。重要的是，这项开销不会随系统中芯粒数量的增加而增长：安全互连一旦部署到位，便可在不增加路由面积的情况下对更多芯粒进行身份验证。换言之，互连层安全的基本成本是封装级固定成本，而非每个芯粒各自承担的成本。

此外，表 IV 汇总的数据表明，每个芯粒都必须集成一个约为 $7{,}000\ \mu m^2$ 的加密核心，每项设计的开销因此达到 0.5–2.0%。单独来看，这些成本似乎不高，但会随芯粒数量线性增长。一个包含 32 个芯粒的系统需要为 32 个冗余加密引擎付费，浪费了原本可用于计算或存储器的硅资源。通过消除这些模块，INTERPUF 在大规模系统中将开销降低了近一个数量级。

如表 V 所示，与近期的安全硬件解决方案相比，INTERPUF 的可扩展性优势更加明显。对于单个芯粒，~~INTERPUF 仅需 0.0078 mm^2，这已经比后量子密码加速器小一个数量级，也比 SECT-HI 等基于安全飞地的方案小数倍 [21]。~~扩展至 32 个芯粒时，总开销仍仅为 0.2501 mm^2，，远低于竞争方案。这证明，与大多数替代方案不同，INTERPUF 不会因规模扩大而失效：即使芯粒数量达到最严苛的要求，其安全机制仍然紧凑。

综合来看，这些结果充分证明了 INTERPUF 的优势。它引入的固定 SiP 级互连开销不到 1%，避免了独立加密带来的逐芯粒线性惩罚，并能平稳扩展至数十个

表 VI：INTERPUF 中互连布线的功耗开销。

设计	基准功耗 (mW)	互连功耗 (mW)	开销 (%)
CVA6 [49]	12.896	0.0093	0.072
NVDLA [50]	185.140	0.0098	0.005
RISC-V [51]	59.164	0.0087	0.015
Ariane [52]	94.157	0.0102	0.011
或1200 [53]	106.610	0.0091	0.009

表格 VII: 功耗 开销 of 芯粒 加密（每个芯粒，采用拟议的改动）。

设计	基准功耗 (mW)	芯粒加密功耗 (mW)	开销 (%)
CVA6 [49]	12.896	0.2357	1.83
NVDLA [50]	185.140	0.2394	0.13
RISC-V [51]	59.164	0.2518	0.43
Ariane [52]	94.157	0.2463	0.26
或1200 [53]	106.610	0.2485	0.23

个芯粒，同时不超出现代片上系统的面积预算。此前的研究均未能展现这种效率与可扩展性的平衡，因此 INTERPUF 尤其适合大规模安全异构集成。

C. 功耗效率与开销分析

对于多芯粒系统中的安全原语而言，功耗开销是一个关键考量因素，因为任何持续性开销都会直接影响运行时能耗预算。表 VI 显示，INTERPUF 中的互连布线和嵌入式物理不可克隆函数在所有评估的片上系统中的功耗均低于 0.011 mW；相对于基线功耗预算（其范围已达数十至数百毫瓦），开销仅为 0.005%–0.072%。这一极低数值并非偶然，而是直接源于该设计在架构上的简洁性。与传统密码加速器不同，INTERPUF 无需宽数据通路、深流水线和繁重的算术运算，而是依赖紧凑的组合逻辑和少量低开销时序寄存器，因此开关活动天然较低。因此，测得的动态功耗低于 10 µW，远低于典型片上系统的功耗变化范围，使得这一开销在实际应用中几乎难以察觉。此外，表 VII 展示了与每个芯粒加密模块的对比：后者每个功耗约为 0.24 mW，开销则因主机设计而异，为 0.2%–2.0%。即便如此，对于典型的开源片上系统／芯粒而言，扩展时的开销仍然较大。通过加入时钟门控，或将加密核心重新用于芯粒的功能用途，可以进一步降低这一开销。

与其他研究相比，INTERPUF 无论规模大小都能保持极小的占用面积，其紧凑的逻辑电路也非常适合采用时钟门控和操作数隔离等标准低功耗技术，从而进一步降低功耗。SECT-HI 和 PQC-HI 等近期研究未报告功耗结果，但考虑到它们的面积达到毫米级且依赖复杂的密码原语，预计其功耗开销会显著更大 [21], [22]。因此，INTERPUF

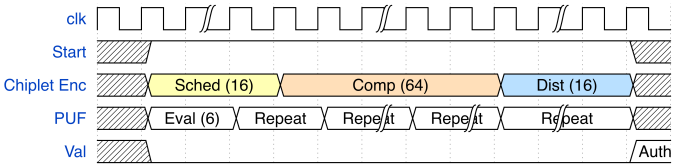

图5：INTER-PUF 中认证阶段的波形图。互连物理不可克隆函数在 6 个周期内完成，并可在单个 SHA-256 窗口（96 个周期）内重复执行多次。这样可确保在认证开始前，互连验证已确立为信任根。

提供了安全性高得多、功耗开销几乎可忽略不计的解决方案。结合其面积效率，这些结果表明，INTERPUF 不仅缩小了硅片面积占用，还确保了安全异构集成的能源可持续性。

D. 认证延迟分解

INTERPUF 中的认证分为两个紧密耦合的阶段：首先使用嵌入式物理不可克隆函数对互连结构进行认证，然后通过 SHA256 流水线验证芯粒级声明。这种分阶段方法可确保系统级封装的通信骨干——互连——在芯粒接入系统之前确立为信任根。

互连认证的开销极低，总共只需 6 个周期：1 个周期用于挑战调度，随后 5 个周期用于 PUF 评估。在 3 GHz 的工作频率下，这相当于约 2 ns。由于占用资源很少，互连可以多次重新评估，几乎不会产生额外开销。例如，在计算单个 SHA256 摘要所需的 96 个周期内，互连最多可认证 $96/6 \approx 16$ 次。这使得重复测试、通过多数表决抑制错误以及轮换路径配置成为可能，从而增强建模抗性，并在芯粒级认证开始前以高置信度验证路由结构。

芯粒认证通过传输哈希签名来完成，并由 SHA256 流水线进行验证。该哈希阶段总共需要 96 个周期，其中消息调度占 16 个周期，压缩轮次占 64 个周期，之后还有少量控制开销；在 3GHz 下，总耗时不到 32ns。此过程与正在进行的互连验证并行运行，因此芯粒摘要准备就绪时，互连结构已经重新检查过数次。

为将每次接受操作绑定到当前由 PUF 派生的会话，我们添加一个常数轮次的两方计算（2PC）阶段（采用不经意传输的姚氏混淆电路）。在线开销主要来自发送一个用于固定函数的混淆电路 f_i （重新计算芯粒身份材料和 $R^\star$ 的 SHA256，并与已存储的承诺进行相等性检查，然后派生会话令牌），再加上一批不经意传输扩展消息，最后公开输出。使用半门时，该电路的混淆电路大小约为 0.6–1.0 MB，安全性为 128 位；在 32 GB/s 的封装内链路上，串行传输耗时为 ≈ 20–$30\,\mu$s；在 128 GB/s 的链路上则为 ≈ 5–$8\,\mu$s。

表 VIII：仿真指标和建模攻击性能（4 个芯粒，PYPUF）。

指标	平均值	标准差
均匀性	0.4986	0.0028
Bias	0.0049	0.0037
唯一性（汉明距离）	0.4648	0.0734
可靠性	0.9816	–
芯片内汉明距离	0.0189	0.0008
位别名	0.4986	–
位翻转敏感度	0.5140	–
建模攻击（准确率）	0.4675	–
建模攻击（曲线下面积）	0.4899	–

轮次复杂度为常数（两到三次交互），因此延迟受带宽限制，而非受深度限制。因此，端到端的单芯粒路径为

$$t_{\mathrm{PUF}} \approx 2\,\mathrm{ns}, \quad t_{\mathrm{SHA256}} < 32\,\mathrm{ns}, \quad t_{\mathrm{2pc}} \approx 5\text{–}30\,\mu\mathrm{s},$$

纳秒级的物理不可克隆函数和哈希阶段完全被微秒级的两方计算所覆盖。多个芯粒可在并行通道中处理，同时互连持续进行周期性 PUF自检。

互连与芯粒认证之间的时序关系如图5所示：互连在少数几个周期内即可完成，并且可以无气泡地重复执行；SHA256 流水线则在单个摘要轮次中处理芯粒签名，而常数轮两方计算会增加一个有界、由带宽决定的尾部延迟。这种重叠结构最大限度地降低了端到端延迟，并建立了清晰的信任层级：先保护互连，再以互连为基准认证芯粒，从而为可扩展的多芯粒系统同时提供速度与稳健性。

*E. PUF*仿真与建模攻击结果

为评估所提出的 INTERPUF 的稳健性，我们使用 PYPUF 框架进行了详细仿真（表VIII）。在噪声和工艺偏差条件下生成了五个合成芯粒实例，并通过基于路由器的置换和稀疏翻转变换挑战。评估结果证实了其良好的统计特性：平均均匀性为 0.4986 ± 0.0028，接近理想值 0.5，偏差仍可忽略，为 0.0049。可靠性高达 98.2%，重复查询中的芯片内汉明距离严格限制在 0.0189，证实了其稳定性。唯一性（以平均芯片间汉明距离衡量）为 0.4648，接近理想值 0.5，而位别名保持均衡，为 0.4986。位翻转敏感度的平均值为 0.5140，符合预期的雪崩特性。为进一步检验其韧性，我们使用逻辑回归进行了机器学习攻击，该模型以 8000 个挑战-响应对进行训练，并在 3000 个未见过的挑战-响应对上测试。模型仅达到 46.7% 的准确率，曲线下面积为

0。4899，这并不优于随机猜测（图6）。这证实，INTERPUF 中的变换和哈希显著增加了预测建模的难度，同时保留了与信任无关的保障。

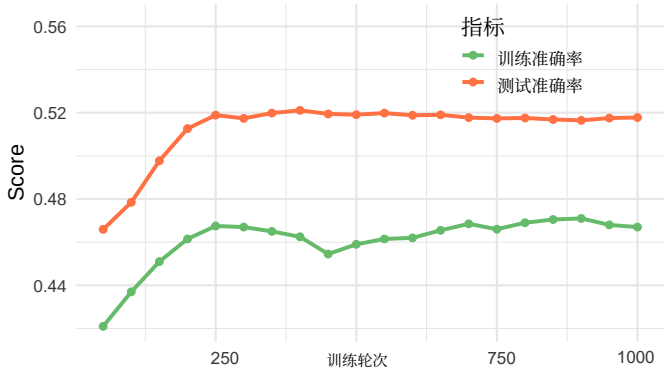

图6：INTERPUF 建模攻击的训练曲线，展示了在 1000 个训练轮次中的训练准确率和测试准确率。训练准确率和测试准确率分别趋近于 46% 和 52%，证实了其能够抵御学习模型。

VII. 安全性分析

对异构多芯粒系统的安全性进行评估时，不仅要考察其密码学强度，还要考虑其抵御各种实际攻击的结构韧性。由于互连充当通信骨干，且芯粒本身可能来自不同供应商，有效的解决方案必须能够抵御建模攻击、重放和降级策略、假冒组件插入、封装级篡改、拒绝服务以及侧信道利用。本节分析了所提架构如何应对这些威胁，并证明在保持多芯粒系统可扩展性的同时，只需极低的开销即可实现稳健的认证。

A. 建模与可靠性攻击

针对基于延迟的物理不可克隆函数的机器学习攻击通常依赖于大量稳定的挑战-响应对。在本设计中，两种机制可有效阻碍此类攻击。首先，响应在公开前始终经过掩码处理，因此攻击者无法看到原始位翻转或稳定性模式；其次，每次会话中的有效质询都会以不同方式进行置换和掩码处理，从而阻止跨会话数据集聚合。此外，通过多数表决进行重复评估可消除可利用的偏差，而尝试门控则限制了攻击者在实际中能够收集的质询数量。这些防御措施共同大幅降低了构建准确预测模型的可行性，如图6所示。

B. 重放、质询修改

身份验证记录与每次启动盐值和会话随机数绑定，确保相同的顶层挑战在不同电源周期中不会映射到相同的内部状态。这一特性使重放攻击失效，先前记录的身份验证记录在未来会话中也无法使用。任何缺失的预期响应或掩码信号都会触发拒绝，芯粒也因此无法被接纳。由于 SHA 极其敏感，修改攻击同样毫无作用（图 7）。

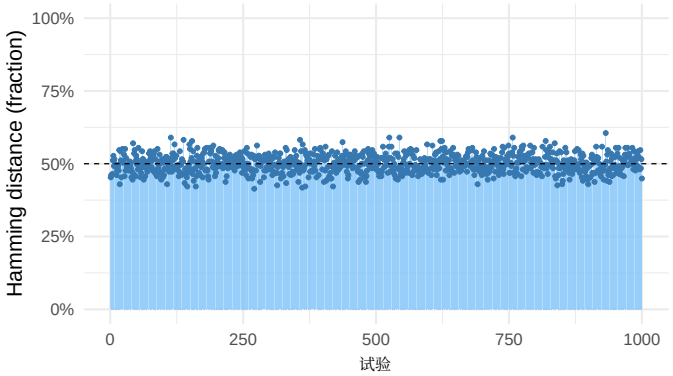

图 7：1000 次试验中的汉明距离分布。该图展示了令牌的稳定性，距离保持在 50% 左右，符合高唯一性的特征。

C. 恶意或伪造芯粒

恶意芯粒可能试图冒充有效身份，或操纵互连流量以引入偏差。该系统会先对互连进行认证，再考虑接纳芯粒，从而防止此类攻击。由于骨干网络首先受到保护，任何插入伪造组件的尝试都只能发生在互连经过多次验证之后。完成认证后，真实芯粒会传输哈希签名，并将其与参考摘要进行验证。由于没有正确的秘密上下文，重放或伪造的签名无法通过验证，因此恶意芯粒实际上会被排除在外。

D. 中介层与封装级攻击

在封装级，攻击者可能尝试通过插入可编程延迟，或经由中介层重新路由信号，来模拟有效的延迟路径。然而，质询与响应的动态哈希使这种做法难以实现：攻击者必须实时复制以设备唯一值为密钥的每次启动重新映射和掩码处理，并在许多并行评估中完成。单个认证会话中的重复证明会显著提高路径覆盖率，从而暴露中介层级操纵引入的细微偏差。

E. 拒绝服务与吞吐量滥用

攻击者可能试图用挑战淹没系统，或故意拖延事务。认证逻辑会限制尝试次数并设置冷却时间，从而防止资源耗尽。由于每次互连证明只需几个周期即可完成，系统能够频繁地重复检查，同时维持吞吐量。此外，芯粒级认证与互连检查并行运行，确保即使遭遇部分拒绝服务攻击，系统也不会陷入停滞。

F. 侧信道与故障注入

由于原始响应从未暴露，通过功耗或时序侧信道泄露的信息已被混淆，且依赖于会话。逻辑电路本身规模精简，采用恒定时延控制路径，开关活动极低，与大型密码学模块相比，可利用的泄露更少。若有需要，还可采用时钟门控、操作数隔离和随机化复查周期等成熟对策，进一步增强韧性。通过对重复评估结果进行多数表决，可缓解故障注入或环境漂移的影响，使系统能够拒绝不一致的结果，并在变化条件下保持稳健。

G. 移除攻击

在基于芯粒的集成中，恶意晶圆厂或供应商可能试图移除认证电路、绕过它，或插入恶意修改。在 INTERPUF 中，通过将认证直接嵌入互连结构，而不是将其视为可拆卸的附加组件，确保了系统的韧性。由于安全原语与路由和交换逻辑交织在一起，因此不存在可将其隔离并移除的明确边界。响应还经过混淆，并与正常的互连活动融合，使其无法与功能信号区分开来。任何修改都会破坏通信一致性，并在反复进行挑战—响应检查时迅速暴露，从而显著增加移除攻击的难度。

VIII. 讨论与局限性

INTERPUF 展现了多项优势。1 它以几乎可忽略的面积和功耗开销实现认证，2 能够高效扩展以支持不断增加的芯粒数量，并且3 通过结合互连级 PUF 证明与芯粒级哈希签名，实施严格的最小信任模型。与以往集中式或重量级的密码学方案相比，这些特性使其开销低、实用性强，非常适合异构集成。尽管如此，仍存在一些局限。我们的评估仅限于 RTL 级原型和仿真，因此仍需在更广泛的 PVT 变化、老化和环境应力条件下进行真实硅验证，以确认其长期稳健性。我们尚未构建或评估 FPGA 或硅原型；计划将这两项工作作为下一步，以验证功能、量化 PVT/老化条件下的稳定性，并测量硬件上的端到端吞吐量和延迟。我们的后续计划包括在受控工艺角条件下流片，并对认证控制器进行基于 FPGA 的仿真，以补充当前基于综合和仿真的结果。

IX. 结论

INTERPUF 提出了一种低开销的中介层驻留式认证框架，将基于延迟的物理不可克隆函数与多方计算相结合，从而在异构芯粒系统中实施严格的最小信任模型。通过将信任根嵌入互连结构，并结合芯粒级密码签名，该方案以可忽略的面积和功耗开销，实现了针对建模、重放和假冒攻击的强大韧性。我们的寄存器传输级实现与仿真结果证实，该设计能够随芯粒数量高效扩展，为安全、可重构的系统级封装集成提供了一条切实可行的路径。

参考文献

[1] M. Tehranipoor 和 C. Wang，《硬件安全与信任导论》。Springer Science & Business Media，2011年。[2] F. Sheikh 等，《2.5D 和 3D 异构集成：新兴应用》，IEEE 固态电路杂志，第 13 卷，第 4 期，第 77–87 页，2021年。[3] J. H. Lau，《采用硅通孔中介层的多系统与异构集成的最新进展与趋势》，IEEE 元件、封装与制造技术汇刊，第 13 卷，第 1 期，第 3–25 页，2023年。[4] D. Stow、Y. Xie、T. Siddiqua 和 G. H. Loh，《利用有源和无源中介层实现可扩展高性能系统的经济高效设计》，载于 2017 IEEE/ACM 国际计算机辅助设计会议（ICCAD），2017年，第 728–735 页。[5] I. Tashdid、V. Terry、J. Merkel、T. Farheen 和 S. Rahman，《BeyondPPA：面向布线后可靠性感知宏单元布局的类人强化学习》，载于计算机辅助设计机器学习国际研讨会，2025年。[6] I. Tashdid、D. Saiham、N. Anjum、T. Farheen 和 S. Rahman，《Ecologic：通过嵌入式 FPGA 增强型片上系统实现循环、混淆和自适应逻辑》，arXiv 预印本 arXiv:2508.04516，2025年。[7] D. Saiham、M. Rabadi、D. Wu 和 S. Rahman，《光子互连能否用于全同态加密加速器中的高吞吐量存储器访问？》，载于 2025 IEEE/ACM 低功耗电子与设计国际研讨会（ISLPED），2025年，第 1–7 页。[8] A. Ayes、E. G. Friedman 和 M. Wolf，《面向异构芯粒集成系统的中介层网络协同设计》，载于 2025 IEEE 电路与系统国际研讨会（ISCAS），2025年，第 1–5 页。[9] Z. Li 和 D. Wentzlaff，《Lucie：用于即插即用集成的通用芯粒中介层设计框架》，载于 2024年第57届 IEEE/ACM 微体系结构国际研讨会（MICRO），2024年，第 423–436 页。[10] F. Li、Y. Wang、Y. Cheng、Y. Wang、Y. Han、H. Li 和 X. Li，《Gia：用于敏捷芯粒集成的可复用通用中介层架构》，载于第41届 IEEE/ACM 国际计算机辅助设计会议论文集，2022年，第 1–9 页。[11] B. Jiao、L. Xu、X. Yu、H. Yang、H. Zhu、Y. Wang、J. Zhu、D. Wen、L. Wang、J. Tao、C. Chen、Y. Han、Q. Liu、N. Sun 和 M. Liu，《FPIA：利用可复用硅中介层上的现场可编程互连结构实现通信感知多芯粒集成》，IEEE 电路与系统汇刊 I：常规论文，第 71 卷，第 9 期，第 4156–4168 页，2024年。[12] B. Shakya、M. Tehranipoor、S. Bhunia 和 D. Forte，《硬件混淆导论：动机、方法与评估》，载于《通过混淆实现硬件保护》。Springer，2017年，第 3–32 页。[13] M. T. Rahman 等，《纵深防御：助力逻辑加锁取得成功的方案》，Integration，第 72 卷，第 39–57 页，2020年。[14] N. V. 等，《ToSHI——迈向安全异构集成：安全风险、威胁评估与保障》，密码学电子印刷档案，论文 2022/984，2022年。 [在线]。可访问：https://eprint.iacr. org/ 2022 984 / M. S. Rahman、H. Li、R. Guo、F. Rahman、F. Farahmandi 和 M. Tehranipoor，《[15] LL-ATPG：逻辑加锁感知测试——在不可信环境中使用代客钥匙》，载于 2021 IEEE 国际测试会议（ITC）。IEEE，2021年，第 180–189 页。[16] S. Rahman 等，《LLE：通过末级编辑缓解集成电路盗版与逆向工程》，载于测试与失效分析国际研讨会，第 84741 卷。ASM International，2023年，第 360–369 页。[17] F. Farahmandi、M. S. Rahman、S. R. Rajendran 和 M. Tehranipoor，《硬件安全计算机辅助设计》。Springer，2023年。[18] M. S. Rahman、A. Nahiyan、F. Rahman、S. Fazzari、K. Plaks、F. Farahmandi、D. Forte 和 M. Tehranipoor，《针对预言机引导攻击的动态混淆扫描链安全评估》，ACM 电子系统设计自动化汇刊（TODAES），第 26 卷，第 4 期，第 1–27 页，2021年。[19] L. K. Biswas、M. S. M. Khan、N. Varshney、P. Craig、Y. Peng、J. Tang 和 N. Asadizanjani，《为探测先进封装的中介层互连而进行的样品制备》，载于 2025 IEEE 微电子与电子器件研讨会（WMED）。IEEE，2025年，第 1–4 页。[20] G. I. Haidar 等，《GATE-SiP：在系统级封装中实现认证加密测试》，载于第61届 ACM/IEEE 设计自动化会议论文集，系列 DAC '24。美国纽约州纽约市：计算机协会，2024年。 [在线]。可访问：https://doi.org/ 10 1145 3649329 3656527 。/ 。G. I. Haidar 等，“[21] Sect-hi：通过为异构集成启用安全测试来防止系统级封装芯片仿冒品。”[22] M. S. Ul Islam Sami 等，“Pqc-hi：在异构集成中实现基于后量子密码的芯粒认证与密钥交换”，载于2024年 IEEE 第74届电子元件与技术会议（ECTC），2024年，第464–471页。[23] I. Tashdid、T. Farheen 和 S. Rahman，“Safe-sip：使用多方计算的系统级封装安全认证框架”，载于2025年大湖区超大规模集成电路研讨会论文集，丛书 GLSVLSI '25。纽约州纽约市，美国：计算机协会，2025年，第391–396页。 [在线]。可访问：https://doi.org/10. 1145/3716368.3735248[24] M. Nabeel、M. Ashraf、S. Patnaik、V. Soteriou、O. Sinanoglu 和 J. Knechtel，“基于中介层的信任根：抓住机遇，实现不可信芯粒的安全系统级集成”，编号 arXiv:1906.02044，2019年，arXiv:1906.02044 [cs]。 [在线]。可访问：http://arxiv.org/abs/ 1906 02044 。I. Tashdid、T. Farheen 和 S. Rahman，“[25] Authentree：一种面向基于芯粒的异构系统的可扩展、基于多方计算的分布式信任架构”，2025 [ ] 2508 13033 。在[26] 线。可访问：https://arxiv.org/abs/。M. S. Riazi、M. Javaheripi、S. U. Hussain 和 F. Koushanfar，“Mpcir- cuits：面向安全多方计算的优化电路生成”，载于2019年 IEEE 硬件安全与可信国际研讨会（HOST），2019年，第198–207页。[27] E. M. Songhori、S. U. Hussain、A.-R. Sadeghi、T. Schneider 和 F. Koushanfar，“Tinygarble：高度压缩且可扩展的时序混淆电路”，载于2015年 IEEE 安全与隐私研讨会，2015年，第411–428页。[28] J. Zhou、Y. Feng、Z. Wang 和 D. Guo，“使用安全多方计算保护许可型区块链中的隐私”，《传感器》，第21卷，第4期，2021年。 [在线]。可访问：https://www.mdpi.com/ 1424-8220/21/4/1540[29] I. Zhou、F. Tofigh、M. Piccardi、M. Abolhasan、D. Franklin 和 J. Lipman，“用于机器学习的安全多方计算：综述”，《IEEE Access》，第12卷，第53 881–53 899页，2024年。[30] S. Halevi、C. Hazay、A. Polychroniadou 和 M. Venkitasubramaniam，“轮数最优安全多方计算”，《密码学杂志》，第34卷，第3期，第19页，2021年5月。 [在线]。可访问：https: // doi.org/ 10 1007 00145 021 09382 3 。/s - - - J. H. Lau，“芯粒异构集成”，[31] 载于《半导体先进封装》。Springer，2021年，第413–439页。[32] P. Ehrett、T. Austin 和 V. Bertacco，“Sipterposer：用于灵活系统级封装设计的容错基板”，载于2019年欧洲设计、自动化与测试会议及展览（DATE），2019年，第510–515页。[33] A. Tsiougkos 和 V. F. Pavlidis，“带电阻式温度检测器的可重复使用无源中介层热监测设计方法”，《IEEE超大规模集成电路系统汇刊》，第33卷，第7期，第1803–1815页，2025年。[34] A. P. Johnson、S. Patranabis、R. S. Chakraborty 和 D. Mukhopadhyay，“远程动态部分重配置：对物联网和嵌入式安全应用的威胁”，《微处理器与微系统》，第52卷，第131–144页，2017年。[35] R. S. Chakraborty、I. Saha、A. Palchaudhuri 和 G. K. Naik，“通过直接修改 FPGA 配置比特流植入硬件木马”，《IEEE 设计与测试》，第30卷，第2期，第45–54页，2013年。[36] S. Charles 和 P. Mishra，“片上网络安全攻击与对策综述”，《ACM 计算调查》（CSUR），第54卷，第5期，第1–36页，2021年。[37] S. R. Bommana、S. Veeramachaneni、S. Ershad 和 M. Srinivas，“通过深度学习和动态部分重配置缓解 FPGA 上的侧信道攻击”，《科学报告》，第15卷，第1期，第13745页，2025年。[38] Y. Safari、P. Aghanoury、S. S. Iyer、N. Sehatbakhsh 和 B. Vaisband，“基于芯粒系统的混合混淆”，载于2023年第60届 ACM/IEEE 设计自动化会议（DAC），2023年，第1–6页。[39] J. W. Lee、D. Lim、B. Gassend、G. E. Suh、M. Van Dijk 和 S. Devadas，“为集成电路构建用于身份识别和认证应用的密钥的技术”，载于《2004年超大规模集成电路研讨会论文摘要》（IEEE 编号 04CH37525）。IEEE，2004年，第176–179页。

[40] D. Lim、J. W. Lee、B. Gassend、G. E. Suh、M. Van Dijk 和 S. Devadas，“从集成电路中提取密钥”，*IEEE*超大规模集成电路系统汇刊，第13卷，第10期，第1200–1205页，2005年。[41] G. E. Suh 和 S. Devadas，“用于设备认证和密钥生成的物理不可克隆函数”，载于第*44*届年度设计自动化会议论文集，2007年，第9–14页。[42] K. Rosenfeld、E. Gavas 和 R. Karri，“传感器物理不可克隆函数”，载于*2010*年*IEEE*硬件安全与可信国际研讨会（*HOST*）。IEEE，2010年，第112–117页。[43] L. Yu、X. Wang、F. Rahman 和 M. Tehranipoor，“具有签名唯一性增强功能的基于互连的物理不可克隆函数”，*IEEE*超大规模集成电路系统汇刊，第28卷，352，2019年2月，339，无页码。[44] A. Stern、H. Wang、F. Rahman、F. Farahmandi 和 M. Tehranipoor，“Aced-it：在零信任环境中保护机密电子设计免受内部人员威胁”，*IEEE*集成电路与系统计算机辅助设计汇刊，第41卷，第10期，第3202–3215页，2021年。[45] Y. Tamir 和 H.-C. Chi，“用于超大规模集成电路通信交换机的对称交叉开关仲裁器”，*IEEE*并行与分布式系统汇刊，第4卷，第1期，第13–27页，1993年。[46] M. Mazeika、D. Hendrycks、H. Li、X. Xu、S. Hough、A. Zou、A. Rajabi、Q. Yao、Z. Wang、J. Tian、Y. Tang、D. Tang、R. Smirnov、P. Pleskov、N. Benkovich、D. Song、R. Poovendran、B. Li 和 D. Forsyth，“木马检测挑战赛”，载于*NeurIPS 2022*竞赛专题论文集，丛书《机器学习研究论文集》，M. Ciccone、G. Stolovitzky 和 J. Albrecht 编，第220卷。PMLR，2022年11月28日至12月9日，第279–291页。 [在线]。可访问：https://proceedings.mlr.press/v220/mazeika23a.html [47] M. A. Bokor Siddik 和 S. H. Alam，“基于物理不可克隆函数的硬件木马：加密电路的设计与新型攻击”，载于*2023*年电气、计算机与通信工程国际会议（*ECCE*），2023年，第1–5页。[48] N. Wisiol、C. Gr¨abnitz、C. M¨uhl、B. Zengin、T. Soroceanu、N. Pirnay、K. T. Mursi 和 A. Baliuka，“PYPUF：物理不可克隆函数的密码分析”，2021年。 [在线]。可访问：https://doi.org/10 5281 6 3901410 。/zenodo。O. Group，“CVA6 RISC-V CPU”，访问日期：2024年11月。 [在线]。可访问：https://github.com/openhwgroup/cva [49] 2024, 6 [50] N. Corporation，“NVDLA硬件：开源深度学习加速器”，https://github.com/nvdla/hw，2017年，访问日期：2025年7月18日。[51] UltraEmbedded，“RISC-V片上系统”，2024年，访问日期：2024年11月。[在线]。可访问：https://github.com/ultraembedded/riscv soc [52] “CVA6 RISC-V CPU。” 6[在线]。可访问：https://github.com/ openhwgroup/cva[53] 1200 [“OR1200片上系统。”在线]。可访问：https://opencores.org/projects/ or1200 soc