一种面向粗粒度可重构阵列的硬件木马检测算法的设计与实现

严迎建 刘 敏* 邱钊洋
(解放军信息工程大学 郑州 450001)

摘 要：硬件木马检测已成为当前芯片安全领域的研究热点，现有检测算法大多面向ASIC电路和FPGA电路，且依赖于未感染硬件木马的黄金芯片，难以适应于由大规模可重构单元组成的粗粒度可重构阵列电路。因此，该文针对粗粒度可重构密码阵列的结构特点，提出基于分区和多变体逻辑指纹的硬件木马检测算法。该算法将电路划分为多个区域，采用逻辑指纹特征作为区域的标识符，通过在时空两个维度上比较分区的多变体逻辑指纹，实现了无黄金芯片的硬件木马检测和诊断。实验结果表明，所提检测算法对硬件木马检测有较高的检测成功率和较低的误判率。

关键词：硬件木马检测；粗粒度可重构密码阵列；逻辑指纹；多变体
中图分类号：TN406 **文献标识码**：A **文章编号**：1009-5896(2019)05-1257-08
DOI: 10.11999/JEIT180484

Design and Implementation of Hardware Trojan Detection Algorithm for Coarse-grained Reconfigurable Arrays

YAN Yingjian LIU Min QIU Zhaoyang
(*The PLA's Information Engineering University*, *Zhengzhou* 450001, *China*)

Abstract: Hardware Trojan horse detection has become a hot research topic in the field of chip security. Most existing detection algorithms are oriented to ASIC circuits and FPGA circuits, and rely on golden chips that are not infected with hardware Trojan horses, which are difficult to adapt to the coarse-grained reconfigurable array consisting of large-scale reconfigurable cells. Therefore, aiming at the structural characteristics of Coarse-grained reconfigurable cryptographic logical arrays, a hardware Trojan horse detection algorithm based on partitioned and multiple variants logic fingerprints is proposed. The algorithm divides the circuit into multiple regions, adopts the logical fingerprint feature as the identifier of the region, and realizes the hardware Trojan detection and diagnosis without golden chip by comparing the multiple variant logic fingerprints of the regions in both dimensions of space and time. Experimental results show that the proposed detection algorithm has high detection success rate and low misjudgment rate for hardware Trojan detection.

Key words: Hardware Trojan detection; Coarse-grained reconfigurable cryptographic array; Logic fingerprints; Multiple variants

1 引言

由于集成电路产业设计和制造的分离，芯片制造商难以控制芯片制造的所有环节，从而使得原始芯片极有可能被恶意修改或植入恶意逻辑，即硬件木马(Hardware Trojan)[1]，并在特定条件下造成信息泄露、功能错误和拒绝服务等严重危害。硬件木马因其破坏性强，隐蔽性高，对芯片和电子系统造成极大威胁[2,3]。随着基于粗粒度可重构阵列(Coarse-Grained Reconfigurable Array, CGRA)的粗粒度可重构密码阵列(Coarse-Grained Reconfigurable Cryptographic Array, CGRCA)越来越广泛地应用在安全芯片中[4]，研究面向粗粒度可重构密码阵列的硬件木马检测技术极为迫切[5,6]。当前硬件木马检测技术多集中在专用集成电路(Application Specific Integrated Circuit, ASIC)和现场可编程门阵列电路(Field-Programmable Gate Array, FPGA)[7,8]，如文献[9]和文献[10]分别提出了ATMR算法和RLF算法，能实现硬件木马检测，但难以直接移植到阵列电路。现有文献较少涉及面向粗粒度可重构阵列的硬件木马检测，针对阵列电路的文献主要集中在硬件木马实时监控和容错，如文献[5]在系统层次实现硬件木马实时监控，但主要针对STC控制器阵列电路中硬件木马，其阵列结构与粗

收稿日期：2018-05-21；改回日期：2018-09-20；网络出版：2018-10-22
*通信作者： 刘敏 15515671017@163.com

粒度可重构密码阵列差异较大。文献[6]提出一种考虑资源开销的阵列安全映射方案，但该方案主要考虑硬件木马的容错，未侧重于硬件木马的检测和区域定位。

本文面向粗粒度可重构密码阵列，提出基于分区和多变体逻辑指纹(Partitioned Multiple Variant based Logic Fingerprinting, PMVLF)的硬件木马检测算法。该算法充分考虑目标电路本身的可重构特性和冗余特性，基于提出的硬件木马攻击模型和目标电路结构特点对电路进行合理分区，在时空维度比较电路不同分区的多变体逻辑指纹，在无需黄金芯片(不含硬件木马的芯片)的条件下实现了硬件木马检测和定位。

2 问题分析

2.1 CGRCA硬件木马攻击模型

CGRCA由数据网络、控制网络和配置网络共同组成[11]，其中数据网络是CGRCA实现可重构特性和冗余特性的关键部件。数据网络的核心是针对密码运算设计的2D-mesh结构可重构处理单元阵列(Reconfigurable Processing Unit Array, RPUA)，用于完成主要的运算操作。RPUA由大量基本同构的可重构运算单元(Reconfigurable Processing Unit, RPU)组成，单元之间通过crossbar互连结构进行数据通信，在运算过程中运算单元FU和互连单元SB/CB都能动态或部分地实现重构。通过对数据网络不同运算单元及单元间互连网络的重构，CGRCA能够组成特定密码运算所需的结构，完成对数据的加解密操作。其总体架构如图1(a)所示，图1(b)表示多个RPU可以通过级联完成多比特运算，图1(c)为单个RPU的结构。

CGRCA结构规整，数据网络中除了RPUA外的其他部分都是常规模块，可以通过已有技术进行保护[12]，而RPUA作为执行密码运算的主要单元，占据了整个阵列大部分面积，极易成为硬件木马的攻击目标，同时其冗余性和可重构特性又使得硬件木马检测难度较大。因此，本文主要面向数据网络的RPUA进行硬件木马检测，并根据其触发单元特征、有效载荷特征和插入阶段特征总结出面向CGRCA上的硬件木马攻击模型：

(1) 硬件木马触发单元特征：硬件木马触发方式多样，本文主要针对逻辑信号触发型，根据触发信号位置具体可分为RPU相关型和RPU无关型硬件木马[6]，如图2所示。攻击过程中，攻击者可以操纵所有的输入明文比特作为木马触发信号，但控制比特越多，其所需要的硬件开销就会越大，导致被检测到的概率越大，因此出于实际考虑，假设攻击者只会选择所有输入n bit中的k bit ($k \ll n$)作为木马触发信号，表示为$\mathrm{Tr}=2^{n-k}+2^{k}$。

(2) 硬件木马有效载荷特征：硬件木马的有效载荷功能多样，通常分为物理破坏型和逻辑错误型。本文主要针对数据网络由于可重构特性引入的逻辑错误型硬件木马，可表示为$Q'=Q\oplus \mathrm{Tr}=Q\oplus(2^{n-k}+2^{k})$，通过观测最终输出结果或中间值来判断阵列中是否含有此类硬件木马。

(3) 硬件木马插入阶段特征：由于阵列全程自主设计，未插入第三方IP(Intellectual Property)核，因此认为设计过程无木马插入，而流片过程中经过诸多厂商，不能保证制造安全性。假设检测网表即是最后芯片实现的网表，本文主要针对制造阶段非安全环境下的木马插入进行检测。

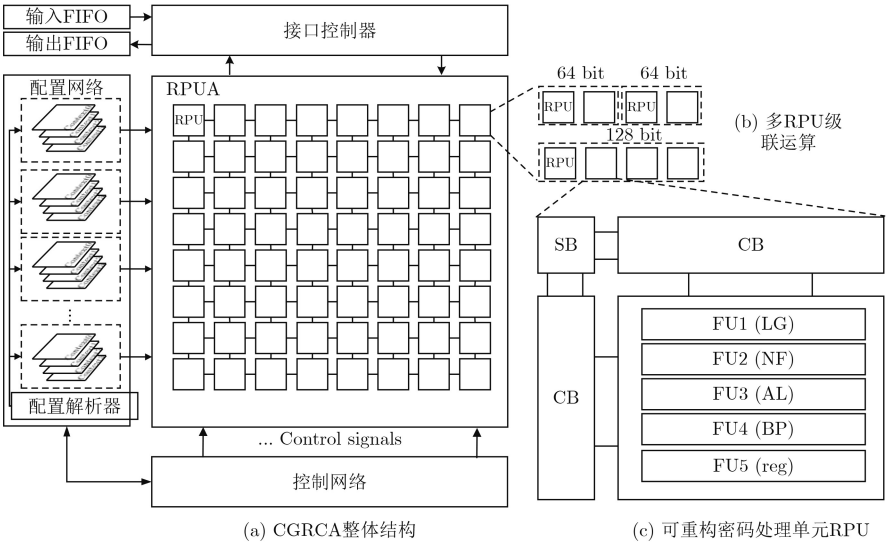
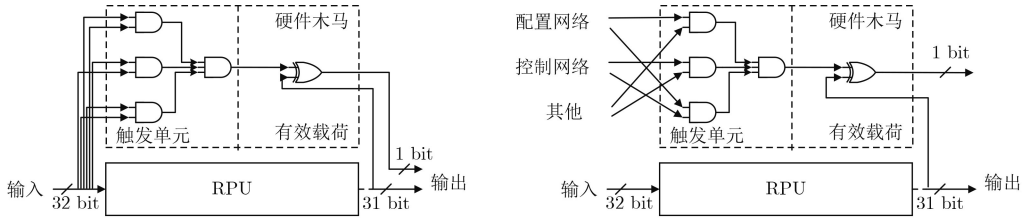


图 1 粗粒度可重构密码阵列硬件结构



(a) RPU相关型硬件木马　　(b) RPU无关型硬件木马

图 2 不同触发信号的硬件木马示意

冗余性和可重构特性使得粗粒度可重构阵列更加灵活高效，但也导致其更易遭到硬件木马攻击。因此，综合以上分析，面向CGRCA进行硬件木马检测时需要考虑以上硬件木马攻击特征，以使硬件木马检测算法更具针对性。

2.2 检测指标

硬件木马检测时可能出现多种情况，如表1所示。通过查阅相关文献，本文采用以下指标评估所提出的检测方法：第一，检测成功率R_s，它表示测试中成功判断电路中是否含有硬件木马的几率，按照检测成功次数N_T与总检测次数的比值N计算，即$R_s = N_T/N$；第二，检测误判率R_f表示为测试中将未含硬件木马的电路误判为感染硬件木马或是将含有硬件木马的电路误判为未感染硬件木马的几率，按照误判次数N_f与总检测次数N的比值计算，即$R_f = N_f/N$。同时，为了评估检测算法的硬件木马容忍能力，借鉴文献[6]中的电路正确输出端口平均比例APCO作为一项检测指标，即正确输出端口占总输出端口的平均比例。

表 1 检测结果可能情况

实际情况	检测情况	
	不含硬件木马	含有硬件木马
不含硬件木马	正确	错误
含有硬件木马	错误	正确

3 硬件木马检测算法设计

3.1 基于分区的多变体逻辑指纹

3.1.1 目标电路分区

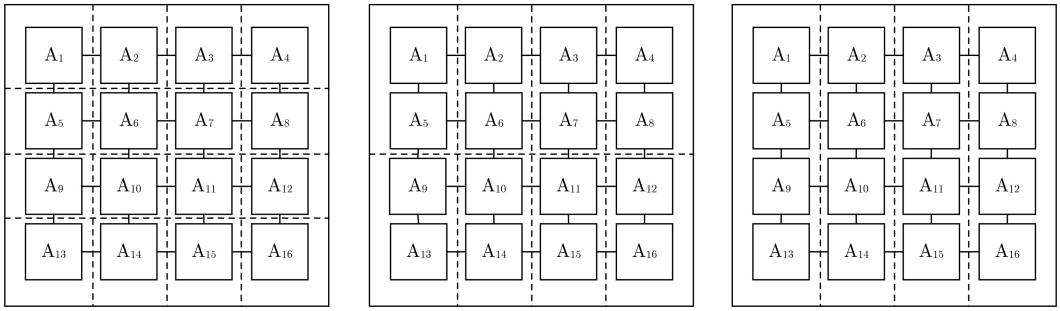
目标电路规模庞大且拥有可重构特性，而硬件木马通常占据整个电路空间的很小一部分，直接进行硬件木马的检测不仅效果不理想，而且耗时较长，因此，本文拟采用“分而治之”的策略将电路进行分区检测。现有硬件木马检测中的分区方法主要采用区域半径(radius)[9]、低翻转节点数量(LTP Nodes)[13]和物理位置(geometric location)[14]等作为分区标准。这些方法采用的基准电路主要是ISCAS标准测试电路，上述基准电路规模相比于CGRCA而言较小，且结构差异较大。现有分区算法难以适用于CGRCA电路，因此必须结合阵列电路自身的结构特征进行区域划分。

CGRCA数据网络结构相对规整，单个RPU可完成相对独立的操作，包含了易于分割的可重构功能单元FU和可重构的互连单元，每个单元数据位宽均为32 bit，且处在同一个时钟域。因此，综合以上分析，考虑分区的区域数量和区域之间连线，将可重构处理单元块RPU作为区域划分的最小单元。在硬件木马检测时，区域面积越小则定位越准确，但是对于分布较为松散的硬件木马而言，区域面积过小可能导致检测成功率下降。为了在提高硬件木马的定位准确度和不降低硬件木马检测成功率上取得折中，针对不同应用场景和需求，考虑将不同数量的RPU组合为一个区域。图3(a), 3(b), 3(c)分别为规模4 × 4的阵列下单个RPU、2个RPU和4个RPU作为一个区域的分区示意。

3.1.2 基于任务多变体的逻辑指纹

逻辑指纹(Logic Fingerprinting, LF)定义为某一区域的输入输出数据组合[9]。如图4所示，区域1输入数据为1001，输出数据为110，则该区域的逻辑指纹为1001110。利用逻辑指纹进行硬件木马检测的原理为，向待测区域持续输入同一测试向量，在时间维度上进行逻辑指纹一致性验证，若不同时刻采样的逻辑指纹一致，则认为不含硬件木马；反之，判定区域感染硬件木马。同时，若逻辑指纹仅为输出变化，可判定其为感染区域；若输入输出同时变化，可判定其为受驱动区域，由此可实现硬件木马的区域定位。

对于功能一定的ASIC电路来说，利用逻辑指纹作为标识检测硬件木马是可行的。但就粗粒度可重构逻辑电路而言，其可重构特性使得电路具体功能和具体配置相关，在不同的配置下同一区域逻辑指纹不同，难以直接进行检测，因此引入多变体技术在空间维度横向比较。硬件木马通常由触发部分(trigger)和有效载荷(payload)两部分组成[15]，而硬件木马攻击者通常会选择稀有条件或序列作为触发条件。文献[10,16]提出，在多核环境和FPGA环境



(a) 单个RPU为一个区域 (b) 2个RPU为一个区域 (c) 4个RPU为一个区域

图 3 CGRCA电路分区示意

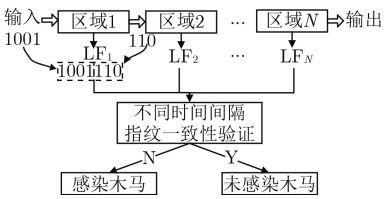
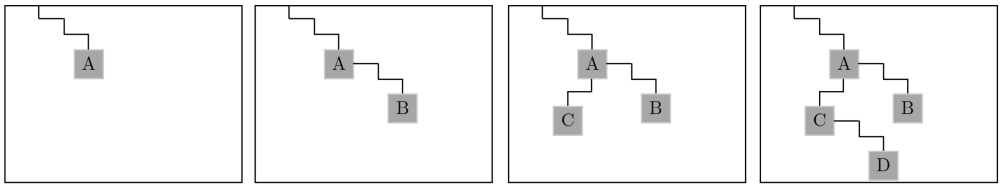


图 4 逻辑指纹检测原理

下，当同一任务采用不同变体(variants)实现时会调用同构单元的不同逻辑资源，从而极少会触发同一个硬件木马，此时，通过比较同一时刻同一任务不同变体的运算结果就可以进行硬件木马检测，如图5所示为多核环境下单变体，双变体，三变体和四变体执行实例。

对于CGRCA而言，其冗余资源允许配置实现同一任务的多个变体，因此在实施硬件木马检测时，可以在不同区域同时配置实现同一任务的多个变体，在空间维度上将不同区域的逻辑指纹进行横向对比，为了全面覆盖阵列电路，对于规模$M \times N$的电路，需同时配置N个变体进行比较，如图6所示。



(a) 单变体 (b) 双变体 (c) 三变体 (d) 四变体

图 5 子任务不同变体执行实例

综上，对于面向粗粒度可重构阵列的硬件木马，采用多变体配置，在时空维度上进行逻辑指纹对比，可实现无需黄金芯片对照的硬件木马检测和区域定位，即使硬件木马在初始时刻触发，也能通过不同任务变体的结果对比检测出来。

3.2 PMVLF算法设计

基于以上分析，本文提出基于分区和多变体逻辑指纹(Partitioned Multiple Variant based Logic Fingerprinting, PMVLF)的硬件木马检测算法。首先，将目标电路划分为多个区域，然后，采用逻辑指纹特征作为电路的标识符，在时空两个维度上比较分区的多变体逻辑指纹，作为硬件木马检测和诊断的依据。与前人工作[9]相比，该方法充分考虑目标电路冗余特性和可重构特性及面向粗粒度可重构阵列电路的硬件木马攻击特点，采用的分区策略更加合理，并同时在时间维度和空间维度进行多次对比，实现了无需黄金芯片的大规模电路硬件木马检测和定位，算法流程如图7所示。

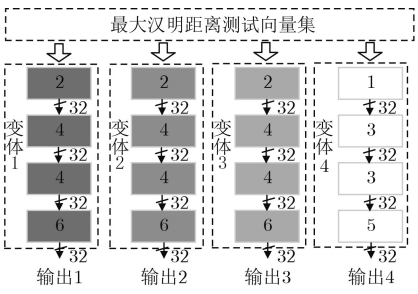


图 6 不同区域对比示意

该算法输入为待测电路网表文件、任务多变体配置库和测试向量集，采用多配置更新以适应目标电路可重构特性，采用多变体逻辑指纹时空比较以适应目标电路冗余特性。对于单个配置，首先，将电路分区并在同一行配置不同变体，依次施加不同持续时长的最大汉明距离测试向量[9]，然后在相同的时间间隔内采样输入\输出值，提取逻辑指纹，最后在时空维度上比较逻辑指纹。时空维度的逻辑指纹比较包括指纹自较和指纹互较，首先，在时间

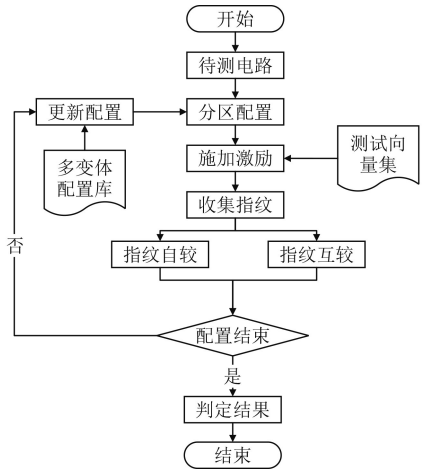


图 7 算法检测流程示意图

维度上进行指纹自较，若配置和激励不变的条件下，输出逻辑指纹前后变化，则可判定区域感染硬件木马；然后，在空间维度上进行指纹互较，若配置和激励相同的条件下，同一任务不同变体的区域逻辑指纹不同，则可根据择多原则[10]判定可疑区域。

单个配置结束后，进行配置更新，若遍历配置库和测试向量集后所有区域均被判为可信，则认为该待测电路不含硬件木马，否则输出可疑硬件木马感染区域。具体步骤如下：

步骤 1　获得待测电路网表CUT，将待测电路网表分为$M \times N$个区域，其中第i行第j列的分区记为$D_\mathrm{CUT}_{(i,j)}$；获取测试向量集Testpattern，设置多变体配置库C为$l \times N$种配置，其中C_i表示配置库C中的第i个任务的N个变体，测试向量数目为q；

步骤 2　对待测电路网表CUT中的每一行随机选取第i组配置C_i，将其导入该行N个分区。对第1行不同分区同时循环施加相同测试向量$\mathrm{Testpattern}_m$至持续时间T_m，每隔Δt时间收集1次各区域$D_\mathrm{CUT}_{(i,j)}$对应逻辑指纹$\mathrm{LF}_{(i,j)}$；

步骤 3　对比不同时刻同一区域逻辑指纹$\mathrm{LF}_{(i,j)}$，若逻辑指纹变化，则判断该区域为可疑区域；对比相同时刻每一行不同区域逻辑指纹，若逻辑指纹不同，则根据择多原则，将结果为少数的区域判断为可疑区域；

步骤 4　选取可疑区域，分析区域逻辑指纹变化情况，若逻辑指纹仅为输出变化，判断为硬件木马感染区域；若逻辑指纹输入输出均变化，则判断为硬件木马受驱动区域，并向其输入正确变体输入，跳转回步骤3进行判断；判断是否遍历配置库和测试向量集，若满足条件则继续步骤5；否则跳转回步骤2；

步骤 5　若待测电路所有区域均为可信区域，判断当前待测电路为可信电路；否则当前待测电路中的输出可疑区域。算法至此结束。

3.3 算法评估

硬件木马种类多样，其物理分布可能是紧凑型也可能是松散型。文献[14]中采用所有区域激活的组合来检测松散型硬件木马，则对于硬件木马最大触发信号为k位，$M \times N$个区域的电路而言，则共有$\sum_{i=1}^{i=k} C_{M\times N}^{i}$种组合需要覆盖，假设有$l \times N$种配置，$q$条测试向量，则其算法计算复杂度为$O\left(l^M q \sum_{i=1}^{i=k} C_{M\times N}^{i}\right)$。而在本文中，直接对每个单独的区域进行检测，无需检测区域的组合，最多检测$M \times N$次，故本算法的计算复杂度仅为$O(l^M qMN)$，因此计算开销大大减少。同时该算法充分考虑了可重构电路本身的冗余特性，避免引入多余检测逻辑造成额外的资源开销。

4 仿真实验

本文基于VCS仿真平台在原始电路上模拟插入了两种硬件木马，并实现了电路网表分区，选择多个任务的不同变体用于RPU映射实现，最后基于PMVLF算法实现面向阵列电路的硬件木马检测。

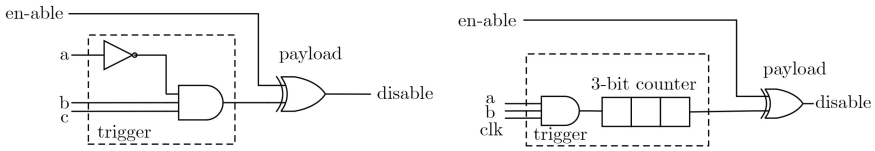
4.1 实验设置

为了评估上述算法的有效性，本文以规模4×4的RPUA(不含S盒和密钥寄存器)为目标电路，分别设计了两种硬件木马，包括一种组合触发功能修改型(C型)和一种时序触发功能修改型(S型)硬件木马，并分别以不同数目随机插入到原始电路数据网络中，硬件木马电路设计示意如图8(a)，图8(b)。同时，挑选典型密码算法AES算法、SMS4算法、A5/1算法在阵列上映射实现作为基准电路，由于单个区域难以实现完整的算法功能，故拆分典型算法的子步骤实现。

实验中以硬件木马类型和数目标识待测电路，如C-10电路表示插入10个C型硬件木马的电路，具体待测电路的属性如表2。电路中单个RPU单元可分为3个模块，包括最小可重构单元FU，电路的输入输出寄存网络和互连网络等，由于硬件木马规模相比原始模块较小，因此实验中不作为单独模块。线网数即为模块之间连线数目。本文主要针对数据网络硬件木马，且硬件木马均为紧凑型分布，所有配置网络和控制网络均为缺省配置。

4.2 实验结果与分析

为了对本文的设计方案进行验证，将规模为4×4的阵列原始电路在CMOS 55 nm工艺下综合得到原始网表文件，并按照4.1节的硬件木马设计插入网表级硬件木马实施检测。为了对检测方法进



(a) 组合触发功能修改型(C型) (b) 时序触发功能修改型(S型)

图 8 硬件木马设计示意图

表 2 待测电路具体情况

电路编号	分区数	线网数	输入	输出	木马面积占比(%)
原始电路	16	260	49	44	/
C-10	16	260	49	44	0.015
S-10	16	260	49	44	0.025
CS-5/5	16	260	49	44	0.031

行评估，将本文算法与ATMR算法[10]、DRMaSV算法[6]、RLF算法[9]的结果进行比较，将检测成功率R_s、检测误判率R_f、正确输出比例APCO[6]作为指标在待测电路和原始电路上进行多次仿真实验，表3列举了在待测电路上的对比结果。

在无硬件木马插入的原始电路上测试时，所有方法均未出现误判。在插入相应硬件木马的待测电路上测试时，从表3数据可看出，本文算法在时序木马检测、组合木马检测和复合木马检测中得到的成功检测率、误判率结果均优于其他算法，表现了本文方法的有效性。图9(a)，图9(b)，图9(c)，图9(d)分别是不同检测算法在各待测电路上平均检测成功率、平均检测误判率、平均检测正确输出比例及检测指标综合对比(为对比清晰，图9(d)中平均检测误判率的值乘以100)。

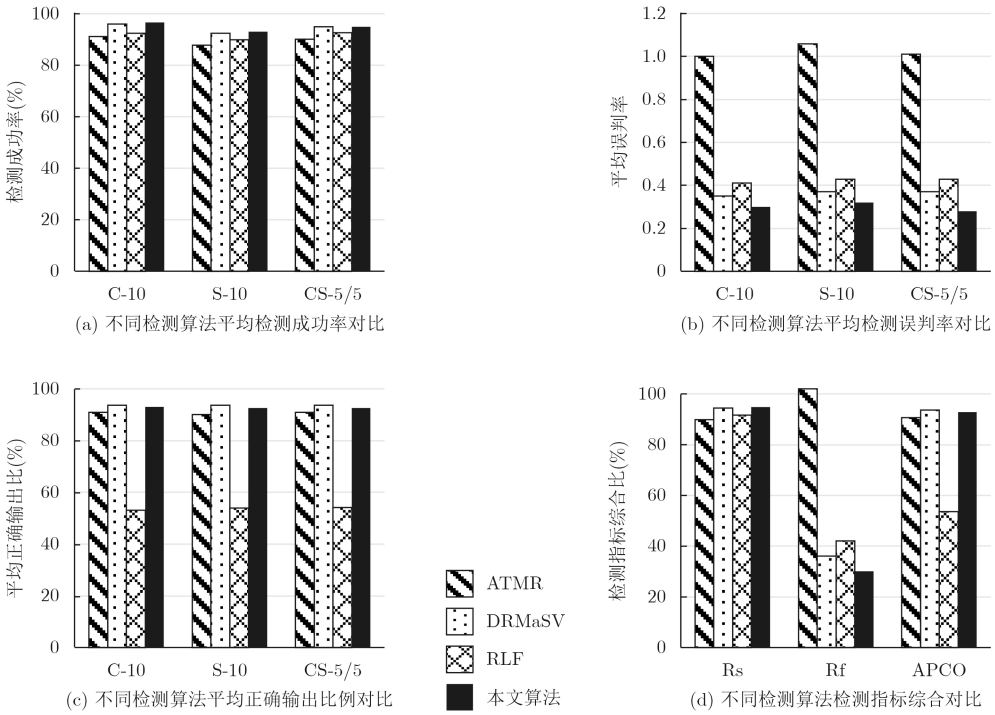
从图9(a)可以看出，本文算法上在通用指标平均检测成功率上略优于ATMR算法和RLF算法，从图9(b)，图9(c)，图9(d)可以看出，本文算法在平均检测误判率上明显低于ATMR算法，在平均正确输出比例上明显优于RLF算法。在所选的3项指标上，本文算法与DRMaSV算法性能比较接近，但本文算法在实现木马容忍的同时可以实现硬件木马的区域定位。综合来看，本文算法相比另外3种算法均有明显优势。分区思想的引入使得本文算法可实现硬件木马的定位，采用逻辑指纹作为区域的标识为硬件木马检测提供了简单而精确的衡量指标，多变体技术消除了逻辑指纹难以检测初始时刻触发的硬件木马的技术缺陷，同时使得算法具有较好的木马容忍能力，增加了算法的鲁棒性和灵活性。

5 结论

随着芯片制造全球化，外包广泛发展，生产商难以控制芯片制造过程中所有流程，使得硬件木马成为影响集成电路硬件安全潜在的巨大威胁，而随着粗粒度可重构阵列在各类芯片中的广泛应用，面向可重构阵列电路的硬件木马检测十分必要。本文以粗粒度可重构密码阵列平台为研究对象，通过对阵列电路进行分区，利用逻辑指纹构建电路的唯一

表 3 检测算法实验结果对比(%)

电路编号		ATMR			DRMaSV			RLF			本文算法		
		R_s	R_f	APCO	R_s	R_f	APCO	R_s	R_f	APCO	R_s	R_f	APCO
C-10	AES	91.1	1.02	91.3	97.2	0.37	93.7	93.1	0.45	52.1	97.3	0.31	93.2
	SMS4	92.9	0.98	91.7	97.3	0.35	93.8	93.2	0.41	54.1	97.7	0.29	93.7
	A5/1	89.7	0.99	89.9	93.5	0.32	93.9	91.1	0.37	52.7	94.7	0.30	92.1
	均值	91.2	1.00	91.0	96.0	0.35	93.8	92.5	0.41	53.0	96.6	0.30	93.0
S-10	AES	89.7	1.13	90.2	93.2	0.39	93.5	89.6	0.47	53.7	93.4	0.32	93.3
	SMS4	88.5	1.05	90.3	93.4	0.38	93.9	90.1	0.44	53.9	93.3	0.31	92.9
	A5/1	85.3	1.01	89.7	90.5	0.33	93.4	90.2	0.39	54.1	92.7	0.33	91.7
	均值	87.8	1.06	90.1	92.4	0.37	93.6	90.0	0.43	53.9	93.1	0.32	92.6
CS-5/5	AES	90.1	1.00	91.0	95.3	0.39	93.8	93.0	0.46	54.6	95.4	0.27	92.1
	SMS4	91.3	0.99	91.3	95.6	0.37	94.0	92.7	0.43	54.7	95.6	0.27	93.2
	A5/1	89.3	1.01	90.7	93.7	0.35	93.5	92.3	0.39	53.3	93.7	0.31	92.7
	均值	90.2	1.00	91.0	94.9	0.37	93.8	92.7	0.43	54.2	94.9	0.28	92.7
均值		89.8	1.02	90.7	94.4	0.36	93.7	91.7	0.42	53.7	94.9	0.30	92.8



(a) 不同检测算法平均检测成功率对比　(b) 不同检测算法平均检测误判率对比

(c) 不同检测算法平均正确输出比例对比　(d) 不同检测算法检测指标综合对比

图 9 不同检测算法结果对比

标识符，并利用可重构特性配置多变体，将不同变体的结果在时空维度上进行对比，实现了无需黄金芯片条件下的硬件木马检测，算法复杂度较低，灵活性好，适合工程实践，对ASIC和FPGA电路硬件木马检测与定位问题也有一定借鉴意义。

参 考 文 献

[1] AGRAWAL D, BAKTIR S, KARAKOYUNLU D, *et al*. Trojan detection using IC fingerprinting[C]. IEEE Symposium on Security and Privacy. IEEE Computer Society, Berkeley, USA, 2007: 296–310. doi: 10.1109/SP.2007.36.

[2] KITSOS P, SIMOS D E, TORRES-Jimenez J, *et al*. Exciting FPGA cryptographic trojans using combinatorial testing[C]. IEEE International Symposium on Software Reliability Engineering, Gaithersbury, USA, 2016: 69–76. doi: 10.1109/ISSRE.2015.7381800.

[3] 赵剑锋, 史岗. 硬件木马研究动态综述[J]. 信息安全学报, 2017, 2(1): 74–90. doi: 10.19363/j.cnki.cn10-1380/tn.2017.01.006.

ZHAO Jianfeng and SHI Gang. A survey on the studies of hardware trojan[J]. *Journal of Cyber Security*, 2017, 2(1): 74–90. doi: 10.19363/j.cnki.cn10-1380/tn.2017.01.006.

[4] COMPTON K and HAUCK S. Reconfigurable computing: A survey of systems and software[J]. *ACM Computing Surveys*, 2002, 34(2): 171–210. doi: 10.1145/508352.508353v.

[5] VEERANNA N and SCHAFER B C. Hardware trojan avoidance and detection for dynamically re-configurable FPGAs[C]. International IEEE Conference on Field-Programmable Technology. Xi'an, China, 2017: 193–196. doi: 10.1109/FPT.2016.7929531.

[6] LIU Leibo, ZHOU Zhuoquan, WEI Shaojun, *et al*. DRMaSV: Enhanced capability against hardware trojans in coarse grained reconfigurable architectures[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017, 37(4): 782–795. doi: 10.1109/TCAD.2017.2729340.

[7] KHALEGHI B, AHARI A, ASADI H, *et al*. FPGA-based protection scheme against hardware trojan horse insertion using dummy logic[J]. *IEEE Embedded Systems Letters*, 2015, 7(2): 46–50. doi: 10.1109/LES.2015.2406791.

[8] PIRPILIDIS F, STEFANIDIS K G, VOYIATZIS A G, *et al*. On the effects of ring oscillator length and hardware Trojan size on an FPGA-based implementation of AES[J]. *Microprocessors & Microsystems*, 2017, 54(1): 75–82. doi: 10.1016/j.micpro.2017.09.001.

[9] SARAN T, RANJANI R S, DEVI M N, *et al*. A region based fingerprinting for hardware Trojan detection and diagnosis[C]. International Conference on Signal Processing and Integrated Networks. Noida, India. 2017: 166–172. doi: 10.1109/SPIN.2017.8049937.

[10] MAL-SARKAR S, KARAM R, NARASIMHAN S, *et al*. Design and validation for FPGA trust under hardware trojan attacks[J]. *IEEE Transactions on Multi-Scale Computing Systems*, 2017, 2(3): 186–198. doi: 10.1109/TMSCS.2016.2584052.

[11] 陈韬, 罗兴国, 李校南, 等. 一种基于流处理框架的可重构分簇式分组密码处理结构模型[J]. 电子与信息学报, 2014, 36(12): 3027–3034. doi: 10.3724/SP.J.1146.2014.00023.
CHEN Tao, LUO Xingguo, LI Xiaonan, *et al*. An architecture of stream based reconfigurable clustered block cipher processing array[J]. *Journal of Electronics & Information Technology*, 2014, 36(12): 3027–3034. doi: 10.3724/SP.J.1146.2014.00023.

[12] WAKSMAN A and SETHUMADHAVAN S. Silencing hardware backdoors[C]. IEEE Security and Privacy. Berkeley, USA, 2011: 49–63. doi: 10.1109/SP.2011.27.

[13] SASHANK K A, REDDY H S, PAVITHRAN P, *et al*. Hardware trojan detection using effective test patterns and selective segmentation[C]. International Symposium on Security in Computing and Communication. Singapore, 2017: 379–386. doi: 10.1007/978-981-10-6898-0_31.

[14] SALMANI H, TEHRANIPOOR M, and PLUSQUELLIC J. A layout-aware approach for improving localized switching to detect hardware trojans in integrated circuits[C]. IEEE International Workshop on Information Forensics and Security, Seattle, USA, 2011: 1–6. doi: 10.1109/WIFS.2010.5711438.

[15] XIAO Kan, FORTE D, JIN Yier, *et al*. Hardware trojans: lessons learned after one decade of research[J]. *ACM Transactions on Design Automation of Electronic Systems*, 2016, 22(1): 1–23. doi: 10.1145/2906147.

[16] MCINTYRE D, WOLFF F, PAPACHRISTOU C, *et al*. Trustworthy computing in a multi-core system using distributed scheduling[C]. IEEE On-Line Testing Symposium. Corfu, Greece, 2010: 211–213.

严迎建：男，1973 年生，教授，研究方向为安全专用芯片设计技术.

刘 敏：女，1995年生，硕士生，研究方向为安全专用芯片设计技术硬件木马检测.

邱钊洋：男，1991年生，博士生，研究方向为信号分析与软件无线电.