FROPUF: 从基于 FPGA 的震荡环 PUF 中提取更多的熵

李昌婷[1,2], 章庆隆[1], 刘宗斌[1], 荆继武[1]

[1] 中国科学院信息工程研究所, 北京 中国 100093
[2] 中国科学院大学, 北京 中国 100049

摘要 FPGA 平台上基于振荡环的物理不可克隆方法(Ring Oscillator based Physically Unclonable Function, 以下简称 RO PUF)以其简洁的架构和优良的属性而备受青睐。但是常用的 RO PUF 结构只能通过比较一对振荡环的频率, 从两个振荡环里提取到 1 比特的熵。在很多应用中, 尤其是基于 PUF 技术的密钥生成和随机数生成中, PUF 响应能够保证提供足够的熵至关重要, 为此, RO PUF 需要部署大量的振荡环从而会消耗更多的资源。硬件资源利用率的低下极大限制了 RO PUF 的应用范围, 尤其是资源受限的情景。针对这个问题, 我们提出了一种简洁高效的方法, 通过利用可编程延迟线(Programmable Delay Line, PDL)对延迟路径的精细控制, 可以从每个振荡环中提取到相当于目前最优方案 6 倍的熵。我们将这种新型 RO PUF 结构命名为深度 RO PUF(Further RO PUF)。本文不仅详细介绍了如何利用从实现振荡环的查找表(Look Up Table, LUT)中推导出的潜在随机变量, 还展示更深层的制造差异变量是如何通过类似于高阶差分算法来提取的。除此之外, 我们还建立了模型进行仿真并在 Xilinx Virtex-6 和 Zynq-7000 系列评估板上进行了实验, 通过展示仿真和实验结果的一致性来证明我们所提出方法的有效性和正确性。

关键词 PUF; 振荡环; 熵; 高阶差分; 可编程延迟线; FPGA
中图分类号: TP309.1 DOI 号 10.19363/j.cnki.cn10-1380/tn.2018.01.002

FROPUF: To Extract More Entropy from Two Ring Oscillators in FPGA-Based PUFs

LI Changting[1, 2], ZHANG Qinglong[1], LIU Zongbin[1], JING Jiwu[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China

Abstract Ring Oscillator based Physically Unclonable Function (RO PUF) on FPGAs is popular for its nice properties and easy implementation. However, the conventional RO PUF only extracts 1-bit entropy by comparing two ROs' frequencies. For many applications, in particular for PUF-based key protection and random number generation, it is imperative that PUF responses provide sufficient entropy. In order to acquire adequate entropy, numerous ROs will be needed. RO PUF's inefficiency in hardware utilization constrains its application range, particularly in resource-constrained environments. Motivated by this inefficient resource usage, we propose an elegant and efficient method which can extract 6 times more entropy than the latest proposals by utilizing the fine control of Programmable Delay Lines (PDL). We call this construction Further ROPUF (FROPUF). In this paper, we present in detail how to take advantage of the underlying random process variation which derives from the Look Up Tables (LUT) of two ring oscillators, and show that the in-depth variation can be extracted by a high order difference calculation. In addition, we reveal the consistency of the evaluation results from Xilinx FPGAs (e.g. Virtex-6, Zynq-7000 65nm) and those by simulation of FROPUF, which confirms the effectiveness and correctness of the proposed method.

Key words PUF; Ring Oscillator; Entropy; High Order Difference; Programmable Delay Lines; FPGA

1 引言

FPGA 以其灵活可重配的特性越来越受到开发设计者的青睐, 被应用于密码算法的硬件实现。在密码算法的实现和使用过程中, 密码运算模块的安全性是一个很重要的因素。密码运算的安全有一个必不可缺的条件是密钥的安全, 包括密钥生成、存储、使用、销毁等整个密钥生命周期管理的安全性。传

通讯作者: 刘宗斌, 高级工程师, 硕士生导师, Email: liuzongbin@iie.ac.cn。
收稿日期: 2016-07-13; 修改日期: 2016-11-03; 定稿日期: 2017-12-05。

统意义上, 可以用受保护的存储单元来存储秘密信息, 防止秘密信息被非法读取, 但是在实际应用中, 这种受保护的存储单元的实现成本和难度都很大。近年来, 针对密钥的攻击手法, 尤其是硬件攻击手段和工具都得到了飞速的发展, 传统的密钥存储方案面临着很大威胁。

在这种趋势下, 需要一种全新的方法来保证密钥的安全。物理不可克隆方法(Physically Unclonable Function, 以下简称 PUF)便可以作为物理信任根, 用于解决密钥的生成和存储问题。这种新型方法并不是尽量去隐藏硬件中的密钥, 而是通过硬件电路的随机物理特征来提取密钥信息。因为电路物理特征的固有性, 密钥可以实时生成, 无需存储。由于在电路制造过程中随机因素的干扰, 即使在设计上一模一样的两块电路, 制造出来后也存在微小的物理特性差异。物理不可克隆方法就是利用这些内在的随机制造差异来提取独一无二的电路指纹从而解决密钥生成、存储等问题。

到目前为止, 研究人员已经提出了多种物理不可克隆方法, 例如 SRAM PUF[1, 2]、Arbiter PUF[3, 4]、Butterfly PUF[5]、Glitch PUF[6,7]、Ring Oscillator PUF[8-14] 等。然而, 其中很多 PUF 难以在现有的商用 FPGA 平台上部署。目前最大的 FPGA 厂商 Xilinx 和 Altera 生产的 FPGA 芯片, 其 SRAM 的上电初始值会被强制设置为复位值, 使得在这些 FPGA 平台上难以构建 SRAM PUF。许多其他的 PUF 类型, 比如 Butterfly PUF 和 Arbiter PUF 则需要非常精细对称的布局布线, 而这一要求在 FPGA 上也是很难实现的。特别的, 构建 Butterfly PUF 的基础单元需要带复位信号和清零信号的锁存器(LATCH), 这种锁存器只在 Xilinx 的 Virtex-5 系列中提供, 在新版的 Virtex-6,7 系列已不再提供同时带这两个信号的锁存器。

Suh 和 Devadas 首次提出的振荡环 PUF(Ring Oscillator PUF, 以下简称 RO PUF)以其简洁的架构和优良的属性, 在 FPGA 以及 ASIC 平台上得到广泛应用。在 FPGA 平台上, 通过硬宏单元(Hard-Macro)技术能够构造出布局布线一样的振荡环模块, 这给部署 RO PUF 带来了很大的便利。按照 Suh 和 Devadas[15]提出的最基本的 RO PUF 结构是通过比较两个布局布线完全一样的振荡环的频率的大小来提取 1 比特的随机信息。相比其他 PUF 结构, RO PUF 的资源利用率较低, 因此随着需要提取的随机信息的数量的增加, RO PUF 占用的资源也将成倍增加。

PUF 响应的熵是衡量某一 PUF 结构所能够提取的实体固有随机物理特征的标准。无论是用于密钥生成、随机数生成或是其他安全应用, PUF 响应能否提供足够的熵至关重要。为了保证 PUF 响应的熵满足应用的安全需求, 相关研究工作主要致力于以下三个方面:

首先由于 PUF 响应序列的分布是由非常复杂混沌的物理过程所决定, 所以往往难以精确计算其中的熵。一些如文献[16-18]的研究试图通过统计特征来建立某一 PUF 结构简化的底层元件模型, 从而推导出该 PUF 响应的分布表达式, 进而计算出响应中的熵。

因为 PUF 响应是带噪的且分布不一定足够均匀, 所以在 PUF 的应用中常引入模糊提取器。模糊提取器一般由一个安全概略(Secure Sketching)和一个随机提取器(Randomness Extractor)组成, 前者用于纠正带噪的响应, 后者则从纠正后的 PUF 响应中提取更为随机的序列。然而进行模糊提取过程也引入了新的熵损失的风险, 因此有许多研究也致力于对熵损失情况的分析和估计, 或者模糊提取器的性能评价[19-21]。

还有部分研究则是改造已有的 PUF 结构, 甚至提出新型的 PUF 结构, 通过提高硬件利用率来保证在一定的资源消耗下 PUF 响应能够提供足够的熵[22-25]。就 RO PUF 来说, 如果可以提高振荡环的随机信息的提取率, 那么提取等量的随机信息消耗的资源就能有效减少。针对这个问题, Habib[22]提出了一种基于可配置 LUT 延迟的 RO PUF 来提高从振荡环提取的随机信息的方案, 并在 Xilinx 的 Spartan-3E 平台上, 验证了该方案的可行性。然而 Majzoobi 等人[26, 27]指出在 Xilinx Virtex-5 中对可配置 LUT 延迟的分析与 Habib 的实验现象不符, 根据 Habib 所说, 由于 FPGA 的型号不一样导致了可配置 LUT 延迟的分布的不同, 这就导致了 Habib 方案的局限性。

本文通过可配置的 LUT 搭建振荡环, 然后利用差分的计算方法提取更细粒度的制造差异, 从而提高两个振荡环所能提取的随机信息数量, 同时有效地消除局部环境噪声等等影响。我们将这种基于差分计算的新型的振荡环 PUF 命名为深度振荡环 PUF(Further RO PUF, 简称 FROPUF)。为了验证所提出的方案的有效性, 我们通过仿真计算和实验测试评估了 FROPUF, 评估结果表明 FROPUF 具有良好的可靠性和随机性。总的来说, 我们在本文中的主要工作如下:

1. 提出了一种简洁有效的基于高阶差分的提取算法来提取随机信息, 同时该算法可以降低局部噪声影响提高提取的随机信息的可靠性。

2. FROPUF 可以从两个振荡环中提取 31 比特的随机信息, 是 Habib 方案能够提取的熵的 6 倍, 大大提高了单位硬件资源所能提取的响应信息的数量。

3. 仿真计算和实际测试的结果的一致性表明了 FROPUF 的有效性。在常温 27℃ 下, 平均片间距离为 49.92%, 平均片内距离为 5.10%。

2 RO PUF 架构

2.1 RO PUF 的概念和基本结构

RO PUF 是一种基于传播延迟的电子 PUF。目前最常用的 RO PUF 的基本架构是 Suh 和 Devadas[15] 提出的, 如图 1 所示。在这个架构中 RO PUF 由 n 个完全一样的振荡环组成, 编号为 RO_1 到 RO_n, 他们的振荡频率分别为 f_i 到 f_n, 其中 MUX1 和 MUX2 是两个选通器, 通过 挑战信号, 选出一对振荡环 RO_i 和 $RO_j(i \neq j)$, 由于内在的制造差异使得 f_i 和 f_j 之间存在随机差异, 所以在相同时间间隔内计数器 1 和计数器 2 的计数值会有所差别, 通过(1)式便能计算得到 RO PUF 关于挑战 (i, j) 的响应 r_{ij}:

$$r_{ij} = \begin{cases} 1 & \text{如果 } f_i > f_j \\ 0 & \text{否则} \end{cases} \tag{1}$$

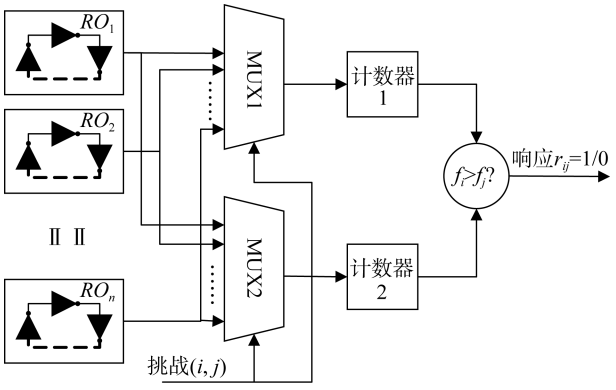

图 1 基本 RO PUF 原理图

Figure 1 Schematic diagram of basic RO PUF

2.2 PUF 的评估体系

评价一个 PUF 的性能, 需要考虑以下三个性质: 不可预测性(Unpredictability), 独特性(Uniqueness)和可靠性(Reliability)。

• 不可预测性是指即使在观测了 PUF 一定数量的挑战-响应对的情况下, 敌手猜测未观测过的挑战所对应的响应值的难易程度。为保证不可预测性, 一方面 PUF 的挑战-响应集应足够大, 另一方面, PUF 的挑战-响应行为应该难易被建模和模仿。

• 独特性是指一个 PUF 实体与其他同类 PUF 实体的区别度。一个好的 PUF, 其不同实体对于同一挑战产生的响应值的区别应该越大越好。

•可靠性是指在不同次测量中, 同一 PUF 实体对于同一挑战所产生的响应差异。由于噪声的影响, PUF 实体对于同一挑战的每一次响应并不是完全一样的。为了保证 PUF 的可用性, 其可靠性应该越高越好。

2.2.1 可靠性

可靠性可以通过片内距离(intra-distance)来衡量。简单来说, 片内距离的计算是多次将同一个挑战信息作用于同一个 PUF 实体得到的多次响应之间的汉明距离的平均值。虽然我们希望 PUF 的响应是稳定不变的, 但是环境因素, 例如温度变化、供电电压浮动等, 都会导致响应出现错误。为了评估 PUF 的稳定性, 设有 N_{puf} 个 PUF 实体, 对于每个实体分别输入 N_{chal} 个挑战测量其响应, 每个挑战测量 N_{meas} 次。如此, 我们便得到 $N_{puf} \times N_{chal} \times N_{meas}$ 个响应序列, 并计算平均片内距离如下:

$$\mu_{intra} = \sum_{\substack{j_1, j_2 = 1 \\ j_1 \neq j_2}}^{N_{meas}} \sum_{i=1}^{N_{puf}} \sum_{k=1}^{N_{chal}} \frac{2 \cdot HD\left(r_i^{j_1}(c_k), r_i^{j_2}(c_k)\right)}{N_{puf} \cdot N_{chal} \cdot N_{meas} \cdot \left(N_{meas} - 1\right)} \tag{2}$$

其中 c_k 表示第 k 个挑战, $r_i^j(c_k)$ 则代表 PUF 第 i 个实体输入第 k 个挑战, 在第 j 次测量时的响应值; $HD(\bullet)$ 代表计算两个响应序列的汉明距离函数。

2.2.2 独特性

独特性可以通过片间距离(inter-distance)来衡量。片间距离就是使用一个相同的挑战信息作用于不同的 PUF 实体获得的响应之间的汉明距离。按照 2.2.1 中的参数定义, 平均片间距离可按照下式计算:

$$\mu_{inter} = \sum_{\substack{i_1, i_2 = 1 \\ i_1 \neq i_2}}^{N_{puf}} \sum_{k=1}^{N_{chal}} \sum_{j=1}^{N_{meas}} \frac{2 \cdot HD\left(r_{i_1}^j(c_k), r_{i_2}^j(c_k)\right)}{N_{puf} \cdot \left(N_{puf} - 1\right) \cdot N_{chal} \cdot N_{meas}} \tag{3}$$

3 相关工作

3.1 震荡环延迟模型

在 J.Cryptology. 2011 上, Maiti 等人[28]提出了一种延迟模型来对振荡环进行建模分析。振荡环的一个环路延时可以按下式建模:

$$d_{LOOP} = d_{AVG} + d_{RAND} + d_{SYST} \tag{4}$$

其中，d_{AVG} 对于相同结构的振荡环来说是一个相同的延迟数值；d_{RAND} 表示由制造差异导致的随机延迟；d_{SYST} 表示系统差异，系统差异表示同一芯片上的振荡环由位置不同导致的延迟差异。根据(4)式，两个振荡环 a 和 b 之间的环路延时差异可以按照(5)式计算。

$$\begin{aligned}\Delta d_{LOOP} &= d_{LOOPa} - d_{LOOPb} \\ &= \left(d_{AVG} + d_{RANDa} + d_{SYSTa}\right) - \left(d_{AVG} + d_{RANDb} + d_{SYSTb}\right) \\ &= \Delta d_{RAND} + \Delta d_{SYST}\end{aligned} \tag{5}$$

从上式可以看出，1 比特响应 r_{ab} 不仅仅和制造差异 d_{RAND} 有关，还和系统差异 d_{SYST} 有关。Maiti 等人指出，系统差异 d_{SYST} 是由于振荡环的物理位置不同而导致的一种现象：某一局部的振荡环的环路延时普遍高于或低于另外一个局部的振荡环的环路延时。这种系统差异的存在会降低 RO PUF 的独特性。文献[28]中给出了一种应对措施：通过比较尽可能相邻的振荡环频率来提取响应。因为相邻的振荡环具有几乎相同的 d_{SYST}，这样就可以保证(5)式计算得到的环路延时差异由制造差异 Δd_{RAND} 主导。

3.2 可配置延迟线

在 FPGA 平台上，查找表(Look-Up Table, LUT)是主要的可配置延迟单元，图 2 是一个 3 输入 LUT 的示意图。LUT 由 SRAM 单元和树形结构的选通器组成。SRAM 单元用来存储预设功能值，树形结构的选通器用来选择特定 SRAM 单元到输出口。

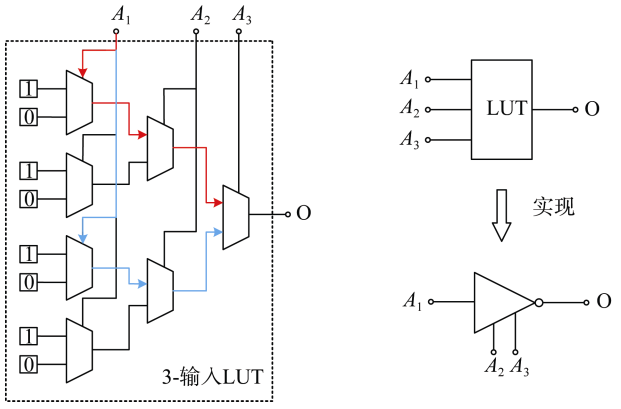

图 2 可配置延迟单元——LUT

Figure 2 Programmable delay lines of LUT

一个 LUT 可以例化成一个反相器，它的输出端 O 是输入端 A_1 的取反值，另外两个输入端 A_2 和 A_3 配置的值对输入端 A_1 和输出端 O 的逻辑关系没有影响。在 CHES 2011 中，Majzoobi 等人[27] 利用 LUT 的延迟配置来实现超细粒度的路径延迟控制。该方案通过改变 LUT 的输入信号来调节路径传播延迟。在图 2 中，虽然输入端 A_2 和 A_3 不会改变这个反相器的逻辑值，但是他们的取值会改变信号从输入端 A_1 到输出端 O 的延迟。Majzoobi 等人指出图 2 中，当 A_2A_3 = 00 时，A_1 到 O 的延时最短，当 A_2A_3 = 11 时，A_1 到 O 的延时最长。

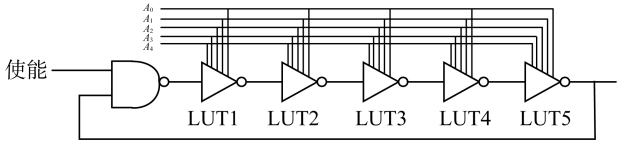

图 3 延迟可配置的五级振荡环示例

Figure 3 Example of 5-stage delay-configurable ring oscillator

在这个结构基础上，Habib 等人[22] 提出了一种基于 4 输入 LUT 的 RO PUF 结构。这种结构的 RO PUF 通过比较两个振荡环在相同 LUT 配置下的频率，可以由两个振荡环得到 8 比特响应，提取到约 5 比特的熵。但是 Habib 的方案有一个非常重要的前提，那就是随着 LUT 输入的变化，不同振荡环的频率变化应该完全不同，使得两个振荡环的频率变化曲线随机交叉，如图(4) 所示。根据 Habib 等人的在 Spartan-3e 上的实验结果来看，这个前提是满足的，该方案的有效性也得到了印证。但是在 Xilinx 最新的基于 6 输入 LUT 的 FPGA 系列(如 Virtex-5, Virtex-6, Kintex-7, Zynq-7000 等)，这个条件已经不再满足。根据我们的实验，对于不同的震荡环，随着 LUT 输入的变化，其频率的变化趋势趋于一致，加之不同振荡环的系统差异较大，使得不同震荡环的频率变化曲线很少有交叉，如图(5)所示(以 Zynq-7000 为例)。也就是说，如果我们知道在 LUT 某个输入下 RO_1 比 RO_2 的频率大，那么在 LUT 其它输入情况下 RO_1 的频率也总是大于 RO_2 的。在这种情况下，Habib 等人的方案便不再有效。

4 RO PUF 建模分析

结合 Majzoobi 等人[26,27]提出的振荡环的延迟模型以及 LUT 可配置延迟模型，本章提出了一种可配置延迟振荡环 PUF 模型。首先，我们根据(6)式来对振荡环 l 的环路延时进行建模。

$$d^j_{LOOP(l)} = d_{AVG} + d^j_{RAND(l)} + d^j_{SYST(l)} \tag{6}$$

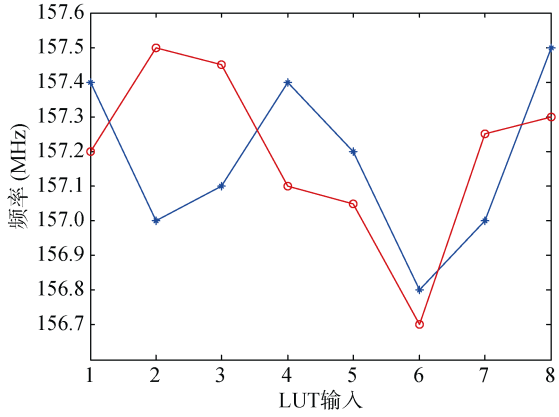

图 4　Habib 方案中不同振荡环在不同 LUT 配置下的频率变化趋势

Figure 4　Frequency's changing pattern of different ring oscillators in Habib's scheme

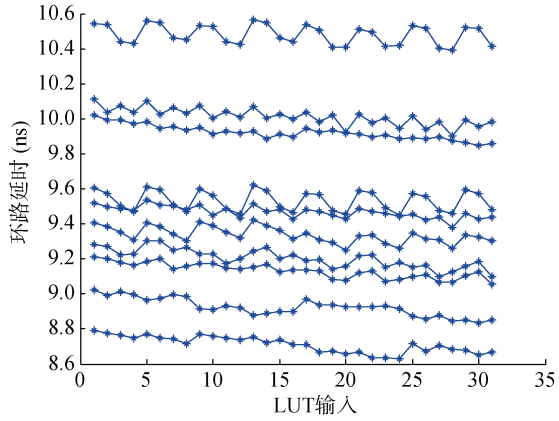

图 5　Zynq-7000 评估板上的不同振荡环在不同 LUT 配置下延时变化趋势示例

Figure 5　Frequency's changing pattern of different ring oscillators on Znyq-7000 evaluation boards

其中，d_{AVG} 是振荡环的理论环路延时，对于所有相同结构的振荡环来说是一样的；$d^{j}_{RAND(l)}$ 表示第 l 个振荡环的 LUT 为配置 j 时，由于制造因素导致的延迟差异；$d^{j}_{SYST(l)}$ 表示系统差异导致的延迟差异。对于一个振荡环，当使用两个不同的 LUT 配置时候，会产生两个系统差异参数 $d^{j_1}_{SYST(l)}$ 和 $d^{j_2}_{SYST(l)}$，由于是同一个振荡环，这两个系统差异参数极其接近，因此在(6)式中，$d^{j_1}_{SYST(l)}$ 和 $d^{j_2}_{SYST(l)}$ 可以简化为 $d_{SYST(l)}$，(6)式就可写成:

$$\mathrm{d}^{j}_{LOOP(l)} = d_{AVG} + d^{j}_{RAND(l)} + d_{SYST(l)} \tag{7}$$

假如总共有 L 个振荡环，当每个振荡环的 LUT 配置都为 j 时，由于制造因素导致的延迟差异就有 L 个值: $d^{j}_{RAND(1)}$，$d^{j}_{RAND(2)}$，…，$d^{j}_{RAND(L)}$。在 HOST 2011 和 ReConFig 2008 中，文献 [14,29] 通过实验表明这些值的分布接近高斯分布。图 6 展示了 3000 个振荡环在不同 LUT 输入下的频率分布图。可见其分布形状也类似一个高斯分布的形状，且对于不同的 LUT 输入，分布的均值略有差异，方差相似。在第 4 小节，将详细描述我们的实验过程。因此，在我们的建模分析中，我们假设所有振荡环在同一 LUT 输入下的延迟服从均值不同，方差相等的高斯分布，如(8)式所示:

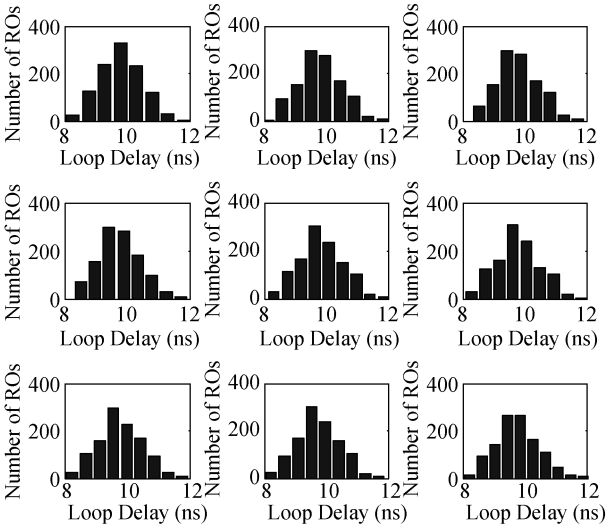

图 6　不同 LUT 输入下 3000 个振荡环的频率分布图

Figure 6　Frequency's distribution of 3000 ring oscillators with different input values of LUT

$$\left(d^{j}_{RAND(1)}, d^{j}_{RAND(2)}, \ldots, d^{j}_{RAND(L)}\right) \sim N\left(\mu_j, \sigma^2\right) \tag{8}$$

也就是说，随机变量 $d_{RAND(j)}$ 服从一个均值分别为 μ_j，标准差为 σ 的正太分布。在 6 输入的 LUT 基础上构建振荡环，有 5 个输入与逻辑无关但可以影响环路延时，因此共有 $2^5 = 32$ 种不同的配置，我们将每种配置下振荡环的制造差异延时分别看作一个随机变量，则共有 32 个服从(8)式正太分布的随机变量。

4.1　二阶差分算法概述

根据上述描述，对于 L 个振荡环，配置信息从“00000”变化到 “11111”，总共有 $32 \times L$ 不同的 $d^{j}_{LOOP(l)}$，可以写成式子(9)。本章首先提出了一种简洁的基于二阶差分算法的方法来提取 PUF 响应。

$$\mathrm{d}^{j}_{LOOP(l)} = d_{AVG} + d^{j}_{RAND(l)} + d_{SYST(l)} (1 \leqslant j \leqslant 32, 1 \leqslant l \leqslant L) \tag{9}$$

基于二阶差分的提取算法可以分为以下两步，以 L 个振荡环为例进行说明:

1) 计算振荡环延时的一阶差分: 对于振荡环 l ，

依次计算相邻配置的延时的差值:

$$\Delta d^{j}_{(1,\ l)}=d^{j}_{LOOP(l)}-d^{j+1}_{LOOP(l)}\ (1\leqslant j\leqslant 31) \tag{10}$$

2) 计算振荡环延时的二阶差分: 对于两个振荡环 A 和 B $\left(A\neq B\right)$, 计算二阶差分值:

$$\Delta d^{j}_{2}=\Delta d^{j}_{(1,\ A)}-\Delta d^{j}_{(1,\ B)} \tag{11}$$

最后根据二阶差分值得到响应 :

$$r^{j}_{2}=\begin{cases}1 & \text{如果}\Delta d^{j}_{2}>0\\ 0 & \text{否则}\end{cases} \tag{12}$$

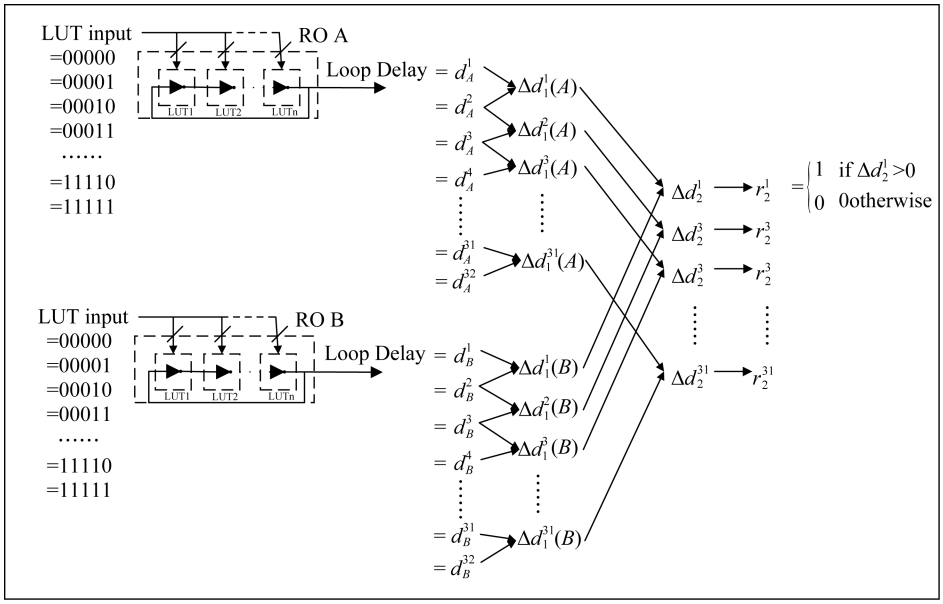

图 7 二阶差分计算示意图

Figure 7 Skeleton diagram of the second-order difference method

经过这两步, 可以从 2 个振荡环中获得 31 比特的响应。基于式子 (9), 一阶差分 $\Delta d^{j}_{(1,\ l)}$ 可以按照下面的式子计算得到。

$$\begin{aligned}\Delta d^{j}_{(1,\ l)}&=d^{j}_{LOOP(l)}-d^{j+1}_{LOOP(l)}\\&=d_{AVG}+d^{j}_{RAND(l)}+d_{SYST(l)}-(d_{AVG}+d^{j}_{RAND(l)}+d_{SYST(l)})\\&=d^{j}_{RAND(l)}-d^{j+1}_{RAND(l)}\end{aligned} \tag{13}$$

从式子(13)中可以看出, 一阶差分计算能够有效消除系统差异因素 $d_{SYST(l)}$ 的影响。根据式(9), $d^{j}_{RAND(l)}$ 和 $d^{j+1}_{RAND(l)}$ 都服从高斯分布:

$$d^{j}_{RAND(l)}\sim N(\mu_j,\sigma^2_j) \tag{14}$$

$$d^{j+1}_{RAND(l)}\sim N(\mu_{j+1},\sigma^2_{j+1}) \tag{15}$$

因为这种基于可配置 LUT 的振荡环中, 其振荡回路其实是多条延迟路径的线性组合, 所以同一振荡环中对于不同的 j, 随机变量 $d^{j}_{RAND(l)}$ 是存在相关性的, 我们假设同一振荡环中任意的两个随机变量 $d^{j}_{RAND(l)}$ 和 $d^{i}_{RAND(l)}(1\leqslant i\neq j\leqslant 32)$ 之间的相关系数是 R, 我们可以获得随机变量 $\Delta d^{j}_{(1,\ l)}$ 的分布如下式所示。

$$\Delta d^{j}_{(1,\ l)}\sim N(\mu_j-\mu_{j+1},2\sigma^2-2\cdot R\cdot\sigma^2) \tag{16}$$

设 μ^{j}_{1} 是 $\mu_j-\mu_{j+1}$, σ^2_1 是 $2\sigma^2-2\cdot R\cdot\sigma^2$。根据随机变量 $\Delta d^{j}_{(1,\ l)}$, 对于不同的振荡环, 我们假设两个随机变量 $\Delta d^{j}_{(1,\ A)}$ 和 $\Delta d^{j}_{(1,\ B)}$ 之间没有相关性, 因此经过第二步二阶差分的计算之后, 我们可以获得随机变量 Δd^{j}_{2} 的分布如下。

$$\Delta d^{j}_{2}\sim N(0,2\cdot\sigma^2_1)\quad (1\leqslant j\leqslant 31) \tag{17}$$

根据式 (17), d^{j}_{2} 大于零和小于零的概率相等, 所以 r^{j}_{2} 取 “0” 或者 “1” 的概率相等, 都是 50%。

4.2 二阶差分算法分析

从两个振荡环提取更多随机熵的关键就是如何提取更多的制造差异, 而这些制造差异的数量级往往很小, 甚至小到接近噪声的。因此, 提取更多熵的方法需要尽最大限度地减小噪声的影响。

在传统的 RO PUF 的架构中, 一个 LUT 仅仅被例化成一个反相器, 这样的反相器只有一个传播路径。在可配置延迟线模型中, 当 LUT 被例化成反相器的时候, 对于 LUT 输入的精细控制使得反相

器具有多个传播延迟。Habib 等人[22] 想通过控制 LUT 的配置信息来提取更多的响应。然而在 Xilinx Virtex-5, 6, 7 系列 FPGA 中, 由于系统差异较大且 LUT 不同配置信息导致的不同传播延迟变化趋势趋于一致, 在这些 FPGA 平台上比较两个振荡环的不同配置信息下的传播延迟, 虽然可以获得 32 比特的响应, 但是能够提取到的熵却远小于 32 比特。

在二阶差分算法中, 将相同的振荡环的不同延迟配置的延迟做一阶差分计算, 计算结果可以很大程度上消除系统差异导致的负面影响。因为同一个振荡环在不同配置下, 其系统差异是非常接近的。一阶差分的计算结果可以看成是进行比较的两个传播路径的制造差异组合, 二阶差分计算的结果就表示两个振荡环之间的这个制造差异组合的差异。按照第 4.1 节中二阶差分算法概述的两步计算, 响应由(16) 式的正负号决定。A, B 表示不同振荡环的序号, j 表示 LUT 的配置。

$$\left(d^{j}_{LOOP(A)}-d^{j+1}_{LOOP(A)}\right)-\left(d^{j}_{LOOP(B)}-d^{j+1}_{LOOP(B)}\right) \quad (18)$$

式(16)可以改写成下面这个形式:

$$\left(d^{j}_{LOOP(A)}-d^{j}_{LOOP(B)}\right)-\left(d^{j+1}_{LOOP(A)}-d^{j+1}_{LOOP(B)}\right) \quad (19)$$

从式子(19)中可以看出二阶差分算法本质上是提取 LUT 配置为 j 时振荡环 A、B 的制造差异与 LUT 配置为 $j+1$ 时振荡环 A、B 的制造差异的差异, 也就是制造差异的差异。从图(8)中可以看出, 不同振荡环对的二阶差分值的变化与图(5)中的不同振荡环的延时变化相比更加散乱随机, 因此可以比 Habib 等人提出的方法提取更多的熵。

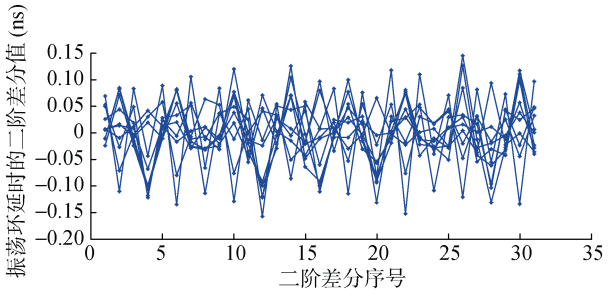

图 8 Virtex-6 上不同振荡环延时的二阶差分变化趋势
Figure 8 Changing patterns of the second-order difference of different ring oscillators on Virtex-6

除此之外二阶差分算法还能从一定程度上增强 PUF 响应的可靠性。为了降低环境变化对响应可靠性的负面影响, Gassend[30]提出了通过比较两个振荡环的频率来产生 1 比特响应的方法。二阶差分算法不仅继承了这种方法的优点, 降低环境变化带来的影响, 还能很有效的消除系统差异导致的负面作用。

4.3 高阶差分方法

我们通过二阶差分的方法可以提取两个振荡环更细粒度的制造差异, 那推广到更高阶差分是否可以提取更多的差异呢? 将二阶差分方法推广到高阶差分, 有两个问题必须解决: 第一个是同一阶响应相邻比特的相关性问题; 第二个问题是上一阶的响应会泄露下一阶响应的熵。

4.3.1 消除相邻响应比特的相关性

为了简化说明我们假设有两个相同的有三种环路延时的振荡环, 按照二阶差分算法我们可以从这两个振荡环提取 2 比特的响应 r_2^1 和 r_2^2。根据(11)式:

$$\begin{aligned}\Delta d_2^1 &= \Delta d^1_{(1,\ A)}-\Delta d^1_{(1,\ B)}\\ &=\left(d^1_{LOOP(A)}-d^2_{LOOP(A)}\right)-\left(d^1_{LOOP(B)}-d^2_{LOOP(B)}\right)\\ &=\left(d^1_{LOOP(A)}-d^1_{LOOP(B)}\right)-\left(d^2_{LOOP(A)}-d^2_{LOOP(B)}\right)\\ &=\Delta d_1-\Delta d_2\end{aligned} \quad (20)$$

同样地有:

$$\Delta d_2^2=\Delta d_2-\Delta d_3 \quad (21)$$

忽略 Δd_1, Δd_2, Δd_3 相等的情况, 结合(12)式, 我们来讨论 $r_2^1r_2^2$ 为不同取值时, Δd_1, Δd_2, Δd_3 的大小所有可能的组合情况。

当 $r_2^1r_2^2=00$ 时, 有 $\Delta d_1<\Delta d_2$ 且 $\Delta d_2<\Delta d_3$, 此时的大小排列只有(小, 中, 大)一种可能;

当 $r_2^1r_2^2=11$ 时, 有 $\Delta d_1>\Delta d_2$ 且 $\Delta d_2>\Delta d_3$, 此时 $\left(\Delta d_1,\Delta d_2,\Delta d_3\right)$ 的大小排列只有(大, 中, 小)一种可能;

当 $r_2^1r_2^2=10$ 时, 有 $\Delta d_1>\Delta d_2$ 且 $\Delta d_2<\Delta d_3$, 此时 $\left(\Delta d_1,\Delta d_2,\Delta d_3\right)$ 的大小排列可能有(大, 小, 中)和(中, 小, 大)两种可能;

当 $r_2^1r_2^2=01$ 时, 有 $\Delta d_1<\Delta d_2$ 且 $\Delta d_2>\Delta d_3$, 此时 $\left(\Delta d_1,\Delta d_2,\Delta d_3\right)$ 的大小排列可能有(小, 大, 中)和(中, 大, 小)两种可能。

因此 $r_2^1r_2^2$ 为 00, 01, 10, 11 的概率分别为 1/6, 1/3, 1/3, 1/6。由 4.1 的分析知, 没有任何前提条件的情况下, r_2^2 为 0 和为 1 的概率是相等的:

$$\mathrm{P}\left(r_2^2=0\right)=\mathrm{P}\left(r_2^2=1\right)=\frac{1}{2},$$

但如果知道了 r_2^1 的值, 则概率变为:

$$\mathrm{P}\left(r_2^2=0\,|\,r_2^1=0\right)=\frac{1}{3} \text{ 和 } \mathrm{P}\left(r_2^2=1\,|\,r_2^1=0\right)=\frac{2}{3},$$

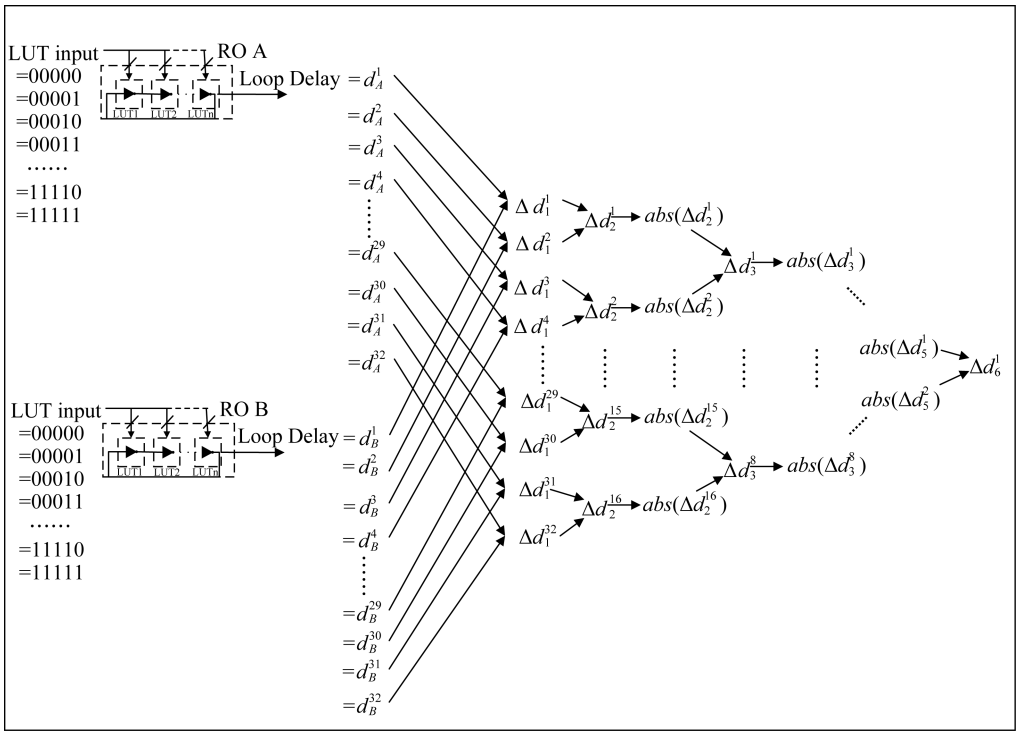

图 9　高阶差分计算示意图

Figure 9　Skeleton diagram of the proposed high-order difference method

显然相邻响应比特 r_2^1 和 r_2^2 是相关的, 响应 $r_2^1 r_2^2$ 的香农熵:

$$H(r_2^1 r_2^2) = -2\cdot\frac{1}{3}\log_2\frac{1}{3} - 2\cdot\frac{1}{6}\log_2\frac{1}{6} \approx 1.92 < 2,$$

因此按照 4.1 所述二阶差分算法得到的 31 比特响应, 其香农熵低于 31 比特。

从直观上我们很容易理解这种相关性的来源。$r_2^1 = 1$ 说明 $\Delta d_1 > \Delta d_2$, 从直觉上这意味着 Δd_2 很可能是一个比较小的值, 所以使得 $\Delta d_2 < \Delta d_3$ 的概率更有可能大于 $\Delta d_2 > \Delta d_3$ 的概率, 反之亦然。这种直观的推理与我们上面具体的数值分析也是相一致的。正是因为计算二阶相邻差分值 Δd_2^1 和 Δd_2^2 都用到了同一个因子 Δd_2, 所以无论知道了 r_2^1 还是 r_2^2, 都多少泄露了另一个响应比特的信息。所以为了消除这种相关性, 同一个因子就不能用于计算两个或两个以上的响应比特。

4.3.2　消除邻阶响应的信息泄露

显然, 我们不能像(22)式那样直接将 4.1 的二阶差分方法推广到高阶差分计算, 这会使得相邻两阶的响应泄露对方的信息。

$$\Delta d_{i+1}^j = \Delta d_i^j - \Delta d_i^{j+1} \quad (1 \leqslant i \leqslant 31, 1 \leqslant j \leqslant 33-1) \tag{22}$$

最明显地, 当上一阶两相邻差分值, 如 Δd_i^j 和 Δd_i^{j+1} 的正负相反时, 不妨设 $\Delta d_i^j > 0$且$\Delta d_i^{(j+1)} < 0$, 这其实也就是已知 i 阶相邻两比特响应 $r_i^j = 1$ 且 $r_i^{j+1} = 0$, 我们可以立刻推理得到第 i+1阶第 j 个差分值 Δd_{i+1}^j=$\Delta d_i^j - \Delta d_i^{j+1} > 0$, 从而得知第 i+1阶第 j 个响应比特 $r_{i+1}^j = 1$。

类似的, 如果已知 $r_i^j = 0$ 且 $r_i^{j+1} = 1$, 则可推断一定有 $r_{i+1}^j = 0$。为了消除这种形式的信息泄露, 将差分方法推广到高阶, 我们在计算下一阶的差分值时, 将当前的差分值全部取绝对值, 从而使得 Δd_i^j 和 Δd_i^{j+1}的符号一致:

$$\Delta d_{i+1}^j = \mathrm{abs}\left(\Delta d_i^j\right) - \mathrm{abs}\left(\Delta d_i^{j+1}\right) \tag{23}$$

其中 abs(·)表示取绝对值运算。再结合之前消除相邻比特相关性的方法, 我们提出了新的利用高阶差分方法计算响应的方案如下:

1) 对于两个振荡环 A 和 B $(A \neq B)$, 先计算它们的 1 一阶差分值:

$$\Delta d_1^j = d_{LOOP(A)}^j - d_{LOOP(B)}^j, (1 \leqslant j \leqslant 32) \tag{24}$$

2) 计算二阶差分:

$$\Delta d_2^j = \Delta d_1^{(2(j-1)+1)} - \Delta d_1^{2j} \tag{25}$$

3) 计算其 3 到 6 阶差分值:

$$\Delta d_i^j = \mathrm{abs}(\Delta d_{(i-1)}^{(2(j-1)+1)}) - \mathrm{abs}(\Delta d_{(i-1)}^{2j}) \tag{26}$$

4) 计算 2 阶到 6 阶响应值:

$$r_i^j = \begin{cases} 1 & \text{如果} \Delta d_i^j > 0 \\ 0 & \text{否则} \end{cases} \tag{27}$$

其中, $2 \leqslant i \leqslant 6$, $1 \leqslant j \leqslant 64/2^i$。最后我们将每一阶的响应值合在一块作为最终的响应值, 共计 31 比特: $r = \left\{ r_2^1,\ r_2^2,\ldots,\ r_2^{16},\ r_3^1,\ r_3^2,\ldots,\ r_6^1 \right\}$。

4.4 高阶差分算法分析

根据 4.1 的分析和(17)式我们已知二阶差分 Δd_2^j 是同分布的, 且 d_2^j 大于零和小于零的概率相等, 即 $\mathrm{P}\left(r_2^j = 0\right) = \mathrm{P}\left(r_2^j = 1\right) = 0.5$, 再加上新的高阶差分计算过程中, 我们消除了各阶差分相邻比特的相关性, 因此可以认为二阶差分变量 Δd_2^j 是独立同分布的, 即 16 比特的二阶响具有 16 比特的熵。同样的, 要证明高阶响应理论上满熵, 需要证明高阶差分变量 Δd_i^j 是同分布的, 且在知道其他阶响应的前提下, Δd_i^j 大于零和小于零的概率应该相等。

因为已知 $\Delta d_2^j (1 \leqslant j \leqslant 16)$ 是同分布的, 所以要证明 $\Delta d_i^j (3 \leqslant i \leqslant 6, 1 \leqslant j \leqslant 64/2^i)$ 同分布, 则只需证明对于任意同分布的随机变量, 它们的差也是同分布的。为此, 设有四个同分布的随机变量 X_1, X_2, X_3, X_4, 令 $Z=X_1 - X_2$, 则 Z 的分布有:

$$f_Z(z) = \int_{-\infty}^{\infty} f_{X_1}(z+x) f_{X_2}(x) \mathrm{d}x \tag{28}$$

因为 $f_{X_1}, f_{X_2}, f_{X_3}, f_{X_4}$ 是相同的, 所以:

$$f_{X_1}(z+x) = f_{X_3}(z+x)$$

$$f_{X_2}(x) = f_{X_4}(x)$$

代入(28)式得:

$$f_Z(z) = f_{X_1 - X_2}(z)$$

$$= \int_{-\infty}^{\infty} f_{X_1}(z+x) f_{X_2}(x) \mathrm{d}x$$

$$= \int_{-\infty}^{\infty} f_{X_3}(z+x) f_{X_4}(x) \mathrm{d}x = f_{X_3 - X_4}(z)$$

所以 $X_1 - X_2$ 和 $X_3 - X_4$ 是独立同分布的。

在给定上一阶响应两相邻比特 $r_i^{2(j-1)+1}$ 和 $r_i^{2j} \left(2 \leqslant i \leqslant 5, 1 \leqslant j \leqslant 64/2^i\right)$ 的前提下, 可以知道的信息只是 $\Delta d_i^{2(j-1)+1}$ 和 Δd_i^{2j} 是大于零还是小于零, 却不能推断 $\mathrm{abs}(\Delta d_i^{2(j-1)+1}) > \mathrm{abs}(\Delta d_i^{2j})$ 还是 $\mathrm{abs}(\Delta d_i^{2(j-1)+1}) < \mathrm{abs}(\Delta d_i^{2j})$; 同理在知道下一阶响应某一比特的值 $r_i^j \left(3 \leqslant i \leqslant 6, 1 \leqslant j \leqslant 64/2^i\right)$, 只能知道上一阶两个差分绝对值的大小关系, 但无法推知上一阶的两个差分值 $d_i^{2(j-1)+1}$ 和 d_i^{2j} 的正负。因此 Δd_i^j 大于零和小于零的概率只取决于它自身的分布, 而与是否知道其他响应比特无关, 即 $\Delta d_i^j \left(2 \leqslant i \leqslant 6, 1 \leqslant j \leqslant 64/2^i\right)$ 不仅是同分布的, 而且是独立的。所以要证明高阶响应满熵, 我们只需证明:

$$\mathrm{P}\left(\mathrm{abs}\left(\Delta d_i^{2(j-1)+1}\right) > \mathrm{abs}\left(\Delta d_i^{2j}\right)\right)$$

$$= \mathrm{P}\left(\mathrm{abs}\left(\Delta d_i^{2(j-1)+1}\right) < \mathrm{abs}\left(\Delta d_i^{2j}\right)\right)$$

$$= 0.5$$

因为 $d_i^{2(j-1)+1}$ 和 d_i^{2j} 是独立同分布的, 所以 $\mathrm{abs}(\Delta d_i^{2(j-1)+1})$ 和 $\mathrm{abs}(\Delta d_i^{2j})$ 也是独立同分布, 我们令 $X = \mathrm{abs}(\Delta d_i^{2(j-1)+1})$, $Y = \mathrm{abs}(\Delta d_i^{2j})$, 则 $X > Y$ 的概率:

$$\mathrm{P}(X > Y) = \int_{-\infty}^{\infty} f_X(x > k) f_Y(y = k) \mathrm{d}k \tag{29}$$

因为 X 和 Y 同分布, 所以有:

$f_X(x > k) = f_Y(y > k)$ 和 $f_Y(y = k) = f_X(x = k)$

代入(29)式, 有:

$$\mathrm{P}(X > Y) = \int_{-\infty}^{\infty} f_X(x > k) f_Y(y = k) \mathrm{d}k$$

$$= \int_{-\infty}^{\infty} f_Y(y > k) f_X(x = k) \mathrm{d}k$$

$$= \mathrm{P}(Y > X)$$

因为 $\mathrm{P}(X=Y)$ 很小, 所以忽略 $X=Y$ 的概率, 有:

$$\mathrm{P}(X > Y) + \mathrm{P}(Y > X) = 1$$

所以 $\mathrm{P}\left(\mathrm{abs}\left(\Delta d_i^{2(j-1)+1}\right) > \mathrm{abs}\left(\Delta d_i^{2j}\right)\right)=0.5$ 且 $\mathrm{P}\left(\mathrm{abs}\left(\Delta d_i^{2(j-1)+1}\right) < \mathrm{abs}\left(\Delta d_i^{2j}\right)\right)=0.5$, 也就是说采用高阶差分算法得到的 31 比特响应 $r = \{r_2^1,\ r_2^2,\ldots,\ r_2^{16},\ r_3^1,\ r_3^2,\ldots,\ r_6^1\}$ 应该具有 31 比特的熵。

5 评估实验

5.1 仿真

为了验证整体方案的正确性和有效性, 我们在

4.1 中提到的延时模型的基础上进行了一系列的仿真和实验。需要指出的是仿真参数的采集和实验均是在约 27℃下的室温进行。

所谓的制造差异一般可以分成系统差异和随机制造差异。系统差异主要受到芯片上的位置的影响。例如, 在 RO PUF 的架构中, 系统差异导致的是一个区域的振荡环的频率平均要比另一个区域的要高, 我们设系统差异造成的环路延时差异 d_{SYST} 服从高斯分布:

$$d_{SYST} \sim N(\mu_{SYST},\ \sigma^2_{SYST}) \tag{30}$$

而随机制造差异则和振荡环的空间位置无关, 如 4 中分析, 我们假设所有振荡环在 LUT 相同配置下的延迟服从均值不同, 方差相等的高斯分布, 如(8)式所示。

除此之外, 因为电路噪声的影响, 每次测量同一振荡环在同一LUT输入下的计数值也是不相同的, 我们设电路噪声服从高斯分布:

$$d_{NOISE} \sim N(\mu_{NOISE},\ \sigma^2_{NOISE}) \tag{31}$$

仿真所需要的这些参数值都是通过在 Zynq-7000 系列评估板进行实验并统计得到的, 获取的参数值在表 1 中列出。

表 1　实验获取的参数列表

Table 1 Parameters acquired from experiment

参数	数值 (单位: ns)
理论环路延时 d_{AVG}	9.70
系统差异的均值 μ_{SYST}	$-2.47^{-17} \approx 0$
系统差异的方差 σ_{SYST}	0.5422
电路噪声均值 μ_{NOISE}	$1.91^{-17} \approx 0$
电路噪声方差 σ_{NOISE}	0.0314
随机制造差异均值 μ_{RAND}	−1.329 ~ 1.2016
随机制造差异方差 σ_{RAND}	0.6488

下面将用一个示例程序来说明我们的模拟生成不同振荡环实体环路延时并生成响应比特串的过程。首先, 我们需要设置几个参数:

表 2　设置参数列表

Table 2　Parameter settings

参数	数值
振荡环数目 N_{RO}	1680
LUT 输入的比特数 N_{IN}	5
同一响应重复测量次数 N_{MEAS}	200

代入参数, 通过下面这个程序, 我们便可以模拟生成用于仿真分析的振荡环环路延时 $d_{(i,j,k)}$, 并根据提出的高阶差分方法计算得到各阶响应比特 r^j_{order}, 其中$1 \leqslant i \leqslant N_{RO}$,$1 \leqslant j \leqslant N_{IN}$,$1 \leqslant k \leqslant N_{MEAS}$, $2 \leqslant order \leqslant 6$。

For i=1 to N_{RO} **do**
  $d_{SYS(i)} \leftarrow N(\mu_{SYS}, \sigma^2_{SYS})$
  For j=1 to N_{IN} **do**
    $d_{PROD(i,j)} \leftarrow N(\mu_{RAND(j)}, \sigma^2_{RAND})$
    For k=1 to N_{MEAS} **do**
      $d_{NOISE(i,j,k)} \leftarrow N(\mu_{NOISE}, \sigma^2_{NOISE})$
    End for
  End for
End for
For i=1 to N_{RO}/Z **do**
For j=1 to N_{IN} **do**
$\Delta d_1^j = (d_{AVG} + d_{SYS(2i-1)} + d_{RAND(2i-1,j)} + d_{NOISE(2i-1,j)}) - (d_{AVG} + d_{SYS(2i)} + d_{RAND(2i,j)} + d_{NOISE(2i,j)})$
End for
End for
For order=2 to 6 **do**
  For j=1 to $\frac{64}{2^{order-1}} - 1$ **do**
    $\Delta d^j_{order} = \mathrm{abs}(\Delta d^{2(j-1)+1}_{order-1}) - \mathrm{abs}(\Delta d^{2j}_{order-1})$
    If $\Delta d^j_{order} > 0$ **do**
      $r^j_{order} = 1$
    else
      $r^j_{order} = 0$
    End if
  End for
End for

5.2　实验

我们在 20 块 Virtex-6 和 30 块 Zynq-7000 系列 FPGA 评估板上分别进行了实验。这两款 FPGA 尤其是 Zynq-7000 系列, 均是 Xilinx 较新的商用 FPGA, 且因为 3.2 中所述原因, Habib 的方案在这两个系列的 FPGA 上均不能有效地提取熵。

我们在每一块 FPGA 上分别例化了 56 个振荡环。每一个振荡环包含 15 个反相器和一个作为使能开关的与门, 其理论环路延时为 9.70ns。所有的反相器和与门都是由 6 输入的 LUT 实现的。图(10)展示了我们整个实验数据采集系统的结构。每一个振荡环都有一个对应的计数器与参考计数器同步进行计数。所有的

计数结果都会通过 UART 发送给电脑进行进一步的处理和分析。方便起见, 所有的响应计算和分析过程都是在电脑上离线处理, 硬件只是负责采集数据。

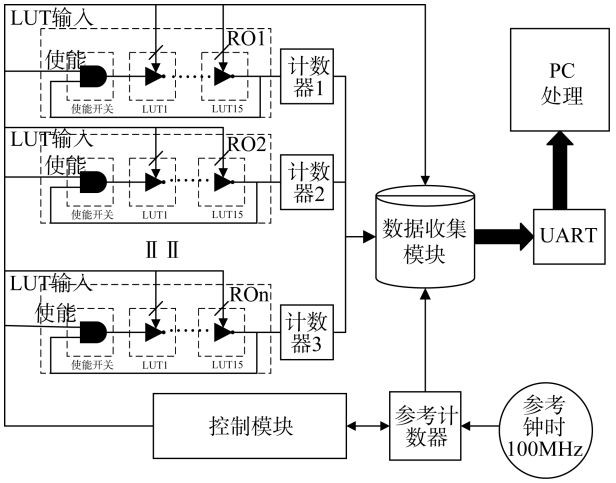

图 10 实验系统架构

Figure 10 Experimental structure

为了尽可能排除其他干扰使得制造差异来主导振荡环的环路延时差别, 我们利用 Xilinx 开发套件提供的 XDC Macro 技术, 来保证例化的振荡环的结构和布线完全一致。XDC Macro 是一个物理约束对象, 可以在电路实现阶段进行相对布局布线。通过这一技术, 我们轻松实现了 56 个振荡环中 15 个反相器和一个与门的相对布局。在 Xilinx 的 FPGA 中有两种逻辑单元 SLICEL 和 SLICEM, 其区别在于 SLICEM 型 LUT 可以被用作分布式 RAM 和移位寄存器。为了保证所有振荡环都是一样的, 我们将振荡环都用 SLICEL 型 LUT 实现。我们使用的 FPGA 中每个单元有 4 个 LUT, 每个振荡环需要 16 个 LUT, 占 4 个单元。图(11)是我们进行布局时的截图。表 3 列出了我们的硬件占用情况。

图 11 振荡环在 FPGA 上的实现和布局

Figure 11 Layout of ring oscillators on FPGA

表 3 硬件资源消耗情况

Table 3 Hardware consumption

逻辑类型	占用的逻辑单元个数	占用百分比
振荡环	4*56=224	12.36%
控制和 UART	839	46.28%
共计	1063	58.64%

5.3 仿真和实验结果

利用表 1 中实验采集得到的参数, 我们仿真生成了 840 对振荡环, 我们假设对每个振荡环测量了 200 次, 然后根据提出的高阶差分算法来计算响应序列。我们将一对振荡环看做一个 FROPUF 实体, 也就是令 N_{puf} =80, 所以对于这个 PUF 只有一个挑战 $N_{chal}=1$。将设置的参数和仿真结果分别代入(1)式和(2)式, 我们得到平均片内距离是 4.46%, 平均片间距离是 50.08%。片内和片间距离的分布分别如图(12)和图(13)所示:

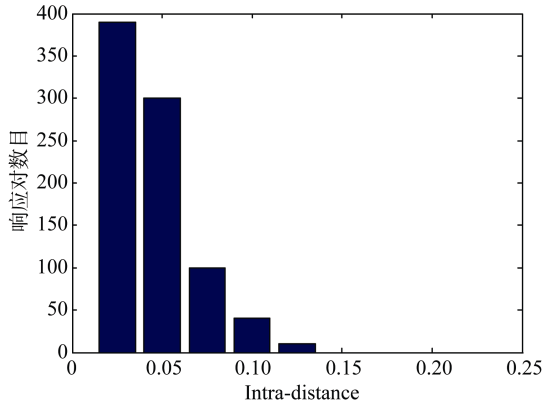

图 12 根据 Zynq-7000 评估板上获取参数仿真所得片内距离分布图

Figure 12 Intra-distance distribution of simulation data based on parameters acquired on Zynq-7000

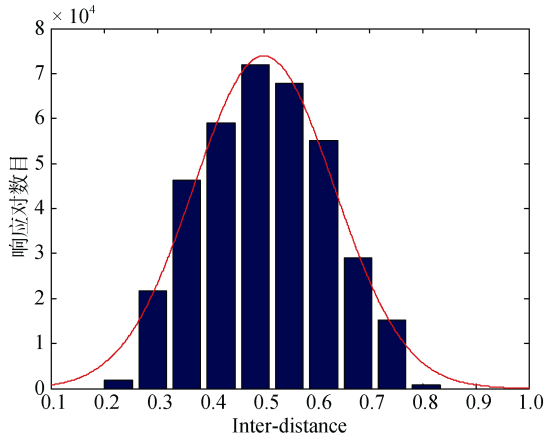

图 13 根据 Zynq-7000 评估板上获取参数仿真所得片间距离分布图

Figure 13 Inter-distance distribution of simulation data based on parameters acquired on Zynq-7000

根据 Zynq-7000 系列评估板上的实验结果，我们得到平均片内距离为 5.10%，平均片间距离为 49.92%。片内和片间距离的分布分别如图(14)和图(15)所示，其中红色曲线为仿真结果的拟合，可以看到实验结果和仿真结果基本一致，从而验证了所提出的模型的正确性。

同时作为比较，我们在 Virtex-6 系列评估板上进行实验得到的平均片内距离为 8.39%，平均片间距离为 49.40%。其分布图如图(16)和(17)所示。

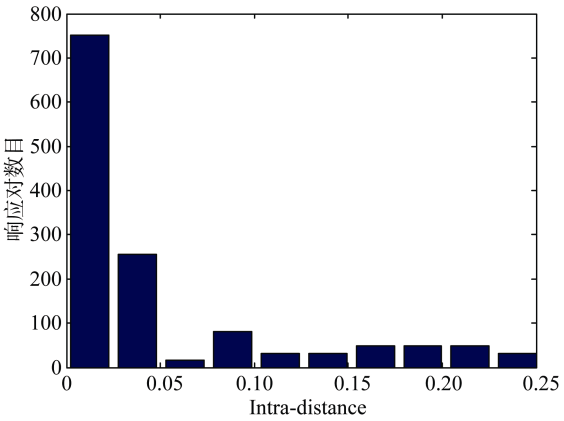

图 14　Zynq-7000 评估板上实验所得片内距离分布

Figure 14　Intra-distance distribution of experimental data acquired on Zynq-7000

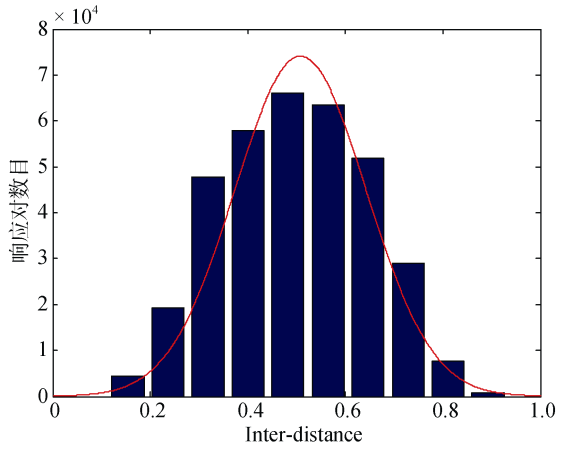

图 15　Zynq-7000 评估板上实验所得片间距离分布

Figure 15　Inter-distance distribution of experimental data acquired on Zynq-7000

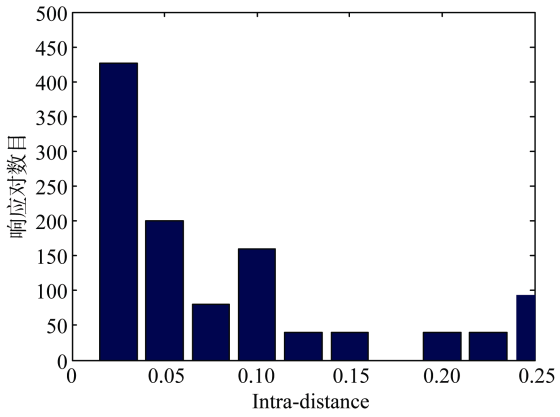

图 16　Virtex-6 评估板上实验所得片内距离分布

Figure 16　Intra-distance distribution of experimental data acquired on Virtex-6

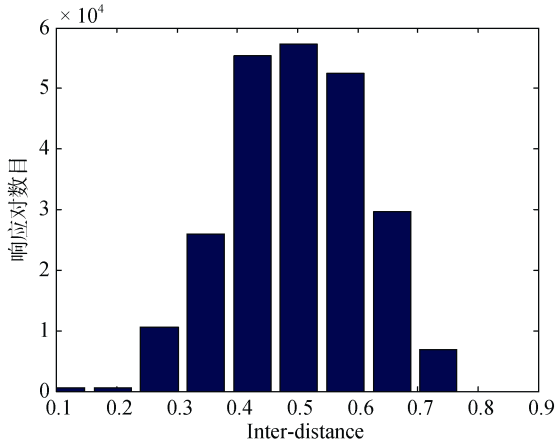

图 17　Virtex-6 评估板上实验所得片间距离分布

Figure 17　Inter-distance distribution of experimental data acquired on Virtex-6

通过比较我们看到，所提出的高阶差分方法在 Xilinx 不同系列的 FPGA 上得到的结果差别并不是很大。Virtex-6 系列上得到的响应序列的稳定性相比 Zynq-7000 系列的差一些，但随机性基本相当。

6　进一步讨论

基于实验数据，我们还统计了实验所得响应序列中每个比特为 0 和为 1 的概率，并代入(31)式计算得到响应的香农熵为 30.85 比特

表 4　与其他 RO PUF 结构的性能比较

Table 4　Performance comparison with other RO PUFs

	振荡环的数目	独立的比特数	响应的熵(bit/RO)	平均片间距离	平均片内距离
本文方案	2	31	16.5	49.92%	5.10%
Habib 提出的方案[22]	130	318	2.44	48.30%	2.12%
Maiti 提出的方案[28]	512	411	≈1	47.31%	≈0%
Suh 和 Devdas 方案[15]	128	128	1	46.15%	0.48%

$$\mathrm{H}(\mathrm{r})=\sum_{i=1}^{31}\left[-\mathrm{P}(r_i=0)\log_2\mathrm{P}(r_i=0)-\mathrm{P}(r_i=1)\log_2\mathrm{P}(r_i=1)\right] \tag{32}$$

与现有的其他方案相比, 我们的方案大大提高了从单位振荡环中提取熵的效率, 具体对比情况如表 4 所示。

观察表 4 会发现, 我们的方案相比其他方案的平均片内距离较高, 其原因可以由图(18)加以说明。图(18)分别统计了各阶响应的平均片内距离, 可见平均片内距离几乎随响应的阶数呈线性增长。这是因为随着差分阶数越来越高, 我们所提取的环路延时的差异的粒度越来越细, 这也意味着制造差异的数量级越来越小, 以至于电路噪声的影响逐渐占了主导。所以, 通过这种高阶差分的方法来计算 RO PUF 的响应也不是阶数越高越好。当能够提取到的制造差异粒度与电路噪声相当时, 得到的响应比特更多的是随机的电路噪声的反映, 所以稳定性难以保证。

不过就我们当前的方案来说, 通过结合 6 阶差分结果得到的响应序列, 其平均片内距离都低于 10%, 这在目前已有的纠错技术下是完全能够接受的。如 R. Maes 在文献[16] 中提出的软决策纠错方案, 就能够在较小的硬件消耗下以低于10^{-6}的失败率纠正平均比特错误率为 15%的 PUF 响应序列中所有的出错比特位。

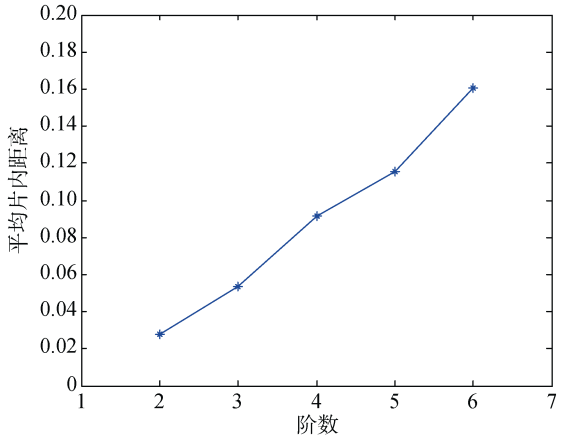

图 18　平均片内距离与差分阶数的关系
Figure 18　The relationship between average intra-distance and differential orders

7　结论

本文基于 LUT 的细粒度控制提出了一种新型的基于振荡环的物理不可克隆方法来提取更多的制造差异。同时, 利用高阶差分提取计算方法, 可以很巧妙地降低系统差异和环境波动带来的负面影响。根据 Zynq-7000 评估板上的实验结果, 在 27℃环境下, 从两个振荡环中提取到的 31 比特响应的平均香农熵为 30.85 比特, 平均片间距离为 49.92%, 平均片内距离为 5.10%, 证明了这种高阶差分算法提取熵的有效性。同时, 仿真结果与实验结果的一致性也证明了我们所提出的模型和高阶差分算法的正确性。

参考文献

[1] Pol Van Aubel, Daniel J. Bernstein, and Ruben Niederhagen. “ Investigating SRAM PUFs in large CPUs and GPUs”. In *5th International Conference on Security, Privacy, and Applied Cryptography Engineering* (SPACE), pp. 228–247, 2015.

[2] Mafalda Cortez, Said Hamdioui, and Ryoichi Ishihara. “Design dependent SRAM PUF robustness analysis”. In *16th Latin-American Test Symposium* (LATS), pp. 1–6, 2015.

[3] Takanori Machida, Dai Yamamoto, Mitsugu Iwamoto, and Kazuo Sakiyama. “Implementation of double arbiter PUF and its performance evaluation on FPGA”. In *20th Asia and South Pacific Design Automation Conference* (ASP-DAC), pp. 6–7, 2015.

[4] Cheng Wei Lin and Swaroop Ghosh. “A family of Schmitt- Trigger-based arbiter-PUFs and selective challenge-pruning for robustness and quality”. In *IEEE International Symposium on Hardware Oriented Security and Trust* (HOST), 2015.

[5] Kumar S., Guajardo J., Maes R., Schrijen G., and Tuyls P. The Butterfly PUF Protecting IP on Every FPGA. In IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2008.

[6] Anderson J. “A PUF Design for Secure FPGA-based Embedded Systems”. In *Asia and South-Pacific Design Automation Conference* (ASP-DAC), pp. 1–6, 2010.

[7] Shimizu K., Suzuki D., and Kasuya T. “Glitch PUF:Extracting Information from Usually Unwanted Glitches”. In *IEICE Transactions on Fundamentals of Electronics*, Communications and Computer Sciences, 2012.

[8] Filip Kodytek and Robert Lorencz. “Proposal and Properties of Ring Oscillator-Based PUF on FPGA”. *Journal of Circuits, Systems, and Computers*, 25, 2016.

[9] Yuan Cao, Le Zhang, Chip-Hong Chang, and Shoushun Chen. “A Low Power Hybrid RO PUF with Improved Thermal Stability for Lightweight Applications”. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34:1143–1147, 2015.

[10] Robert Lorencz Filip Kodytek. “A Design of Ring Oscillator Based PUF on FPGA”. In *18th IEEE International Symposium on Design and Diagnostics of Electronic Circuits & Systems* (DDECS), 2015.

[11] Md. Tauhidur Rahman, Domenic Forte, Fahim Rahman, and Mark Tehranipoor. “A pair selection algorithm for robust RO-PUF against environmental variations and aging”. In *IEEE International*

Conference on Computer Design (ICCD), 2015.

[12] Phuong Ha Nguyen and Durga Prasad Sahoo and Rajat Subhra Chakraborty and Debdeep Mukhopadhyay. “Efficient attacks on robust ring oscillator PUF with enhanced challenge-response”. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition* (DATE), pp. 641–646, 2015.

[13] Chi-En Daniel Yin and Gang Qu. “LISA: Maximizing RO PUF’s Secret Extraction”. In Proceedings of *the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust* (HOST), pages 100–105, 2010.

[14] Abhranil Maiti, Jeff Casarona, Luke McHale, and Patrick Schaumont. “A Large Scale Characterization of RO-PUF”. In Proceedings of *the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust* (HOST), 2010.

[15] G.E. Suh, S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation”, *Asia and South Pacific Design Automation Conference* (ASP-DAC), pp. 9-14, 2007.

[16] R. Maes, “Physically Unclonable Functions: Constructions, Properties and Applications”, Springer Publishing Company, Incorporated, 2013.

[17] R.V.D. Berg, “Entropy analysis of Physical Unclonable Functions”, white paper, available on line at: http://alexandria.tue.nl/extra1/afstversl/wsk-i/vandenberg2012.pdf

[18] R.V.D Berg, B. Skoric, V.V.D. Leest, “Bias-based modeling and entropy analysis of PUFs”, *Acm Trusted*, 2013, pp. 13-20.

[19] P. Koeberl, J. Li, A. Rajan, W Wu, “Entropy Loss in PUF-based Key Generation Schemes: The Repetition Code Pitfall”, IEEE 2014, pp. 44-49, 2014.

[20] H. Kang, Y. Katashita, M. Hagiwara, K. Lwamura, “Performance A nalysis for PUF Data Using Fuzzy Extractor”, Lecture Notes in Electrical Engineering, pp.277-284, 2014.

[21] D. Merli, D. Schuster, F. Stumpf, G. Sigl, “Side-Channel Analysis of PUFs and Fuzzy Extractors”, Lecture Note in Computer Science, pp. 33-47, 2011.

[22] B. Habib, K.Gaj, J.P. Kaps, “FPGA PUF Based on Programmable LUT Delays”, *Euromicro Conference on Digital System Design* (DSD), pp. 697-704, 2013.

[23] Q. Chen, G. Csaba, P. Lugli, U Schichtmann, U. Ruhrmair, “The Bistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions”, IEEE International Symposium on *Hardware-oriented Security & Trust*, pp. 134-141, 2011.

[24] D.E. Holcomb, K. Fu, “Bitline PUF: Building Native Challenge-Response PUF Capability into Any SRAM”, Springer Berlin Heidelberg, pp. 510-526, 2014.

[25] C.E. Yin, G. Qu, “LISA: Maximizing RO PUF's Secret Extraction”, IEEE International Symposium on *Hardware-oriented Security & Trust*, pp. 100-105,2010.

[26] Majzoobi Mehrdad, Koushanfar Farinaz, and Devadas Srinivas. “FPGA PUF using programmable delay lines”. In *IEEE International Workshop on Information Forensics and Security* (WIFS), pp. 1–6, 2010.

[27] Majzoobi Mehrdad, Farinaz Koushanfar, and Srinivas Devadas. “FPGAbased true random number generation using circuit metastability with adaptive feedback control”. In *Cryptographic Hardware and Embedded Systems* (CHES), pages 17–32, 2011.

[28] Maiti A. and Schaumont P. “Improved Ring Oscillator PUF: An FPGA friendly Secure Primitive”. *Journal of Cryptology*, pp. 24: 375–397, 2011.

[29] Knut Wold and Chik How Tan. “Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings”. In *International Conference on Reconfigurable Computing and FPGAs*, 2008.

[30] Gassend B., Clarke D., van Dijk M., and Devadas S. Silicon “Physical Random Functions”. In *ACM Conference on Computer and Communications Security* (CCS), pp. 148–160, 2002.

李昌婷 于 2013 年在清华大学电子信息科学与技术专业获得学士学位。现在中国科学院信息工程研究所信息安全专业攻读博士学位。研究领域为嵌入式系统安全。研究兴趣包括: 加密算法的硬件实现、PUF 技术的应用、侧信道攻击等。Email: lichangting@iie.ac.cn

章庆隆 于 2016 年在中国科学院大学信息安全专业获得工学博士学位。研究领域为硬件密码算法优化, 嵌入式安全研究。研究兴趣包括, 国密算法高速实现, 物理不可克隆方法。Email: qlzhang@is.ac.cn

刘宗斌 于 2012 年在中国科学院研究生院信息安全专业获得博士学位。现在中国科学院信息工程研究所第三研究室, 高级工程师。研究领域为嵌入式系统安全。研究兴趣包括: 密钥保护、随机数检测、密码算法安全实现、侧信道攻击等。Email: liuzongbin@iie.ac.cn

荆继武 于 2003 年在中国科学院研究生院获得博士学位。现任中国科学院信息工程研究所副所长。研究领域为嵌入式系统安全。主要从事系统安全, 特别是入侵容忍系统方面的研究工作。Email: jingjiwu@iie.ac.cn