论 文

基于混沌的公开可验证 FPGA 知识产权核水印检测方案

张吉良^①, 林亚平^{①*}, 吕勇强^②, 王湘奇^③^① 湖南大学信息科学与工程学院, 长沙 410082^② 清华大学信息技术研究院, 清华信息技术国家实验室, 北京 100084^③ 湖南师范大学数学与计算机科学学院, 长沙 410006

* 通信作者. E-mail: yplin@hnu.edu.cn

收稿日期: 2012-10-25; 接受日期: 2013-07-12

国家自然科学基金项目 (批准号: 61173038, 61228204)、“核高基”国家科技重大专项(批准号: 2013ZX01039001-002-003) 和湖南省研究生创新项目 (批准号: CX2012B142) 资助

摘要 现有的 FPGA 知识产权核 (intellectual property, IP) 水印技术在公开验证时可能会泄漏敏感信息, 使得恶意的验证者或者第三方很容易将水印从 IP 中移除然后重新出售. 零知识 FPGA IP 水印检测虽能有效的解决敏感信息泄露问题, 但易遭受嵌入攻击, 使得公开验证时无法防止非诚实 IP 购买者 (验证者) 抵赖侵权. 本文提出一种新的基于混沌的公开可验证 IP 水印检测方案, 不仅能防止敏感信息泄漏, 而且能抵抗嵌入攻击, 防止证明者、验证者或者可信第三方的欺骗. 传统的 FPGA IP 水印技术, 水印隐藏在未使用的 Slice 中, 因此资源开销与水印嵌入数量成正比. 而本文提出的方案中, 水印隐藏在已经使用的 Slice 的未用的查找表 (lookup table, LUT) 中, 资源和时延开销为 0; 此外, 混沌系统具有良好的随机统计特性且对初值敏感、易于产生数量众多的互相关性极低的伪随机数序列, 混沌系统的这些优点刚好满足方案中对 FPGA 位流文件的 LUT 随机位置置换的特殊要求, 使得位置置换具有极高鲁棒性; 最后, 引入时间戳机制来抵抗嵌入攻击以防止非诚实的 IP 购买者抵赖. 实验结果和分析表明本文提出的方案在水印开销和位置置换鲁棒性方面均明显优于现有的方案.

关键词 水印 现场可编程门阵列 知识产权核保护 混沌 公开可验证 零知识 时间戳

1 引言

随着可重用设计方法学在 IC 设计领域的盛行, 集成电路设计工业面临着越来越严重的知识产权侵权问题. 模块化设计的知识产权核 (IP) 不需要逆向工程就很容易被第三方复制或者出售, 这样会导致 IP 生产商的经济损失, 同时降低其产品的市场占有率. 如何有效地防止集成电路知识产权侵权是 FPGA 厂商和 IC 设计者面临的一个严重问题.

FPGA IP 保护技术主要有: 1) 基于加密的方案; 2) 基于数字签名的方案. 基于加密的方案是对配置比特流进行加密然后将其载入现场可编程门阵列 (FPGA)^[1]. Xilinx 平台 (Virtex-II and Virtex-II Pro FGAs) 支持这种技术. 然而对于基于静态储存器的 FPGA, FPGA 设计以比特流的形式保存在外部存储器 (例如 EEPROM). FPGA 启动时, 加载比特流来配置 FPGA. 在加载的过程中, 通过 wiretap

引用格式: 张吉良, 林亚平, 吕勇强, 等. 基于混沌的公开可验证 FPGA 知识产权核水印检测方案. 中国科学: 信息科学, 2013, 43: 1096–1110, doi: 10.1360/112012-466

的方式很容易对比特流进行克隆^[2]. 因此, 这项技术很难阻止 IP 被转卖给未授权的用户. 基于数字签名的方案是通过在 IP 内嵌入一个加密的签名即水印或指纹来代表 IP 所有权. 当 IP 涉嫌侵权时, 其所有者可以申请一个可信第三方来从被盗用的 IP 中恢复签名, 以证明其版权. 水印是一种应用于集成电路知识产权保护的新兴技术, 近几年来已经在学术界和工业界得到广泛的关注和研究^[3~9].

然而为了成功的检测 IP 侵权, 仅仅研究如何嵌入鲁棒性水印以及如何有效的检测出嵌入的水印是远远不够的. 现有的 FPGA 水印检测方案通常可由一个指定的验证组 (designated verification team) 来验证水印, 这样验证很安全, 但是在实际的验证过程中要保证验证组中所有成员都可信是很难的, 因此由这样的验证组公布的验证结果可能并不能让 IP 购买者或 IP 所有者觉得可信. 相应的, 不需要指定验证组参与的公开验证中, 如果验证者自己能跟踪 IP 中的水印则验证结果是可信的. 然而在公开验证的时候, 证明者需要向验证者提供敏感信息 (如水印内容和水印嵌入的位置等), 一旦这些敏感信息泄露, 恶意的攻击者很容易把水印从 IP 中移除然后重新出售. 这对于水印技术应用的安全性是一个严重的威胁. 此外, FPGA IP 本质上是一个位流文件 (bit-file core), 该文件中嵌入的水印相对于 ASIC 更容易被篡改和覆盖. 因此, 公开可验证是 FPGA IP 水印领域面临的重要问题, 也是阻碍其应用的主要障碍之一, 然而目前国内外处理该问题的研究很少.

2 相关工作

为了防止敏感信息的泄露, 文献 [5] 提出将多个小的水印代替一个大的水印来增强水印的鲁棒性, 但是该方法在公开验证之后会泄露水印位置集合的一部分, 多次侵权发生后, 更多的水印位置会被泄露, 这样攻击者可移除的水印数会更多. 之后文献 [6,7] 提出了公开可检测 (publicly detectable) 的 VLSI 水印技术, 它嵌入一个独立的公开水印用于公开验证. 但该方法不适合 FPGA, 因为对 FPGA 知识产权核来说, 水印被嵌入位流文件, 在公开验证的时候, 公共水印的位置会泄露, 攻击者很容易篡改、移除或覆盖嵌入位流文件中的公共水印, 从而导致知识产权的错误认证. 在文献 [8] 中, 作者提出了一种零开销的水印技术且能有效的抵抗移除攻击, 但是该方法和 [5,9] 一样在公开验证之后, 水印内容和水印嵌入的位置会被泄露, 这对水印技术的应用构成了严重的威胁. 零知识水印检测利用密码学中零知识和知识证明的思想和算法, 实现在水印检测时使得验证者确信水印存在性的同时又不泄漏相关敏感信息, 已被广泛应用于很多领域^[10,11]. 近来, 文献 [12] 提出了一种基于零知识协议的 FPGA 数字签名验证方案, 克服了文献 [5,9] 在公开验证过程中泄露有关数字签名敏感信息的安全问题, 但文献 [12] 提出的方案易遭受下面的攻击: 非诚实的 IP 购买者使用相同的水印嵌入算法在 IP 中非法嵌入自己的水印而不影响 IP 的功能. 这种攻击会导致文献 [12] 的零知识协议的 Step 3a 的验证永远都是 false, 即 $D'_M \neq M'_C(D)$. 又由于 IP 中已嵌入非诚实的 IP 购买者的水印信息, 因此即使在 Step 3b 验证成功的时候, 非诚实的 IP 购买者也能宣称所使用的 IP 属于自己, 然后提取出非法嵌入的水印以抵赖侵权. 我们称这种攻击为嵌入攻击. 公开可验证性的目的是降低或者消除协议中某一方对协议其他参与方的可信度的依赖, 减少对协议的验证者身份的限制, 从而提高整个方案的安全性, 并且当协议参与各方出现争端时, 方便争端的仲裁^[13]. 由于文献 [12] 提出的零知识方案依赖于 IP 购买者 (验证者) 可信 (否则易受嵌入攻击), 因此实际上不具备真正的公开可验证性.

本文提出一种新的基于混沌序列的公开可验证 FPGA IP 水印检测方案以解决 FPGA 水印技术在公开验证时存在的敏感信息泄露和现有 FPGA 零知识水印检测方案易受嵌入攻击的问题. 在本文提出的方案中: 1) 水印隐藏在已经使用的 Slice 的未用的查找表 (lookup table, LUT) 中, 通过编码水

图 1 Virtex-II 逻辑块的结构

Figure 1 Virtex-II logic block architecture

印后的 FPGA 设计 $\hat{I}$ 的 LUT 内容可防止水印内容的泄漏; 2) 利用混沌序列随机性好和混沌序列之间互相关性很弱的特性, 在每轮验证中产生一个实数范围内的混沌序列, 对实值混沌序列进行二值化处理, 二值化后的混沌序列 ρ 作为本文的位置映射算法 $\pi(\rho)$ 的输入用于控制 FPGA 比特流文件的 LUT 的位置置换即 $\pi(\rho)$ 应用于 $\hat{I}$ 得到乱置的设计 $\mathcal{L}$, 然后 $\mathcal{L}$ 中水印的位置和 ρ 用于证明者和验证者之间交互, 证明者通过零知识协议使得验证者相信水印存在 IP 中而不泄露水印的位置; 3) 引入时间戳机制^[14]来抵抗嵌入攻击以防止非诚实 IP 购买者(验证者)抵赖. 因此与现有的零知识 FPGA 水印检测方案[12]相比, 本文提出的验证方案不仅证明者能够证明 IP 中确实存在水印而不泄露水印内容和水印位置, 而且任何验证者(包括不可信的验证者)能够验证水印的合法性、有效的抵抗嵌入攻击, 从而防止证明者、验证者或者可信第三方的欺骗, 因此称为公开可验证的零知识 FPGA IP 水印检测方案. 从文献[12]可知资源开销和零知识协议的随机置换鲁棒性是衡量水印检测方案的两个主要性能指标. 实验结果显示本文提出的方案的资源开销为 0, 随机置换算法具有极高的随机置换鲁棒性.

3 FPGA IP 水印方案

设 I 为需要水印的 FPGA 位流文件 IP 核. FPGA IP 水印方案可定义为一个三元组: $\Phi = (G(W, K, S), E(\hat{I}, I, W, R), D(\hat{I}, W, R))$, 其中:

- $G(W, K, S)$ 为水印生成算法. S 为 IP 所有者的签名, 采用公钥加密算法(K 作为密钥)加密 S 以生成水印 W . 如果加密后的签名信息过长, 则采用单向哈希函数生成固定长度的摘要信息.

- $E(\hat{I}, I, W, R)$ 为水印嵌入算法. I, W 和 R (水印嵌入的位置) 为水印嵌入算法的输入, 水印后的 FPGA 设计为 $\hat{I}$.

- $D(\hat{I}, W, R)$ 为水印检测算法. 水印检测算法 D 的输出为布尔值, 1 表示水印存在 $\hat{I}$ 中, 0 则表示水印不存在 $\hat{I}$ 中.

下面首先介绍 FPGA 水印原理, 然后分别介绍本文所提出的水印生成、嵌入和检测方案.

3.1 FPGA 水印原理

图 1(a) 描述了一个 Xilinx Virtex-II 逻辑块, 称为 CLB. 在 Xilinx Virtex-II FPGA 中一个 CLB 单元包含 4 个 Slice. 从图 1(b) 中可以看到, 一个 Slice 包含两个触发器, 两个四输入 LUT (一个 G-LUT 和一个 F-LUT) 以及其他运算电路(未显示在图 1(b) 中). CLB 在 FPGA 上排列成一个二维阵列, 并且可以通过可编程互连矩阵相互连接.

Slice 实现的逻辑功能在布局和布线阶段之后分为三类: 1) 组合逻辑和时序逻辑 (combinational with register): 这些 Slice 用来实现组合逻辑和时序逻辑. 2) 纯时序逻辑 (register only): 这些 Slice 仅用来实现时序逻辑. 3) 纯组合逻辑 (combinational with no register): 这些 Slice 仅用于实现组合逻辑. 而在这三种已被使用的 Slice 中包含了大量的未使用的 LUT, 我们将这些 LUT 作为水印的载体, 为方便起见, 我们称之为 ILUT^[8]. 四输入的 ILUT 是较小的存储单元, 它能实现任意四输入的布尔函数. 特别地, 每个 LUT 中包含的 SRAM 单元保存了由该 LUT 实现的逻辑功能的真值表. 根据该特性, 一个四输入的 ILUT 可存储 16 bits 的数字水印. 例如, 一个 4-bit 水印 $w = "1001"$, 可用布尔函数 $F(A_1, A_2, A_3, A_4) = \sim(A_1 \text{ XOR } A_2)$ 来实现, 该布尔函数可被配置成一个 ILUT. 在 FPGA 设计中, ILUT 存在的原因与布局和布线阶段有关的, 通过让一些 LUT 空闲 (未被使用) 来提高电路面积设计的效率, 这样可更加有效地利用 FPGA 的域资源和互连资源^[15].

3.2 水印生成和嵌入

本文中水印生成和嵌入过程描述如下:

Step1: 水印生成. 首先签名 S 用加密算法进行加密, 然后用单向 Hash 函数 (如 SHA-2) 处理得到固定长度的摘要信息 $\hat{S}$, 最后用经过散列处理的混沌序列^[16] (其混沌初值作为密钥 K_1) 置乱 $\hat{S}$ 得到水印 W .

Step2: 定位水印嵌入的位置. 用伪随机数生成器 (如 RC4, 混沌等, K_2 作为密钥), 生成一个伪随机序列作为水印嵌入的位置.

Step3: 水印嵌入. 将水印以 LUT 所能容纳水印的最大值分组, 然后嵌入 ILUT 中^[8].

Step4: 将水印的 ILUTs 输入和输出与原始电路的 “don’t care” 输入连接来伪装嵌入的水印.

相关的 FPGA 水印嵌入方法可参考文献 [5,8,9,12]. 文献 [5,9,12] 的水印载体均为未使用的 Slice, Slice 开销与水印量成正比. 本文水印思想的水印载体为已使用的 Slice 中的未用的 LUT, 因此 Slice 开销为 0, 在开销上明显优于文献 [5,9,12].

3.3 基于混沌的水印验证协议

3.3.1 协议概述

零知识公开可验证的目的是证明 IP 所有者的签名存在 FPGA 的比特流文件中而不泄露水印的内容和水印的位置. 我们称证明者叫 Alice, 验证者叫 Bob, Alice 根据 3.2 小节所述的水印生成和嵌入算法首先根据 S 生成 W , 然后将 W 嵌入 FPGA 设计 I 中得到一个水印的 FPGA 设计 $\hat{I}$. Alice 现在想要证明在 $\hat{I}$ 中存在 W 而不泄露 W 的位置.

零知识水印验证协议步骤如下:

Step 1: Alice 生成一个随机的实值混沌序列 x (混沌初值作为密钥), 将 x 二值化处理后得二值混沌序列 ρ , ρ 用于位置置换算法 $\pi(\rho)$, 将 $\pi(\rho)$ 应用于 $\hat{I}$ 进行随机位置置换即计算 $\mathcal{L} = \pi(\rho) \hat{I}$, 然后将 $\mathcal{L}$ 发送给 Bob.

Step 2: Bob 知道 $\hat{I}$, $\mathcal{L}$, 然后抛硬币 i 让 Alice 执行 Step 2a 或者 Step 2b.

Step 2a: 公开 ρ .

Step 2b: 公开 $\mathcal{L}$ 中的水印位置.

Step 3: 若 $i = 0$, Alice 执行 Step 2a; 若 $i = 1$, Alice 执行 Step 2b.

图 2 用于证明者 Alice 和验证者 Bob 之间交互的零知识验证协议

Figure 2 The zero-knowledge verification protocol used for interaction between prover Alice and verifier Bob

Step 4: Bob 根据 Step 2a 或 Step 2b 相应地执行 Step 4a 或 Step 4b.

Step 4a: 确认 $\mathcal{L}$ 是 $\hat{I}$ 的一个合法置换即 $\mathcal{L} = \pi(\rho)\hat{I}$.

Step 4b: 验证 W 存在于 $\mathcal{L}$ 中.

Step 5: 混沌序列 ρ 被丢弃.

Step 1 ~ Step 5 执行 R 轮, 每轮验证都需要生成一个新的混沌序列用于随机位置置换.

根据 Alice 能成功欺骗 Bob 的概率, Step 1 ~ Step 5 需迭代执行多次, 如果都通过, 则验证者接受. 此外, 我们通过编码 $\hat{I}$ 的 LUT 内容可防止水印内容的泄漏. 研究如何安全有效的编码并非本文的重点, 现有的成熟的编码方法有很多种, 如线性反馈移位寄存器^[17]、非线性反馈移位寄存器^[18]等. 零知识验证协议流程如图 2 所示. 值得注意的是本文提出的水印验证方案也适合其它的 FPGA IP 水印嵌入方法, 如 [5,9,12], 但本文使用经过散列处理的混沌序列置乱水印^[16]能增强水印的鲁棒性, 同时水印嵌入方法的资源开销为零 (如表 1 所示).

3.3.2 协议实现

在零知识 FPGA IP 水印检测协议中, FPGA 比特流文件的随机置换是零知识协议中重要的组成部分. 随机置换需满足 2 个特殊要求: 1) 要求随机置换次数足够多; 2) 要求随机置换的结果之间的相关性低. 混沌现象是在非线性动力系统中出现的确定性的、类似随机的过程, 这种过程既非周期又不

表 1 资源开销比较

Table 1 Resource overhead comparison

Benchmark circuits	FPGA	#Slices in Original IP core	Method in [5,9,12]		Ours	
			#Slices in	Slice	#Slices in	Slice
			Watermarked IP core	Overhead	Watermarked IP core	Overhead
Spi	XC2V250-6cs144	396	428	8.081%	396	0%
Vga_lcd	XC2V1500-6bg575	828	860	3.865%	828	0%
B14	XC2V250-6cs144	1231	1263	2.600%	1231	0%
Pci_bridge32	XC2V1500-6bg575	1395	1427	2.294%	1395	0%
S38584	XC2V1500-6bg575	1626	1658	1.968%	1626	0%
S38417	XC2V1000-6bg575	1766	1798	1.812%	1766	0%
Ethernet	XC2V4000-6bf957	2372	2404	1.349%	2372	0%
B20	XC2V1000-6bg575	2385	2417	1.342%	2385	0%
B21	XC2V1000-6bg575	2538	2570	1.261%	2538	0%
B22	XC2V1000-6bg575	3700	3732	0.865%	3700	0%
Des_perf	XC2V4000-6bf957	7498	7530	0.427%	7498	0%
Average				2.35%		0%

收敛, 混沌系统对初值敏感, 易于产生数量众多、互相关性极低的伪随机数序列, 且混沌序列具有良好的统计属性^[19]. 由于混沌序列的这些特点刚好满足位流文件随机置换的特殊要求, 因此本文将基于混沌理论的镜像加密算法^[20]用于零知识验证过程中的随机位置置换. 该随机置换算法首先产生二值混沌序列, 然后产生的二值混沌序列作为位置映射算法 $\pi(\rho)$ 的输入以控制位流文件核的 LUT 位置置换, $\pi(\rho)$ 应用于 $\hat{I}$ 得 $\mathcal{L}$, 下面详细介绍该协议的实现方法.

1) 二值混沌序列的产生.

一维离散混沌映射 $f: U \rightarrow U, U \subset R$,

$$x_{k+1} = f(x_k, \lambda), x_k \in U, \lambda \in R, \quad (1)$$

其中 k 表示迭代次数, λ 是控制系统混沌行为的参数. 对于 λ 的一个特定值, 有限空间 U 由周期序列和混沌序列组成, 恰当地选取 λ 值, x 就能形成混沌序列.

Logistic 映射是一类被广泛研究的动力系统, 其定义如下:

$$x_{k+1} = \mu x_k (1 - x_k), \quad (2)$$

其中 $\mu \in [0, 4]$, $x_k \in (0, 1)$. 当 $3.5699456 \cdots < \mu \leq 4$ 时, Logistic 映射工作于混沌状态, 即由初始条件 x_0 在 Logistic 映射下所产生的混沌序列是非周期、不收敛且对初值非常敏感的. 例如定义在区间 $(-1, 1)$ 上的 Logistic 映射, 其形式如下:

$$x_{k+1} = 1 - \mu x_k^2. \quad (3)$$

当我们选取参数 $\mu = 2$ 的满射条件下, 此时产生的混沌序列的概率分布函数 (probability density function) 可写为

$$\rho(x) = \begin{cases} \frac{1}{\pi\sqrt{1-x^2}}, & -1 < x < 1, \\ 0, & \text{else.} \end{cases} \quad (4)$$

通过式 (4) 可以得到本文所关注的混沌序列的 2 个统计特性: 混沌序列轨迹点的均值和混沌序列之间的互相关性.

混沌序列轨迹点的均值为

$$\bar{x} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} x_i^2 = \int_0^1 x \rho(x) dx = 0. \quad (5)$$

独立选取 2 个初始值 x_0 和 y_0 , 则序列的互相关性为

$$c(l) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} (x_i - \bar{x})(y_{i+l} - \bar{y}) = \int_0^1 \int_0^1 \rho(x, y)(x - \bar{x})(\tau^l(y) - \bar{y}) dx dy = 0. \quad (6)$$

由式 (2) 产生的实值混沌序列 x 用二进制表示^[19]:

$$x = 0.b_1(x)b_2(x), \dots, b_i(x) \dots, \quad x \in [0, 1], \quad b_i(x) \in \{0, 1\}, \quad (7)$$

其中 $b_i(x)$ 如式 (8) 所示,

$$b_i(x) = \sum_{r=1}^{2^i-1} (-1)^{r-1} \Theta_{(r/2^i)}(x), \quad (8)$$

其中 $\Theta_t(x)$ 为阈值函数, 如式 (9) 所示,

$$\Theta_t(x) = \begin{cases} 0, & x < t, \\ 1, & x > t. \end{cases} \quad (9)$$

二值化后可得二值混沌序列:

$$\rho_i^n = \{b_i(\tau^n(x))\}_{n=0}^{\infty}, \quad (10)$$

其中 n 是序列的长度, $\tau^n(x)$ 是 logistic 映射的第 n 次迭代. 值得注意的是二值混沌序列仍具有很好的自相关及互相关特性^[21]. ρ_i^n 在随机位置置换算法中起着重要的作用, 用于控制 LUT 之间是否进行位置置换.

2) 随机位置置换算法.

定义 1 忽略时序逻辑, FPGA 可看作由 M 行 N 列 LUT 构成的二维阵列, 每个 LUT 在 FPGA 上的位置编号 (x, y) , 因此位流文件为一个二维数组 $I_{M \times N} = [f(i, j)]_{M \times N}$ (N, M 分别表示 x, y 方向上 LUT 的数目), 数组元素 $f(i, j)$ ($0 \leq i \leq M-1, 0 \leq j \leq N-1$), 是在位置 (i, j) 处的 LUT 内容.

定义 2 位置置换算法为 $\pi(\rho)$, 其中 ρ 代表二值混沌序列. 忽略可布线性, 位置置换后的设计为 $\hat{I}_{M \times N}$, 即 $\hat{I}_{M \times N} = \pi(\rho)I_{M \times N}$.

基于定义 1 和定义 2, 我们使用混沌镜像加密算法对 $I_{M \times N}$ 随机位置置换, 置换过程如下:

Step 1: 确定一维混沌系统 (本文采用 Logistic 映射, 如式 (3) 所示) 和初值 $x(0)$.

Step 2: 根据式 (3) 生成实值混沌序列 $x(k)$ ($k = 0, 1, 2, \dots, n$).

Step 3: 根据式 (7)~(10) 二值化混沌序列 $x(k)$ ($k = 0, 1, 2, \dots, n$) 得 $\rho(k)$ ($k = 0, 1, 2, \dots, n$).

Step 4: 根据所使用的 FPGA 设计 $I_{M \times N}$ 确定 M 和 N 的值. 设置 $k = 0$.

Step 5: for $i = 0$ to $M/2-1$

for $j = 0$ to $N-1$

Swap $_{\rho(k)}$ ($f(i, j)$, $f(i + M/2, j)$);

$k = k + 1$;

end

end

Step 6: for $i = 0$ to $M-1$

for $j = 0$ to $N/4-1$ and $N/2$ to $(3/4)*N-1$

Swap $_{\rho(k)}$ ($f(i, j)$, $f(i, j + N/4)$);

$k = k + 1$;

end

end

Step 7: for $i = 0$ to $M/4-1$ and $M/2$ to $(3/4)*M-1$

for $j = 0$ to $N-1$

Swap $_{\rho(k)}$ ($f(i, j)$, $f(i + M/4, j)$);

$k = k + 1$;

end

end

Step 8: for $i = 0$ to $M-1$

for $j = 0$ to $N/2-1$

Swap $_{\rho(k)}$ ($f(i, j)$, $f(i, j + N/2)$);

$k = k + 1$;

end

end

Step 9: 位置置换完成, 停止算法.

算法中的交换操作 Swap $_{\rho(k)}$ (A, B): 当 $\rho(k) = 1$ 时交换 A 和 B 的值, 当 $\rho(k) = 0$ 时不交换 A 和 B 的值.

随机位置置换的安全性非常重要, 定理 1 证明了本文使用的随机位置置换算法具有高安全性.

定理 1 除了二值混沌序列, 公开可验证协议中的位置置换算法是公开的. 基于混沌二值序列的随机置换算法可用的位置置换数目是 $2^{2 \times M \times N}$.

证明 从算法中可看出 LUT 是否进行位置置换由二值混沌序列控制, 算法的第 5 步到第 8 步所需的二进制比特数为 $2 \times M \times N$. 因此, 可能的位置置换的结果数为 $2^{2 \times M \times N}$, 证毕.

混沌系统对初始条件 $x(0)$ 敏感, $x(0)$ 在协议的每轮循环取值都不一样, 甚至每个 IP 验证时我们可以使用不同的混沌系统和不同的 μ 产生混沌序列, 因此每轮验证使用的二值混沌序列不可预测. 以测试电路 b22 为例, $M = 320$, $N = 63$. 可能的位置置换数为 $2^{2 \times 320 \times 63} = 2^{40320}$. 巨大的位置置换数以及每轮验证都使用不同的混沌序列进行位置置换使得攻击者很难通过收集原始的位流文件核和它们的置换结果来预测下一次的位置置换结果, 因此本文采用的位置置换算法具有高安全性.

					F-LUT		G-LUT	
12 Bytes		2 Bytes	1Byte	2 Bytes	2 Bytes	1Byte	2 Bytes	12 Bytes

图 3 LUT 在帧中的位置 [22]

Figure 3 The positions of LUTs in frames [22]

4 协议分析

4.1 嵌入攻击分析

非诚实的 IP 购买者 (验证者) 使用了未经授权的 IP, 诚实的 IP 所有者 (证明者) 想要证明此 IP 中含有自己的水印. 在实际的公开验证过程中, 验证者常常是不可信的, 因为验证者会努力使得诚实的 IP 所有者无法证明自己非法使用了它的 IP: 即使自己明明非法使用了 IP 所有者的 IP 而 IP 所有者无法证明成功. 由于 FPGA IP 本质上是一个位流文件, 恶意的攻击者容易在该文件中嵌入非法的水印. 因此, 现有的 FPGA 零知识水印检测系统易遭受嵌入攻击. 下面以 Xilinx Virtex-II 或 Virtex-II Pro FPGA 为例介绍实现嵌入攻击的方法.

位流文件核由 packet 构成. 这些 packet 中有一个 packet 包含配置内容, 配置内容由很多帧 (frame) 组成. 这些 frame 是 Virtex-II 配置内存空间的最小可编址段. Frame 中 LUT 的位置由图 3 所示. 一个 Slice 由 5 字节构成, 其中包含一个 2 字节的 G-LUT、2 字节的 F-LUT 和 1 个分隔字节, 其中 F-LUT 的内容是反序的.

根据这些信息, 文献 [22] 给出了 G-LUT 和 F-LUT 在配置位流文件中的地址计算公式, 如式 (11) 和 (12) 所示:

$$\text{Addr}_{\text{G-LUT}}(x, y) = (31 + \lfloor x/2 \rfloor \times 22 + x \bmod 2) \times F_{\text{Len}} + 12 + ((S_r - 1) - y) \times 5, \quad (11)$$

$$\text{Addr}_{\text{F-LUT}}(x, y) = \text{Addr}_{\text{G-LUT}}(x, y) + 3, \quad (12)$$

其中 F_{Len} 表示帧的长度, (x, y) 表示 Slice 的坐标, S_r 表示 FPGA 中 Slice 行的数目.

FPGA 位流文件中空闲的 LUT 的内容为全 1 或全 0 [22], 根据式 (11) 和 (12), 攻击者很容易定位未水印的空闲 LUT 的位置, 然后随机选取一些空闲的 LUT 嵌入自己的水印信息.

因此要防止嵌入攻击、避免侵权者的抵赖, 不仅需要对 FPGA 位流文件核进行数字签名, 还必须保证该 FPGA IP 在某一时间之前的存在性, 我们借助于时间戳 [14] 机制很容易解决该问题. 本文提出的基于时间戳认证的水印方案的基本思想是: 作者在设计的 IP 出售之前, 向时间戳权威机构申请时间戳, 将代表自己身份的水印、IP、以及申请的时间戳三者进行绑定, 以此证明某个时间开始设计者对 IP 的所有权. 方法如下:

在出售已水印的 IP 之前, IP 所有者必须发送该 IP 给时间戳权威机构, 请求加盖时间戳, 时间戳权威机构记录收到的 IP 的日期和时间 T , 然后使用对称私钥为 IP 所有者添加签名的时间戳, 接着将时间戳的 IP 发回给 IP 所有者. 这样, 当发生版权纠纷时, 攻击者得到拷贝的 IP 并实施嵌入攻击嵌入伪水印的时间必然滞后于 T , 在时间戳验证阶段, 时间戳权威机构基于时间戳值公示该 IP 的实际拥有者以解决嵌入攻击造成的非法抵赖.

4.2 协议属性分析

零知识协议必须满足 3 个属性: 完全性 (completeness)、可靠性 (soundness) 和零知识性 (zero-

knowledge). 下面分别讨论本文所提出的零知识方案的这 3 个属性.

完全性: 在所提出的零知识协议的每轮验证中, 验证者通过抛硬币决定 Alice 是执行 Step 2a 还是 Step 2b. 执行 Step 2a 或 Step 2b (只要 Step 2a 和 Step 2b 不同时执行) 不会泄漏水印在 $\hat{I}$ 的真实位置. 在协议执行多轮的情况下, 如果 Alice 确实能证明 Step 2a 或 Step 2b, 那么验证者总是接受证明者的证明的, 因此一个诚实的证明者总能让验证者相信他确实在 FPGA IP 中嵌入了水印, 该协议是完全的.

可靠性: 在协议的某一轮验证中, 欺骗的验证者无法抵赖诚实的证明者公布的验证结果; 欺骗的证明者若使用包含真实水印的另一个设计给验证者验证, 验证者通过抛硬币以 $1/2$ 的概率来验证 Step 4a 或 Step 4b, 若选择 Step 4b 则验证通过, 若选择 Step 4a 则验证失败, 因此本文所提出的零知识协议执行一次, Alice 能成功欺骗 Bob 的概率为 $1/2$, 协议迭代 R 次后 Alice 都能成功欺骗 Bob 的概率为 $1/2^R$, 协议验证的可靠性 $S = 1 - 1/2^R$. 当 R 足够大时, S 趋近于 1, 因此, 一个不诚实的证明者, 他能欺骗验证者相信他在 FPGA 设计中嵌入了合法水印的可能性可以忽略.

零知识性: 在协议的任意一轮迭代中, 若选择执行 Step 2a, 水印的内容和水印的位置都不会被泄露; 若选择执行 Step 2b, 泄露的是位置置换后的位流文件核的水印位置, 这些位置并不是原始设计中水印的位置, 因此这些位置是无效信息. 其次, 由于混沌所固有的优点, 在协议的每次迭代中, 位置置换都不一样且每轮位置置换的相关系数极低, 这使得攻击者很难通过收集原始的位流文件核和它们的置换结果来预测下一次的位置置换结果. 验证协议执行后, 验证者获得的关于水印内容和水印位置的信息与他验证前知道的一样, 因此, 在公开验证后本文所提出的零知识方案不会泄露水印的内容和水印的位置信息, 该协议是零知识的.

5 实验结果分析与比较

在本文实验中, 综合工具使用 Xilinx ISE 9.1i, 实验电路来源于 IWLS2005 基准测试电路, 该基准测试电路集的电路种类众多, 不仅包含了简单有效的模块单元电路而且还包含了复杂的现实大规模数字电路. 四种不同类型的 Xilinx FPGA 芯片 (XC2V250、XC2V1000、XC2V1500 和 XC2V4000) 用于验证和实现本文提出的水印方案和文献 [12] 的水印思想. 本文所提出的零知识协议由 Matlab 和 C 语言编程实现, 并在一台操作系统为 Windows XP SP3 的 PC 机上进行. 配置 CPU 主频为 3.0 GHz、内存为 3.0 G 的 Intel®Pentium®E5700 机器. 我们将签名信息生成水印后按嵌入算法嵌入到这些标准测试电路程序中, 通过水印的开销实验和零知识协议的位置置换鲁棒性实验对本文的所提出的水印方案进行分析, 实验结果与最新的相关文献 [12] 进行比较.

5.1 水印开销

水印的资源开销和时延开销分别用 Slice 开销和最小时钟周期退化来衡量. 表 1 和表 2 分别给出了嵌入 1024 bits 水印前后的 IWLS2005 基准测试电路的资源开销和时延开销. 为了与最新的 FPGA 零知识验证思想 [12] 比较, 我们实现了文献 [12] 的水印思想. 因为文献 [12] 的水印思想和文献 [5,9] 一样, 水印的载体是未用的 Slice, 因此它的 Slice 开销随水印嵌入量成正比, 而本文水印的载体是已经使用的 Slice 中的未用的 LUT, 因此水印嵌入的 Slice 开销为 0. 此外, 方法 [5,9,12] 对于所耗逻辑资源较小的 IP, Slice 开销大; 而对所耗逻辑资源较大的 IP, Slice 开销相对较小. 从表 1 的实验结果可以看出

表 2 时延开销比较

Table 2 Timing overhead comparison

Benchmark circuits	FPGA	#Minimum	Method in [5,9,12]		Ours	
		clock period	#Minimum clock period	Timing	#Minimum clock period	Timing
		in original	in watermarked	overhead	in watermarked	overhead
		IP core (ns)	IP core (ns)		IP core (ns)	
Spi	XC2V250-6cs144	10.197	10.209	0.12%	10.197	0%
Vga_lcd	XC2V1500-6bg575	6.027	5.999	−0.46%	6.027	0%
B14	XC2V250-6cs144	17.123	15.210	−11.17%	17.123	0%
Pci_bridge32	XC2V1500-6bg575	8.874	9.079	2.31%	8.874	0%
S38584	XC2V1500-6bg575	5.585	6.176	10.58%	5.585	0%
S38417	XC2V1000-6bg575	8.904	9.672	8.63%	8.904	0%
Ethernet	XC2V4000-6bf957	6.970	7.217	3.54%	6.970	0%
B20	XC2V1000-6bg575	17.019	20.297	19.26%	17.019	0%
B21	XC2V1000-6bg575	16.523	15.696	−5.01%	16.523	0%
B22	XC2V1000-6bg575	16.310	16.838	3.24%	16.310	0%
Des_perf	XC2V4000-6bf957	7.254	7.092	−2.23%	7.254	0%

文献 [12] 的 Slice 开销为 8.081% ~ 0.427% (平均为 2.35%) 而本文提出的思想无论是对所耗逻辑资源较小的 IP 还是对所耗逻辑资源较大的 IP 的 Slice 开销都为 0. 需要注意的是时延开销是水印性能的重要衡量指标, 但文献 [12] 未给出时延开销数据, 因此我们测试了时延开销. 从表 2 看出文献 [5,9,12] 的时延开销有正有负, 范围为 -11.17% ~ 19.26%, 水印数量对时延的影响是随机和非单调的 [5], 而本文提出的思想不会影响设计的最小时钟周期. 这是因为我们在电路的物理版图嵌入水印, 对原始设计的布线修改很小, 由于水印对设计的布线的影响会造成设计的部分 “hold to clock clk”, “setup to clock clk” 和 “clock clk to pad” 变化, 但最小时钟周期保持不变 (表 2 所示), 因此实际上时延开销几乎为 0, 本文提出的方法在开销上具有明显的优势.

H.264/AVC 是一种工业的视频压缩标准. 它是目前最常用的录像、压缩和高清视频格式之一. 它被广泛应用于蓝光系统、YouTube、iTunes 和其他互联网视频分布中心. 尤其是对拥有视频解码能力的移动智能手机和其它低功耗便携式系统来说, 设计高效的 H.264/AVC 是一个巨大的挑战. Nova¹⁾ 是一种针对移动应用的低功耗的 H.264/AVC 基线解码器. Nova 使用 Verilog 硬件描述语言编写. 为了验证我们提出的水印思想对大规模集成电路的适用性, 我们选择在 Nova 中嵌入水印. 水印嵌入流程如 3.2 小节所述. 水印前后的 Nova 都在 FPGA 器件 XC5VLX50T-3ff1136 上实现, XC5VLX50T-3ff1136 可用 Slices 总数为 7200, Nova 综合后 Slices 占用率为 98%. 嵌入 1024 bits 水印前后的实验数据如表 3 所示. 实验结果表明本文提出的水印方案的 Slice 开销和时延开销都为零.

5.2 位置置换鲁棒性

位置置换鲁棒性是衡量零知识协议安全性的重要指标. 我们同文献 [12] 一样使用平均曼哈顿距离 $\mu_N(\delta_L)$ 、曼哈顿距离的标准差 $\sigma_N(\delta_L)$ 及 LUT 的位置与曼哈顿距离之间的相关系数 $\rho_N(l, \delta_L)$ 来衡量

¹⁾ H.264/AVC Baseline Decoder :: Overview. [Online] Available at: <http://opencores.org/project,nova>

表 3 水印 Nova 的实验数据

Table 3 Experimental data of watermarking Nova

	Original Nova	Watermarked Nova	Watermarking overhead
# Slices	7108	7108	0%
#6-input LUTs	26105	261121	0.06%
#Minimum clock period	13.306 ns	13.306 ns	0%

所提出的置换算法的位置置换的鲁棒性. $\sigma_N(\delta_L)$ 越大, $\rho_N(l, \delta_L)$ 越小说明位置置换的鲁棒性越好. 设 (x_k, y_k) 和 (x'_k, y'_k) 分别为第 k 个 LUT 的位置置换之前的位置和位置置换之后的位置. 第 k 个 LUT 位置置换前后的曼哈顿距离为: $\delta_{L_k} = |x_k - x'_k| + |y_k - y'_k|$. 则平均曼哈顿距离为

$$\mu_N(\delta_L) = \frac{1}{N} \sum_{k=1}^N \delta_{L_k}. \quad (13)$$

标准差为

$$\sigma_N(\delta_L) = \left[\frac{1}{N} \sum_{k=1}^N (\delta_{L_k} - \mu_N(\delta_L))^2 \right]^{\frac{1}{2}}. \quad (14)$$

设第 k 个 LUT 的位置为 $l_k = (x_k, y_k)$ 则 LUT 的位置与曼哈顿距离之间的相关系数用 $\rho_N(l, \delta_L)$ 衡量:

$$\rho_N(l, \delta_L) = \left[\rho_N(x, \delta_L)^2 + \rho_N(y, \delta_L)^2 \right]^{\frac{1}{2}}, \quad (15)$$

其中,

$$\rho_N(x, \delta_L) = \frac{\text{cov}(x, \delta_L)}{\sigma_N(x) \sigma_N(\delta_L)} = \frac{N \sum_{k=1}^N x_k \delta_{L_k} - \left(\sum_{k=1}^N x_k \right) \left(\sum_{k=1}^N \delta_{L_k} \right)}{\left[N \left(\sum_{k=1}^N x_k^2 \right) - \left(\sum_{k=1}^N x_k \right)^2 \right]^{\frac{1}{2}} \left[N \left(\sum_{k=1}^N \delta_{L_k}^2 \right) - \left(\sum_{k=1}^N \delta_{L_k} \right)^2 \right]^{\frac{1}{2}}},$$

$\rho_N(y, \delta_L)$ 的计算类似.

由表 4 可以看出由于本文使用的位置映射算法基于混沌理论, 其随机性有密码学的理论保证, 因此本文提出的零知识水印验证思想的位置映射的鲁棒性与文献 [12] 相比明显增强, 其中 $\rho_N(l, \delta_L)$ 的数量级为 $10^{-2} \sim 10^{-3}$, 大部分比文献 [12] 降低一个数量级.

6 结论

本文提出的基于混沌序列的公开可验证 FPGA IP 水印检测方案不会泄露有关水印内容和水印位置的任何信息, 同时采用时间戳机制有效的解决了现有 FPGA 水印检测方案易受嵌入攻击的问题. 混沌系统所固有的优点刚好满足零知识协议中对位流文件随机位置置换的特殊要求, 使得本文所提出的验证方案的随机位置置换具有极高的鲁棒性. 实验结果表明与最新的零知识 FPGA IP 水印验证方案相比, 本文提出的方案具有零开销和更高的位置映射鲁棒性.

现有的 FPGA 零知识验证协议的可靠性 $S = 1 - 1/2^R$, 只有当 R 足够大时, S 才趋近于 1, 因此协议需执行 R 轮, 效率较低. 针对 FPGA IP 水印检测这一特定的应用, 提出一种 S 趋近于 1 的常数轮

表 4 位置置换鲁棒性

Table 4 The robustness of location permutation

Benchmark circuits	Size of array	#4-input LUTs in watermarked IP core	$\mu_N(\delta_L)$	$\sigma_N(\delta_L)$	$\rho_N(l, \delta_L)$	$\mu_R(\delta_L)$	$\sigma_R(\delta_L)$	$\rho_R(r, \delta_L)$
Spi	192×32	765	33.77	15.07	0.0288	31.46	14.83	0.0297
Vga_lcd	384×80	1327	41.91	15.02	0.0097	39.75	14.86	0.0107
Pci_bridge32	384×80	1911	43.55	15.98	0.0126	42.51	15.47	0.0130
Ethernet	640×144	3089	112.61	31.27	0.0073	110.46	30.61	0.0097
Des_perf	640×144	10556	113.58	32.96	0.0082	111.97	32.11	0.0113
B14	192×32	2501	31.27	14.63	0.0197	30.51	14.28	0.0203
S38417	320×63	2590	38.16	16.71	0.0074	36.24	15.89	0.0087
B20	320×63	4720	37.24	15.92	0.0081	36.83	15.42	0.0084
B21	320×63	5069	37.81	15.98	0.0086	37.10	15.17	0.0152
B22	320×63	7361	37.96	16.14	0.0100	37.06	14.83	0.0129
S38584	384×80	3053	43.91	16.88	0.0083	42.19	16.18	0.0091

零知识的公开可验证协议是我们下一步的工作重点. 此外, 水印是一种“被动”的 FPGA IP 保护技术, 如何研究一种“主动”的 FPGA IP 保护技术^[23,24]是一个极具挑战性的课题.

参考文献

- 1 Yip K, Ng T. Partial-encryption technique for intellectual property protection of FPGA-based products. *IEEE Trans Consum Electr*, 2000, 46: 183–190
- 2 Tehranipoor M, Wang C. Introduction to hardware security and trust. Springer Publishing Company, Incorporated, 2011. 1–38
- 3 Cui A, Chang C H, Tahar S, et al. A robust FSM watermarking scheme for IP protection of sequential circuit design. *IEEE Trans Comput Aid D*, 2011, 30: 678–690
- 4 Chang C H, Cui A. Synthesis-for-testability watermarking for field authentication of VLSI intellectual property. *IEEE Trans Circuits-I*, 2010, 57: 1618–1630
- 5 Lach J, Mangione-Smith W H, Potkonjak M. Robust FPGA intellectual property protection through multiple small watermarks. In: *Proceedings of 36th IEEE Design Automation Conference, New Orleans, 1999*. 831–836
- 6 Qu G. Publicly detectable techniques for the protection of virtual components. In: *Proceedings of the 38th Annual Design Automation Conference, New York, 2001*. 474–479
- 7 Qu G. Publicly detectable watermarking for intellectual property authentication in VLSI design. *IEEE Trans Comput Aid D*, 2002, 21: 1363–1368
- 8 Zhang J L, Lin Y P, Wu Q, et al. Watermarking FPGA bitfile for intellectual property protection. *Radioengineering*, 2012, 21: 764–771
- 9 Lach J, Mangione-Smith W H, Potkonjak M. Fingerprinting techniques for field-programmable gate array intellectual property protection. *IEEE Trans Comput Aid D*, 2001, 20: 1253–1261
- 10 Adelsbach A, Sadeghi A R. Zero-knowledge watermark detection and proof of ownership. In: *Proceedings of International Workshop on Information Hiding, 2001*. 273–288
- 11 Beth T. Efficient zero-knowledge identification scheme for smart cards. In: *Proceedings of Advances in Cryptology-Eurocrypt'88, LNCS 330, Heidelberg, 1988*. 77–86

- 12 Saha D, Sur Kolay S. Secure public verification of IP marks in FPGA design through a zero-knowledge protocol. *IEEE Trans VLSI Syst*, 2012, 20: 1749–1757
- 13 He Y Z, Wu C K, Feng D G. Publicly verifiable zero knowledge watermark detection. *J Softw*, 2005, 16: 1606–1616
[何永忠, 武传坤, 冯登国. 公开可验证的零知识水印检测. *软件学报*, 2005, 16: 1606–1616]
- 14 Haber S, Stornetta W S. How to time-stamp a digital document. *J Cryptol*, 1991, 3: 99–111
- 15 Dehon, A. Balancing interconnect and computation in a reconfigurable computing array (or, why you don't really want 100LUT utilization). In: *Proceedings of ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, 1999. 69–78
- 16 Wang H X, He C, Ding K. Public watermarking based on chaotic map. *IEICE Trans Fund*, 2004, 87: 2045–2047
- 17 Beker H, Piper F. Cipher systems: the protection of communications. Northwood Books, 1982
- 18 Dubrova E, Teslenko M, Tenhunen H. On analysis and synthesis of (n,k)-non-linear feedback shift registers. In: *Proceedings of Design, Automation and Test in Europe (DATE2008)*, 2008. 1286–1291
- 19 Kohda T, Tsuneda A. Statistics of chaotic binary sequences. *IEEE Trans Inform Theory*, 1997, 43: 104–112
- 20 Yen J C, Guo J I. A new chaotic mirror-like image encryption algorithm and its VLSI architecture. *Pattern Recogn Image Anal*, 2000, 10: 236–247
- 21 Xiang T, Liao X, Tang G, et al. A novel block cryptosystem based on iterating a chaotic map. *Phys Lett A*, 2006, 349: 109–115
- 22 Zhang J L, Lin Y P, Che W, et al. Efficient verification of IP watermarks in FPGA designs through lookup table content extracting. *IEICE Electr Express*, 2012, 9: 1735–1741
- 23 Zhang J L, Lin Y P, Lyu Y Q, et al. Binding hardware IPs to specific FPGA device via inter-twining the PUF response with the FSM of sequential circuits. In: *Proceedings of IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM2013)*, Seattle, 2013. 227
- 24 Zhang J L, Lin Y P, Lyu Y Q, et al. FPGA IP protection by binding finite state machine to physical unclonable function. In: *Proceedings of 23rd International Conference on Field Programmable Logic and Applications (FPL2013)*, Portugal, 2013

A chaotic-based publicly verifiable FPGA IP watermark detection scheme

ZHANG JiLiang¹, LIN YaPing^{1*}, LYU YongQiang² & WANG XiangQi³

1 College of Information Science and Engineering, Hunan University, Changsha 410082, China;

2 Research Institute of Information Technology of Tsinghua University, National Laboratory for Information Science and Technology, Beijing 100084, China;

3 College of Mathematics and Computer Science, Hunan Normal University, Changsha 410006, China

*E-mail: yplin@hnu.edu.cn

Abstract IP (intellectual property) infringement has been a serious problem to FPGA vendors or IP designer. Watermarking is a novel technique for FPGA IP protection, while existing FPGA watermarking techniques may give away sensitive information during the public verification (e.g. the content or position of watermarks), which enables malicious verifiers or third parties to remove the watermarks and resell the design. Existing zero-knowledge watermarking verification schemes are able to address the sensitive information leakage issue but are vulnerable to embedding attacks, which makes it ineffective in preventing denying infringement of un-trusted buyers (verifiers). This paper presents a new chaotic-based publicly verifiable zero knowledge watermark detection scheme for FPGA IP protection. This scheme is both resilient against the sensitive information leakage issue and removing attacks, thus robust to cheating from provers, verifiers or the third party. In traditional FPGA watermarking techniques, watermarks are embedded in the used Slices, thus the area overhead is proportional to the number of embedded

watermarks. In the proposed scheme, watermarks are hidden in the unused LUTs of used Slices, introducing zero area and timing overheads. Chaotic systems are sensitive to initial values and convenient to produce numerous pseudo-random numbers with ultra-low coefficients and perfect statistical property. These advantages match the specific requirements for the random location permutation of the LUTs in the FPGA bit-stream file in the zero knowledge protocol, bringing an ultra-high robustness of LUT location permutation. The timestamp mechanism is also introduced in this paper to resist embedding attacks against denying the infringement from un-trusted IP buyers (verifiers). Experimental results and analysis show that the proposed scheme is significantly better than the latest related literatures in both watermarking overheads and robustness of location permutation.

Keywords watermarking, FPGA, IP protection, chaotic, publicly verifiable, zero knowledge, time-stamp

ZHANG JiLiang was born in Changde, China, 1986. He received the B.E. degree from Shandong University of Science and Technology, Qingdao, China in 2009. During 2010–2011, he worked as a joint training graduate student at Tsinghua University. He is currently working toward his Ph.D. degree in the College of Information Science and Engineering, Hunan University. In 2013–2014, He works as a

visiting Ph.D. student at the Department of Electrical and Computer Engineering, University of Maryland, College Park. He is a student member of ACM and CCF. His research interests include watermarking and Physical Unclonable Functions (PUFs) for VLSI/FPGA intellectual property protection, VLSI physical design, dynamic reconfigurable System-on-Chip.

LIN YaPing was born in Shaoyang, China, 1955. He received his BS degree at Hunan University, in 1982, MS degree at National University of Defense Technology, in 1985, and PhD degree at Hunan University, in 2000. He got an exception promotion to be an professor in 1996, 1991–1993, he works as the deputy director of Department of Computer Science, Hunan University; 1993–1999, he became the director of

this Department; 2004–2005, he has been to Arlington campus, the University of Texas in the United States for study visit; he became the dean of the Software School, Hunan University in 1999. He works as an Ph.D. supervisor in the Laboratory of Computer Networking and Machine Learning, Hunan University. His research interests include network and distributed system security. Currently, He focuses on wireless security and system security.

LYU YongQiang received the B.S. degree in Computer Science from Xi Dian University in 2001, the M.S. degree and the Ph.D. degree in Computer Science from Tsinghua University in 2003 and 2006 respectively. From 2006 to 2009 he was with Synopsys ATG and Magma where his main responsibility was the research and development on the leading techniques and products in physical synthesis. Currently he is an

assistant professor in Research Institute of Information Technology at Tsinghua University. His research interest focuses on the hardware-software fusion architecture in emerging computing systems, such as Internet of Things, mobile computing and reconfigurable computing.

WANG XiangQi was born in Hunan, China, 1989. She received the B.E. degree from Changsha University, Changsha, China in 2011. Currently, she is a Ph.D. candidate in College of Mathematics and Computer Science, Hunan Normal University. Her research interests include the numerical solution of differential equations.