基于 SRAM 物理不可克隆函数的高效真随机种子发生器设计

李 冰[①] 涂云晶[①] 陈 帅[*①] 吉建华[②]
[①](东南大学集成电路学院 南京 210000)
[②](深圳大学信息工程学院 深圳 518060)

摘 要：该文设计了一种基于 SRAM 物理不可克隆函数(PUFs)的高效真随机种子发生器。通过将不提供熵值的稳定节点和提供低熵值的噪声节点筛除，只选用能够提供较高熵值的噪声节点来生成满熵种子，大幅降低需要处理的数据量，提高节点数据的处理效率。通过测试 SRAM PUFs 内部噪声节点的振荡特性，提出筛选出 SRAM PUFs 内部高熵值的噪声节点的最佳策略，最终基于此策略设计出真随机种子发生器。该设计可以产生 128~256 bit 长度的满熵的种子且处理的节点数据量只有当前方法的 0.5%~4%。生成的种子满足 NIST 架构的随机数生成器要求，产生的伪随机数全部通过了随机数检测。与现有设计相比，该文提出的真随机种子发生器是一种高效的、适用范围较广的设计。

关键词：物理不可克隆函数；噪声节点；真随机种子；高效率
中图分类号：TN47 **文献标识码**：A **文章编号**：1009-5896(2017)06-1458-06
DOI: 10.11999/JEIT160835

Efficient Design of Truly Random Seed Generator Based on SRAM Physical Unclonable Functions

LI Bing[①] TU Yunjing[①] CHEN Shuai[①] JI Jianhua[②]
[①](*School of Integrated Circuit*, *Southeast University*, *Nanjing* 210000, *China*)
[②](*College of Information Engineering*, *Shenzhen University*, *Shenzhen* 518060, *China*)

Abstract: An efficient design of truly random seed generator based on SRAM Physical Unclonable Functions (PUFs) is proposed in this paper. Only the noisy cells of high min-entropy are selected to generate full entropy seeds in this design. Therefore, it can reduce the amount of data to be processed significantly and improve the efficiency of seed generation. The oscillating characteristics of the noisy cells inside SRAM are measured, and screening strategies for filtering out the selected noisy cells inside the SRAM are put forward. Finally, based on the strategies, a truly random seed generator is designed, which can generate full entropy seeds. The length of seeds generated by this design is from 128 bit to 256 bit. The number of the selected cells which are used to generate seeds is from 0.5% to 4% of all SRAM cells. Compared to the current design, it is shown that the proposed design in this paper is efficient and widely applicable.

Key words: Physical Unclonable Functions (PUFs); Noisy cells; Truly random seed; Efficient

1 引言

在当今社会，信息安全的问题日益被人们关注。物理不可克隆函数(Physical Unclonable Functions, PUFs)[1]作为其中重要的组成部分，被越来越多的研究者所研究。SRAM PUFs 因其拥有设计简单、经济性好、可靠性较高[2]等特点而广受青睐。无论是身份认证[3,4]、随机数产生[5]还是知识产权保护[6]、信息加密[7,8]，SRAM PUFs 都有很好的应用前景。

在伪随机数产生过程中，真随机种子的熵值决定了由其产生的伪随机数的可预测性强弱。目前，基于 PUFs 的伪随机数发生器大多是针对环形振荡器 PUFs[9,10]进行设计，而基于 SRAM PUFs 的随机数发生器也有一定研究，但是当前的设计方法[11,12]在处理大量节点数据方面效率较低。

当前，大多研究者从 SRAM 中提取熵值的办法都是将 SRAM 所有节点上电之后的值输出给调节算法(conditioning algorithm)，这些节点的值通过一些处理之后由不同的压缩算法，将所有节点的值压缩成固定长度的种子。如果这些节点的熵值足够多，那么产生的种子是满熵的。

以这种方式处理节点的优点在于设计结构和处

收稿日期：2016-08-15；改回日期：2017-01-11；网络出版：2017-03-07
*通信作者：陈帅 chenshuai_ic@seu.edu.cn
基金项目：国家自然科学基金(61571116)
Foundation Item: The National Natural Science Foundation of China (61571116)

理逻辑较为简单，但是缺点也是显而易见的。首先，为了减少需要处理的节点数据量，此方法大多是采用小容量的片上 SRAM 或者 FPGA 内部的 Block RAM 充当熵源。这些熵源的节点数量相对较少，可能会因为最小熵之和相对较低，而只能产生较短的满熵种子。这导致所产生的种子无法应用到安全性要求较高的场景中，限制了其应用范围。

其次，如果单纯地使用更大的片上 SRAM 芯片来充当熵源，那之前的结构就不能满足需要。因为调节算法一次处理的数据量是有限的，数据量的增加将会导致处理次数大大增加，这不仅使得算法的控制逻辑变得复杂，而且处理轮次的增加将会拖慢整个系统的处理速度，导致节点数据处理效率很低。

最后，一旦片上 SRAM 内部节点的数据泄漏，想要更换熵源是一件极其困难的事情，如果继续使用不安全的熵源，那么产生的随机数的安全性就无法得到保证，因而存在安全隐患。

为了解决当前方法中的问题，本文提出如下改进方法：首先，在保证汉明距离符合要求的前提下，使用较大容量的片外 SRAM 充当熵源。这样做的好处在于，在发现节点数据不安全的情况下，可以更换新的 SRAM 芯片来保证安全性。同时，根据应用场景的需要可以在一定范围内选择片外的 SRAM 芯片的容量，提高了整个设计的应用范围。

其次，为了提高节点数据处理效率，需要对原来的处理方法做出改变。本文认为方法有两种，一种是改进当前调节算法的处理能力或者是使用处理能力更高的新算法。另外一种改进方法就是通过制定筛选策略将噪声节点从所有节点中筛选出来，然后对这些节点直接处理得到种子。

本文旨在设计一种基于 SRAM PUFs 的高效的真随机种子发生器。其产生的满熵种子可以传递给加密模块充当随机密钥或者传递给随机数生成平台充当随机种子。本文通过先筛除 SRAM PUFs 中的强偏性节点，再分析弱偏性节点振荡规律，然后设计出能够将大量低振荡次数的弱偏性节点筛除的节点筛选策略。一般情况下，最终用来产生种子的节点数量只占总节点数量的 0.5%~4%，大大提高了处理效率。

本文其余部分安排如下：第 2 节提出了 SRAM PUFs 芯片内部的两类节点及节点最小熵计算公式；第 3 节提出了本文的设计方法，通过增加节点的筛选模块，将不需要的节点过滤，只处理需要的节点；第 4 节制定了筛选节点的策略的相关参数；第 5 节将本文设计在 FPGA 上实现，并且借助 NIST 检测套件测试了所产生的种子的质量是否合格；第 6 节总结全文。

2　SRAM 节点

2.1　两类节点

SRAM PUFs 的节点特性已经被深入研究，其中 Holcomb 等人[3]在 2009 年就提出 SRAM PUFs 作为熵源的可行性，其提出了 SRAM 节点在掉电之后再上电的值是不一样的，这些节点可以分成两类节点，一类是上电之后的值固定偏向于 0 或者 1，另外一类是上电之后的值具有不确定性，其值会受到外界环境影响，有时偏向于 0，有时会偏向于 1。Cortez 等人[13]也对 SRAM 内部节点在上电之后的行为建立模型进行分析，发现生产制造之中的微小差异、外界环境的温度、供电电压、NMOS 管的 V_{th} 和晶体管的对称结构等都对某些节点上电之后的值有影响。

因此，本文对于这两类节点进行如下区分：那些具有强偏性的节点，本文中称之为稳定节点，上电之后，这些节点的值都是固定为 0 或者是 1，即它们的值是可预测的；那些弱偏性的节点，本文中称之为噪声节点，每次上电之后，这些节点的值因受到各种内在差异和外界条件的影响而表现为有时值为 0，有时值为 1，即它们的值是不可预测的。

SRAM 作为熵源原理在于其噪声节点上电值的不可预测性，这些噪声节点所提供的最小熵之和关系到最后生成的种子的熵值，通过最小熵计算公式，可以计算出这些节点的最小熵之和是否满足要求。

2.2　最小熵计算方法

本文将采用美国国家标准与技术研究院(National Institute of Standards and Technology, NIST)的标准文档[14]中的标准来计算节点值的最小熵。

对于一个节点而言，其值只有 0 和 1 这两种可能，对这个节点进行多次测量之后分别可以得到值为 0 的概率 P_0 和值为 1 的概率 P_1，这两个概率相加的结果必然是 1。用 $P_{\max}$ 来表示这两个概率值中的较大值，即为

$$P_{\max} = \max\{P_0,\ P_1\} \tag{1}$$

那么对于一个节点的最小熵，可以通过式(2)来求得：

$$H_{\min} = -\log_2(P_{\max}) \tag{2}$$

对于 SRAM 内部众多的节点，假定节点之间是相互独立、互不干扰的，那么对于 n 个独立的节点来说，他们的最小熵之和应该是

$$(H_{\min})_{\text{t}} = \sum_{i=1}^{n} -\log_2(P_{i\max}) \tag{3}$$

如果 n 个节点的最小熵之和小于所产生的种子

长度 N，那么这 n 个节点所产生的种子就不是满熵的。对于随机数发生平台来说，得到的种子熵值越高，其产生的伪随机数的抗预测性越好。

3 种子发生器结构设计

根据前文所述，之前的研究者大多采用的是图 1 中的方法，这种方法是将所有节点上电后的数据用来提取熵值，产生真随机种子，但是为了提高处理效率，本文对此架构作一定改进。

图 2 为本文所设计的真随机种子发生器(Truly Random Seed Generator, TRSG)，其在熵源和调节算法之间增加了一个筛选模块。此模块的功能是对 SRAM 上电之后所有节点数据的振荡特性进行分

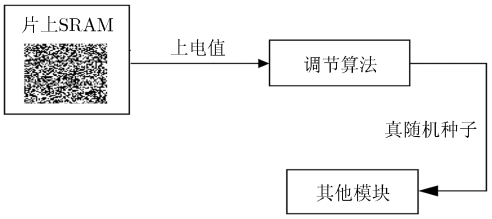

图 1 传统的熵提取架构

选模块，筛选模块内部的结构如图 3 所示。筛选模块中有两个 SRAM 控制器，一个是用于读取片外 SRAM 数据的控制器，一个是用于在片内存储相关数据的控制器。结构中的浮点计算单元是用来计算筛选出来的节点值的最小熵之和。

筛选模块在每一块新的 SRAM 芯片插入读取基座时都会进行一次筛选操作，这个操作的目的是为了检测插入的芯片能否提供足够的熵值用于生成满熵的种子。若熵值较低则需要更换新的芯片，若熵值足够则将筛选出来的噪声节点的地址存储在片内的 SRAM 中。当其他模块请求种子时，在掉电时间结束之后种子发生器就会给片外 SRAM 熵源上电并且根据已经保存的噪声节点的地址信息，读取对应地址的节点数据，然后将数据传输给调节算法模块生成满熵种子，最后种子通过接口传输给其他模块。

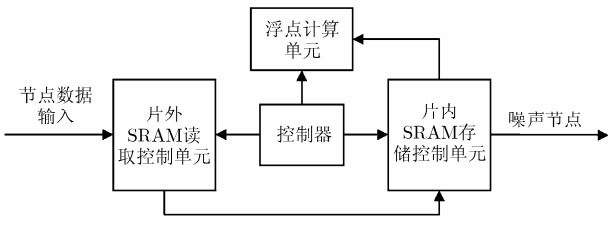

图 3 筛选模块结构

析，先从所有节点中筛选出噪声节点，然后根据已经制定好的筛选策略对噪声节点再次筛选，选出符合要求的噪声节点。同时根据式(3)实时计算出这些节点值的最小熵之和并且判断最小熵之和是否满足要求，若最小熵之和满足要求则将筛选出来的噪声节点地址进行保存。当外界其他模块请求种子产生器产生种子时，筛选模块会按照已经保存的地址去读取对应节点上电之后的值，并将这些噪声节点的值传输给调节算法，最终由调节算法根据外界请求的种子长度生成相应的 128～256 bit 长度的种子，生成的种子能够保证满熵。

本设计的创新之处就是在结构中增加了一个筛

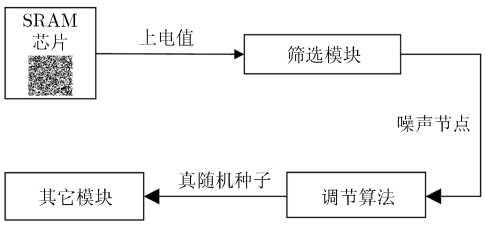

图 2 改进后的真随机种子发生器架构图

4 筛选策略的制定

本文设计中的筛选模块能够有效工作的前提就是筛选策略的制定。制定的筛选策略将会作为参数提前设定在系统中，这些参数也可以通过外部输入指令加以更改。

4.1 筛选读取次数设置

筛选读取次数的大小的设定是根据的 Leest 等人[5]的研究成果，其在实验中测试发现，当读取次数在 100 次及以上时，SRAM 所有节点的平均最小熵趋于稳定值，也就是说，所有的噪声节点的值基本上都能够发生一次或者更多次的振荡。因此，本设计将筛选读取次数设置为 100 次。

4.2 掉电时间设置

在众多 SRAM PUFs 的文章中，研究者大多都没有介绍其测试 SRAM 节点的平均最小熵的过程中所设定的 SRAM 掉电时间。文献[15]中提到先前写入的数据对于 SRAM 上电值的影响很小，因此针对 SRAM 掉电时间的研究只需要确定一个最合适的掉电时间即可。

根据文献[16]的发现，正常温度下 SRAM 数据滞留时间大约在 ms 级，但是当温度趋近于 0℃时，有一部分芯片的数据滞留时间超过了 1 s，当温度继续降低到−15℃时，部分芯片数据滞留时间会超过 10 s 甚至更多。

结合以上两点可知，如果掉电时间比数据滞留时间短，则噪声节点不会发生振荡，此时如果用噪声节点的值生成的种子是不安全的；如果掉电时间过长，那么对于实际应用而言是不现实的，因此掉电时间的设置对于种子发生器来说尤为关键。针对于这个问题，本文研究了不同掉电时间对于噪声节点的数目变化的影响。如图 4 所示，本文对 HY62256ALP 系列 5 块芯片做了测试，这 5 块芯片的测试工作持续了 5 周时间，测试温度为正常室温，反复读取次数为 100 次，每个组别中有 5 个柱状图形代表着该掉电时间下的 5 块芯片对应的测试数据，每一块芯片的每一组的数据都是测量 10 次的平均值。其中横坐标时间组别从 1 到 7 分别代表设置的掉电时间为 1 s, 3 s, 5 s, 10 s, 15 s, 1 min, 5 min，纵坐标为噪声节点占全部节点数量的百分比。需要注意的是，只要某个节点的值在 100 次读取测试中变化了一次，本文就认为该点是噪声节点。

同样，图 5 是本文对 WS62256LLP 系列的 5 块芯片测试结果。从图中可以看出此系列芯片的数据滞留时间都很低，相比于 HY62256 系列芯片而言，不同掉电时间之间噪声节点数目差别不大。需要指出的是，对于 WS62256 系列的第 5 块芯片，其掉电时间无论为多少，噪声节点数量都极少，可见在同一系列的同一批次的产品中，因为生产制造上的差异，导致表现出来的结果也会有差别。

因此，为了保证每一块芯片都能够用于种子产生，对芯片进行一次筛选读取是极有必要的。对于部分 SRAM 芯片来说，为了保证其能够充当熵源，那么掉电时间一定要设置成合适的值，根据实验结果，本设计将掉电时间设置为 5 s。

4.3 最低振荡次数设置

在筛除稳定节点之后，需要处理的节点数量大幅降低，一般只为全部节点的 2%~8%。由此可见，筛选后的噪声节点数量还是较大，本文想进一步提高处理数据效率就需要将那些提供熵值很低的节点筛选掉，选取那些能提供很高熵值的噪声节点，在损失尽可能少的熵值的基础上，进一步减少需要处理的数据量。

前文对于噪声节点的判定是认为在 100 次测试中，节点值改变了至少一次的节点是噪声节点。但是以下节点需要特别考虑：

(1)由于某些外界环境的强刺激发生了一次或者几次偶然振荡的强偏性的节点。

(2)P_0 和 P_1 相差很大的振荡次数较低的弱偏性的节点。

(3)P_0 和 P_1 很接近的振荡次数较低的弱偏性的节点。

根据式(2)，前两类节点所提供的最小熵较低，但是这两类节点的数量占整个噪声节点的数量的百分比相对较大。所以本文为了进一步提高效率，决定将低振荡次数的噪声节点筛除。但是，低振荡次数的节点除了前两类节点之外，还有第 3 类节点。第 3 类节点虽然可以提供较大的熵值，但是这一类噪声节点的数目极少，如果筛除这些节点，对于种子产生来说影响很小，然而却可以大大减小筛选难度，综合考虑之下，本文决定筛除这类节点。因此，本文需要通过实验来判断，发生多少次以上振荡的节点是符合要求的节点。

如图 6 所示，本文将 HY62256ALP 系列的一块芯片得到的噪声节点按照振荡次数进行区分，横坐标 $\ln(t)$为掉电时间的自然对数，纵坐标 Hpb 为平均每比特的最小熵，图中从下到上的 5 簇折线分别是振荡 1 次及以上、10 次及以上、20 次及以上、30 次及以上、40 次及以上噪声节点的每比特最小熵，每一簇折线由 10 根折线组成，分别代表着 10 次重复测试的结果。从图上可以看出，振荡 10 次及以上的噪声节点的每比特最小熵数值最稳定，无论掉电时间是 1 s 还是 5 min，测得的噪声节点的每比特最小熵的值都稳定在 0.4 附近。从图 6 中还能发现振荡次数每提升十次，节点所能提供的每比特最小熵大约提高 0.1。以上现象并不是偶然的，在本文测试的 10 块芯片当中，有 9 块芯片都有同样的规律。

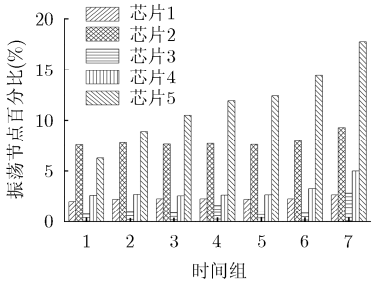

图 4 在不同掉电时间下 HY62256 ALP 芯片的噪声节点的百分比

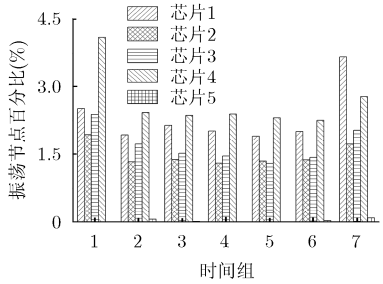

图 5 在不同掉电时间下 WS62256 LLP 芯片噪声节点的百分比

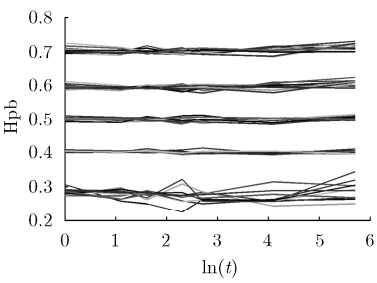

图 6 一块芯片中不同振荡次数的节点的每比特最小熵的对比

图 7 中，横坐标是表示振荡 10 次以上的噪声节点占全部噪声节点的比例，纵坐标为对应比例出现的数据次数，此图是正常室温下，反复读取次数为 100 次，掉电时间为 5 min 所得到 50 组数据统计结果，从图中可看出振荡 10 次以上的噪声节点占全部噪声节点的比例相对较小，其占全部噪声节点的比例大于 50%的次数只有 14 次。这意味在筛除振荡 10 次以下的噪声节点之后，需要处理的数据量将大大减少，同时剩余噪声节点所提供的熵值也完全足够。

图 6 中，虽然振荡次数较高的节点的每比特最小熵相对较高，但是这些节点的数目相对较少，所能提供的熵值可能不足，这会导致产生的种子不满熵，因此综合考虑，本文将最低振荡次数设置为 10 次。

5 TRSG 的硬件实现

5.1 FPGA 资源消耗

本文根据制定的筛选策略，完成了真随机种子发生器(TRSG)的设计，整个平台是在 Altera 公司的 Cyclone IV 系列芯片上搭建完成，工作的时钟频率为 50 MHz，使用的逻辑单元(Logic Elements, LE)为 11153 个，使用的存储资源为 366592 bit。所设计的发生器能在初始化之后向其他模块传输 128～256 bit 长度的满熵种子。一般情况下，此改进方法每次需要处理的数据量为原先的 0.5%～4%，所需处理的数据量大幅降低。本设计产生的种子可以充当加密模块的随机密钥，也可以充当伪随机数发生平台的种子。如表 1 所示，伪随机数发生平台根据本文设计的真随机种子发生器产生的种子而产生的伪随机数经过 NIST 的检测套件检测之后，所有检测项目的 P-value 值也都大于 0.01，这说明本设计达到了预期。

5.2 安全性讨论

在实验的过程中，本文发现并非所有的 SRAM 芯片都适合用于产生种子，有些芯片内部的噪声节点极少，不足以产生满熵的种子。图 8 是测试 HM62256ALP 系列芯片内部噪声节点个数，从图中

表 1 测试结果

测试项目	P值	百分比(%)
频率检验	0.474986	97.0
块内频数检验	0.075719	100.0
向前累加和检验	0.191687	98.0
向后累加和检验	0.779188	98.0
游程检验	0.350485	100.0
块内最长游程检验	0.026948	99.0
二元矩阵秩检验	0.383827	98.0
离散傅里叶变换检验	0.213309	95.0
非重叠模块匹配检验	0.477180	98.9
重叠模块匹配检验	0.181557	98.0
Maurer通用统计检验	0.759756	100.0
近似熵检验	0.779188	100.0
随机游动检验	0.481197	99.2
随机游动状态频数检验	0.410814	99.0
序列检验	0.440435	98.0
线性复杂度检验	0.494392	99.0

可以看出此系列芯片噪声节点个数普遍偏低，大多都不适合作为熵源。本文认为出现这样的情况的原因可能是这些 SRAM 芯片内部结构的不对称，也有可能是因为生产该系列芯片所用的工艺有所不同。在每一块新的 SRAM 芯片被使用前，都需要经过一些操作以判断其是否有足够的熵值。

本设计方法从理论上来说，是比现有的方法更加安全。首先，本方法能够产生长度更长的满熵种子，这就保证了由种子产生的伪随机数的抗预测性更好。其次，现有的方法一般使用的都是节点数目较少的片上 SRAM 或者 FPGA 内部的 Block RAM。其目的是为了减少需要处理的数据量提高种子产生效率，但是这样导致节点数量过少而容易被攻击者攻破，而且一旦节点数据被泄漏，那么想要更换芯片就比较困难。如果继续使用不安全的节点信息，那么产生的种子也是不安全的。因此，理论上本设计相对于当前的设计方法而言，是较为安全的。

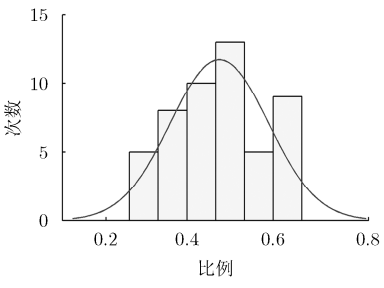

图 7 振荡 10 次以上节点占噪声节点比例分布直方图

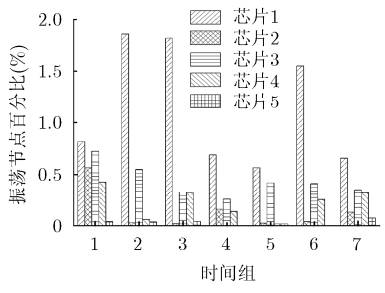

图 8 在不同掉电时间下 HM62256ALP 芯片的百分比

6　结束语

本文提出了一种基于 SRAM PUFs 的真随机种子发生器设计，此设计可以解决现有方法处理大量节点数据效率低下的问题。为了有效地减少需要处理的数据量，本文先是筛选掉不能提供熵值的稳定节点，然后针对不同振荡次数的噪声节点进行分类，找到了能够筛选掉较低振荡次数的噪声节点的策略以进一步减少需要处理的节点数据量。实验结果表明，最终本设计只需使用大约总节点数的 0.5%～4% 的节点，就能够生成长度从 128～256 bit 的满熵的种子。本设计所产生的真随机种子传递给伪随机数发声平台，由伪随机数发生平台产生大量的随机数。根据测试结果可以判断，本设计所产生的真随机种子的随机性较高，因此，本文提出的设计适合应用在对安全性要求较高的场景当中，而不适合应用在轻量级的场景中。

参 考 文 献

[1] PAPPU R, RECHT B, TAYLOR J, *et al*. Physical one-way functions[J]. *Science*, 2002, 297(5589): 2026–2030. doi: 10.1126/science.1074376.

[2] BARBARESCHI M, BATTISTA E, MAZZEO A, *et al*. Testing 90 nm microcontroller SRAM PUF quality[C]. IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era, Naples, Italy, 2015: 1–6.

[3] HOLCOMB D E, BURLESON W P, and FU K. Power-Up SRAM state as an identifying fingerprint and source of true random numbers[J]. *IEEE Transactions on Computers*, 2008, 58(9): 1198–1210. doi: 10.1109/TC.2008.212.

[4] XIAO K, RAHMAN M T, FORTE D, *et al*. Bit selection algorithm suitable for high-volume production of SRAM-PUF[C]. IEEE International Symposium on Hardware-Oriented Security and Trust, Arlington, Virginia, USA, 2014: 101–106.

[5] LEEST V V D, SLUIS E V D, SCHRIJEN G J, *et al*. Efficient implementation of true random number generator based on SRAM PUFs[J]. *Lecture Notes in Computer Science*, 2012, 6805: 300–318. doi: 10.1007/978-3-642-28368-0_20.

[6] ZHANG J, LIN Y, LYU Y, *et al*. A PUF-FSM binding scheme for FPGA IP protection and pay-per-device licensing[J]. *IEEE Transactions on Information Forensics & Security*, 2015, 10(6): 1137–1150. doi: 10.1109/TIFS.2015.2400413.

[7] DELVAUX J, GU D, SCHELLEKENS D, *et al*. Helper data algorithms for PUF-based key generation: Overview and analysis[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2015, 34(6): 889–902. doi: 10.1109/TCAD.2014.2370531.

[8] KIM H and HONG S. AES Sbox GF(2^4) inversion functions based PUFs[C]. IEEE International SoC Design Conference (ISOCC), Jeju, South Korea, 2014: 15–16.

[9] VARCHOLA M, DRUTAROVSKY, M, and FISCHER V. New universal element with integrated PUF and TRNG capability[C]. International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico. 2013: 1–6.

[10] HUSSAIN S U, MAJZOOBI M, and KOUSHANFAR F. A built-in-self-test scheme for online evaluation of physical unclonable functions and true random number generators[J]. *IEEE Transactions on Multi-Scale Computing Systems*, 2016, 2(1): 2–16. doi: 10.1109/TMSCS.2016.2519902.

[11] LI D, LU Z, ZOU X, *et al*. PUFKEY: A high-security and high-throughput hardware true random number generator for sensor networks[J]. *Sensors*, 2015, 15(10): 26251–26266. doi: 10.3390/s151026251.

[12] HERREWEGE V A, VINCENT V D L, SCHALLER A, *et al*. Secure PRNG seeding on commercial off-the-shelf microcontrollers[C]. International Workshop on Trustworthy Embedded Devices, Berlin, Germany, 2013: 55–64.

[13] CORTEZ M, DARGAR A, HAMDIOUI S, *et al*. Modeling SRAM start-up behavior for physical unclonable functions[C]. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Austin, TX, USA, 2012: 815–828.

[14] BARKER E and KELSEY J. Recommendation for random number generation using deterministic random bit generators[R]. NIST Special Publication, 2012: 800–890.

[15] GUAJARDO J, KUMAR S S, SCHRIJEN G J, *et al*. FPGA intrinsic PUFs and their use for IP protection[C] Cryptographic Hardware and Embedded Systems, CHES 2007, International Workshop, Vienna, Austria, 2007: 63–80.

[16] SKOROBOGATOV S. Low temperature data remanence in static RAM[J]. *University of Cambridge Computer Laboratory Technical Report*, 2002, 536: 1–11.

李　冰：男，1968 年生，教授，博士生导师，研究方向为数模混合集成电路设计、信息安全和云计算等.

涂云晶：男，1992 年生，硕士，研究方向为数字集成电路设计和信息安全.

陈　帅：男，1989 年生，博士，研究方向为数字集成电路设计、信息安全和数据压缩.

吉建华：男，1970 年生，教授，硕士生导师，研究方向为光通信和信息安全.