From Statistics to Circuits: Foundations for Future Physical Unclonable Functions

Inyoung Kim, Abhranil Maiti, Leyla Nazhandali, Patrick Schaumont, Vignesh Vivekraja, and Huaiye Zhang

1 Introduction

Identity is an essential ingredient in secure protocols. Indeed, if we can no longer distinguish Alice from Bob, there is no point in doing a key exchange or in verifying their signatures. A human Alice and a human Bob identify one another based on looks, voice, or gestures. In today's networked world, Alice and Bob are computer programs. Their identity relies on the computer hardware they execute on and this requires the use of a trusted element in hardware, such as a Trusted Platform Module (TPM) [1]. In the future, Alice and Bob will include tiny embedded computers that can sit anywhere – in a wireless key, in a cell phone, in a radio-frequency identifier (RFID) [2]. Such tiny electronics cannot afford a full-fledged TPM for identification and authentication. There is thus a great need to develop cost-efficient, reliable, stable, and trustworthy circuit identifiers that can fit in a single chip combined with the rest of the system.

Traditional approaches to hardware identity, such as non-volatile memories, increase system cost, may be tamperable, and at times are not trustworthy. We are therefore investigating electronic fingerprints that are based on the existing, small and random manufacturing variations of electronic chips.

Physical unclonable functions (PUFs) are a known solution to create an on-chip fingerprint. However, the issues of scalability, cost, reliability, and the threats of reverse engineering have not been fully investigated. We advocate a cross-disciplinary approach to combine recent advances in the field of statistics with those in circuits design. Our main concerns in PUF design are to come up with designs that (a) are cheap to build and integrate, (b) show a high reliability toward environmental changes and aging, and (c) are able to prevent some common attacks. In this chapter, we will report several steps toward achieving these objectives.

I. Kim (✉)
Statistics Department, Virginia Tech, Blacksburg, VA 24061, USA
e-mail: inyoungk@vt.edu

A.-R. Sadeghi, D. Naccache (eds.), *Towards Hardware-Intrinsic Security,* Information Security and Cryptography, DOI 10.1007/978-3-642-14452-3_3,

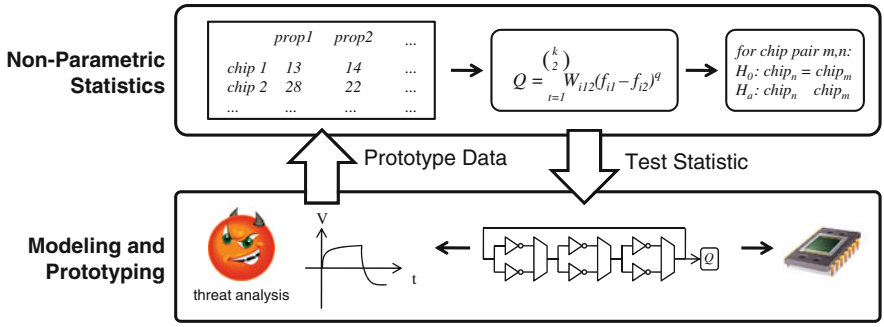

Fig. 1 Combining statistics, architecture, and circuit in the PUF design process

Figure 1 illustrates the two layers in our cross-disciplinary approach: statistics and modeling and prototyping. Novel data-processing ideas are created in the statistics layer and passed down to the architecture layer. The architecture layer, in turn, implements prototypes and specific optimizations and returns prototype test data to the statistics layer.

- *Statistics*: Based on data obtained from prototype architectures, we are developing a novel *test statistic* that can be used as a unique on-chip fingerprint. A test statistic (TS) is an expression that transforms the measurement data into a single number. Besides a TS, we are also working on adequate hypothesis testing techniques to distinguish chips. Based on non-parametric statistics, the measurement data can be directly used, and assumptions on the underlying statistical distributions are avoided [3].
- *Modeling and Prototyping*: We are also working on prototype architectures in CMOS technology. The target architectures include field-programable gate arrays and standard cells. The designs are driven by the TS requirements, but they extend it with architecture-level optimizations and circuit-level optimizations. The modeling layer supports threat analysis by investigation of the response of the resulting PUF designs to specific active and passive attacks.

This chapter is structured as follows. In the next section, we describe a generic model for a PUF architecture. We identify the major phases of PUF operation (sample measurement, identity mapping, and quantization), and we demonstrate how each phase can be handled using different techniques. Next, we discuss related work, with specific attention to the research that has been done to improve reliability, cost, and threat sensitivity. We then cover our research efforts in PUF design. This includes a discussion on circuit-level optimization techniques based on subthreshold voltage operation (Sect. 3), a discussion on architecture-level optimization techniques to efficiently implement redundancy (Sect. 4), and a discussion on statistical techniques for identity mapping an testing (Sect. 5). We provide outlook and conclusions in Sect. 6.

2 Components and Quality Factors of a PUF Design

This section describes a generic template for CMOS-based PUF designs. We provide numerical expressions for the various PUF quality factors. We also discuss sources of wanted and unwanted variability in CMOS technologies. For a review of existing PUF technologies, we refer to the chapter by Maes and Verbauwhede.

2.1 Components of a PUF

Figure 2a illustrates the three components of a PUF. They include sample measurement, identity mapping, and quantization. Figure 2b shows the example of a Ring Oscillator PUF (RO PUF), which takes a 2-bit challenge C and which produces a single-bit response R.

- A *Sample Measurement* converts a digital challenge C into a vector of physical measurements that reflects the identity of the device. In the example of the RO PUF in Fig. 2b, the challenge selects two out of four oscillators. To complete the measurement, the frequency of both selected ROs is measured. The frequencies are determined by the round-trip delay of each RO, which in turn depends on the process manufacturing variations of the digital components for the RO. Hence the pattern of frequencies is unique for each chip.
- In *Identity Mapping*, a vector of measurements is converted into a single, real valued, decision variable. In the example of the RO PUF, the test statistic

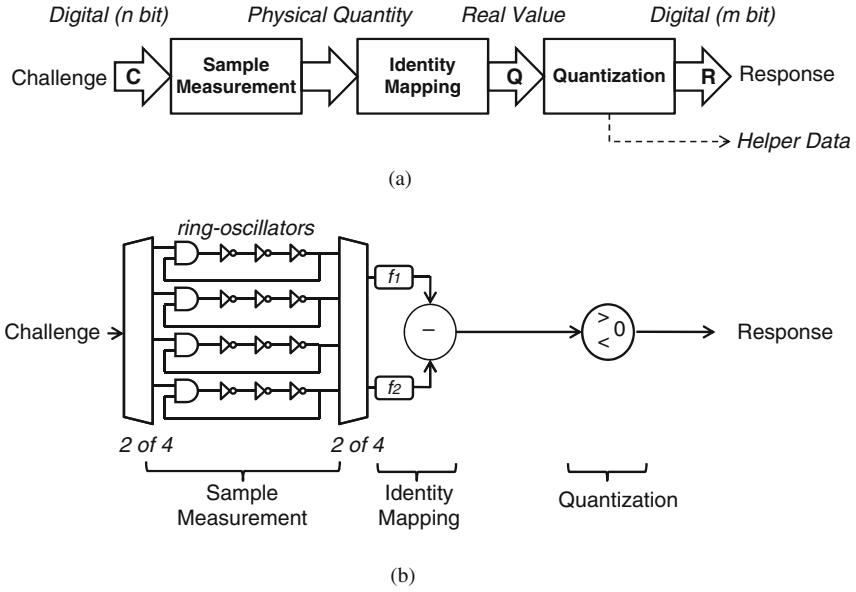

Fig. 2 (**a**) Generic PUF structure (**b**) A sample implementation: the ring oscillator PUF

(TS) is the frequency difference of the two selected RO. Because physical measurements can be noisy, the TS will be influenced by noise as well.
- The *Quantization* step maps the real-valued TS into a digital response *R*. In the example of Fig. 2b, the quantization function is a simple check on the sign of the TS value: negative yields a 0 response and, positive or zero yields a 1 response. Several researchers have generalized the quantization step as a fuzzy-extraction process that generates, besides the response *R*, additional helper-data bits [4, 5]. The role of these helper-data bits is to correct for the effects of noise in the TS. Indeed, using channel-coding techniques, the helper-data bits can be used to correct for bit errors in future noisy responses [6, 7]. This guarantees that a given challenge will always map to the same response.

Figure 2a can be used as a template for different types of PUF technologies. In recent years, proposals have been made that exploit the startup state of SRAM and flip-flops [8–10], on-chip logic delay [11–13], and the resistance of the on-chip power grid [14, 15]. Hence, there is general agreement that modern silicon technology contains ample amounts of process variation to implement Identity Mapping.

The typical use of a PUF is as follows. The C/R pairs available from a PUF are used for authentication of the PUF. A verifier that wants to use a PUF obtains a table with known C/R pairs. This table is provided by a trusted source that can *enroll* PUF devices before they are deployed. To authenticate the PUF, the verifier then selects a challenge from the table, sends it to the PUF, and compares the response with the enrolled response in the table. In order to prevent playback attacks, each C/R pair in the table may be used only once. Hence, the trustworthy lifetime of a PUF is determined by the number of C/R pairs in the table. In addition, in order to prevent aliasing between PUFs, each C/R pair must be unique over the PUF population. Therefore, the total number of unique C/R pairs determines the number of PUF circuits that can be fielded.

2.2 PUF Quality Factors

In their chapter, Handschuh and Tuyls already defined reliability and security as two PUF quality factors. We can add a third metric, namely design cost. A good PUF design will seek a trade-off between these three factors.

- *Cost*: A low-cost on-chip fingerprint means that the smallest possible circuit area is used to identify a given population of chips or that a given PUF design can distinguish the largest possible population of chips.
- *Reliability*: A stable on-chip fingerprint requires that a given PUF characteristic is insensitive to environmental variations (temperature, voltage, noise) and to temporal variations (aging).
- *Security*: A high-quality PUF should provide a high entropy (a large amount of secret bits), should be resistant against tampering, and should provide unclonabil-

ity. Majzoobi describes unclonability as being resistant to reverse engineering, as well as being resistant to emulation [16].

We need to be able to quantify these factors. This is not an easy task. The most straightforward factor is design cost, which is directly proportional to the silicon area of a PUF design. A smaller design is cheaper to manufacture.

Maes and Verbauwhede described two metrics in their chapter that can be used to estimate the PUF reliability and entropy. They introduce *intra-distance* as the Hamming distance between two evaluations in a single PUF under the same challenge. They also define *inter-distance* as the Hamming distance between the responses of two different PUFs under the same challenge.

We can use these metrics as an estimate of *reliability* and *security*. Assuming we have a set of PUF, then the reliability can be defined as the average number of bits from a response that will stay unchanged under the same challenge. Increased reliability simplifies the quantization step in Fig. 2. Numerically, reliability is estimated as follows.

$$S|_{C_1} = 100\% - \max_{i,j} \frac{\{\mathrm{HD}(R_i, R_j)\}}{m} \times 100\% \tag{1}$$

with HD equal to the Hamming distance between any two responses R_i and R_j from the same PUF to the same challenge C_1, and m the number of response bits. The optimal reliability is 100%. Measuring reliability implies that one is able to control the environmental factors that can affect a PUF's responses.

A similar metric is created from the inter-distance metric, which is called *uniqueness* by Maes and Verbauwhede in their chapter. Uniqueness is an estimate for the entropy available from a PUF. Over a similar population of PUF, uniqueness can be calculated as follows:

$$U|_{C_1} = \frac{2}{k(k-1)} \sum_{i=1}^{i=k-1} \sum_{j=i+1}^{j=k} \frac{\mathrm{HD}(R_i, R_j)}{m} \times 100\% \tag{2}$$

with HD equal to the Hamming distance between any two responses R_i and R_j from *different* PUFs to the same challenge C_1, k the number of PUFs in the population under test, and m the number of response bits. The optimal uniqueness is 50%.

We refer to the earlier chapters by Maes and Verbauwhede and by Handschuh and Tuyls, for further discussion on PUF quality factors.

2.3 Sources of CMOS Variability and Compensation of Unwanted Variability

In order to design new CMOS PUF architectures or to improve upon existing ones, we need to understand the sources of variability in digital circuits. We distinguish several different sources in Fig. 3.

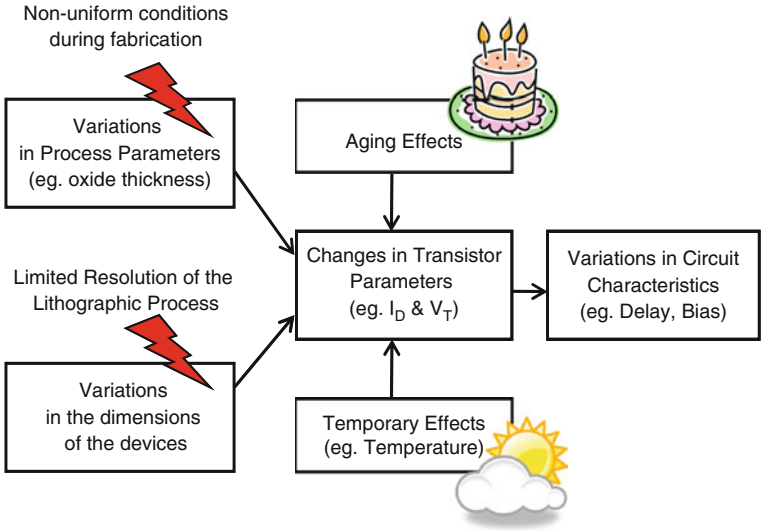

Fig. 3 Causes of variability in circuit characteristics

- *Process Manufacturing Variations*: The random and permanent deviation from the designed, nominal value of a circuit structure, caused by random effects during manufacturing [25]. Process manufacturing variations (PMV) can be separated into two categories. The first category covers variations in process parameters, such as impurity concentration densities, oxide thicknesses, diffusion depths. These result from non-uniform conditions during the deposition and/or the diffusion of the dopants. The second category covers variations in the dimensions of the devices. These result from limited resolution of the photo-lithographic process which in turn causes width and length variations in transistors.
- *Environmental Variations*: These are temporary variations caused by changes in the environmental parameters, including temperature, operating voltage, and external noise coupling. Because the PUF environment cannot always be controlled, the effect of these variations should be minimized. A increase in temperature, and a decrease in power supply, will slow down a circuit. Moreover, the performance loss is not linear, and it may affect different parts of a circuit differently. This will affect the reliability of the PUF.
- *Aging*: Ultra-slow, but eventually permanent, variations that generally deteriorate circuit performance. Aging results in slower operation of circuits, irregular timing characteristics, increase in power consumption, and sometimes even in functional failures [26–28]. Circuit aging is accelerated by use of increased voltages. Aging is largely dependent on how often the circuit is used. Therefore, blocks of a design that are used more often suffer a larger deviation of characteristics with time.

As shown in Fig. 3, each of these variabilities eventually has a similar impact on the observable circuit characteristics. In order to create a stable PUF behavior, we

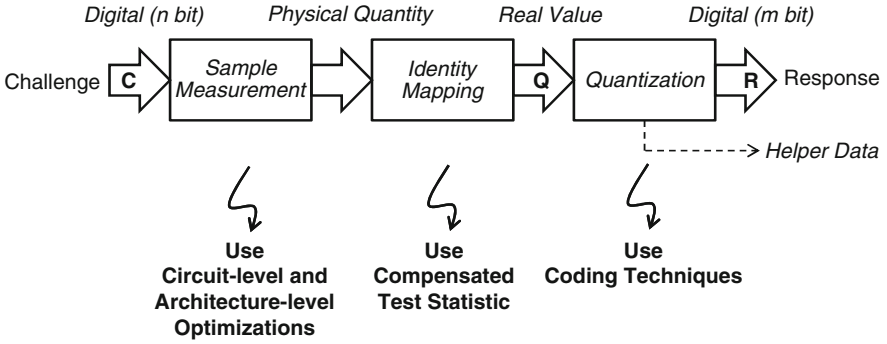

Fig. 4 Compensation of unwanted variability in a PUF

need to detect the process manufacturing variations while tolerating or compensating as much as possible for environmental variations and aging. Figure 4 demonstrates three different strategies to remove the effect of these unwanted variations. Each of these three techniques is implemented on different parts of a PUF design. We briefly describe the general concepts in this section, and we will address their detailed implementation in later sections.

The first approach is to use circuit-level and architecture-level optimizations to provide sample measurement with improved reliability. This will not entirely remove the effect of environmental variations, but it will reduce their effect on the response bits R. The second approach is to improve the identity mapping step, by selecting a test statistic that takes environmental parameters into account. Such a test statistic will estimate the process manufacturing parameters, while ignoring others. The third approach is to use coding techniques that generate helper data in addition to the response. The helper data can be used to reconstruct the correct response at a later time [6, 7]. Each of these three approaches thus work at a different level of abstraction, and therefore they can be combined.

3 Circuit-Level Optimization of PUF

In a CMOS-based PUF, a design built as a CMOS circuit is used to produce random responses, which constitute the basis of the PUF. It is, therefore, very important that the circuit is optimized to improve the quality of the PUF. Conventional circuit design strives to achieve quality factors such as low power and high speed for circuits. Another goal of conventional circuit design is to build the circuit such that a large population of chips have a similar set of characteristics so that they can be conveniently used in their targeted application. In PUF circuit design, although power consumption and overall speed of the PUF have some importance, they are second-hand citizens to quality factors discussed in Sect. 2.2. Therefore, circuit-level decisions have to be made in a completely different fashion.

These decisions include circuit operating voltage, body bias voltage, technology node (transistor length), gate size (transistor width), gate family (high-speed low-

V_T versus low-speed high-V_T gates), and layout decisions (placement of the gates). These decisions vary in terms of their effectiveness in optimizing the PUF quality. In this chapter, we study the effect of the first two in this list, namely, circuit operating (supply) voltage and body bias voltage.

3.1 Methodology

The graphs and numbers presented in this chapter were collected using SPICE simulations. The simulations were performed on a 90 nm technology node, using transistors models and process variation models from UMC [29]. The test circuit is an RO PUF (see Sect. 2.1) with 32 ring oscillators of 11 NAND stages each. The characteristics of 20 different PUF ICs were obtained by applying Monte Carlo simulation on the SPICE model, while enabling both the intra-die and inter-die process variation flags. We ensured that our results are as close as possible to the actual implementation by using the simulation libraries from a commercial foundry and by running the simulation at the highest possible accuracy setting.

3.2 Background: Operating Voltage and Body Bias

It is well known that the power consumption and the frequency of a CMOS circuit are critically controlled by the supply voltage and to some extent by the body bias. The purpose of this section is to explain the impact of these two parameters, not on power and performance, but on the sensitivity of the circuit to process variation, which is key to quality of a PUF. In this regard, we use a key metric: *Coefficient Variation (CV)*. CV is defined as the ratio of standard deviation of a population to its average. For example, the CV of a group of 'n' ring oscillators is the ratio of the standard deviation of its characteristic frequency 'f' to the average characteristic frequency of all ring oscillators, as shown in formula-(3). In statistics, CV is used to compare two different populations with two different averages, in order to see which one has a population whose members are more spread apart. Consequently, in our example, if we build our PUF with two different circuit configurations and the first configuration has a higher value of CV, it indicates that the frequency of the ring oscillators in different chips for this configuration is more spread apart. In other words, the design is more susceptible and less tolerant toward process variation. The motivation behind studying CV is that we believe higher variability can result in higher uniqueness. We investigate this claim in the rest of this section.

$$\mathrm{CV} = \frac{\sigma(f_1, f_2, \ldots f_n)\,(n)}{\Sigma(f_1, f_2, \ldots f_n)}. \tag{3}$$

3.2.1 Operating Voltage

Traditionally, the reduction of supply voltage, also known as voltage scaling, has been successfully employed to reduce the power consumption of a circuit. How-

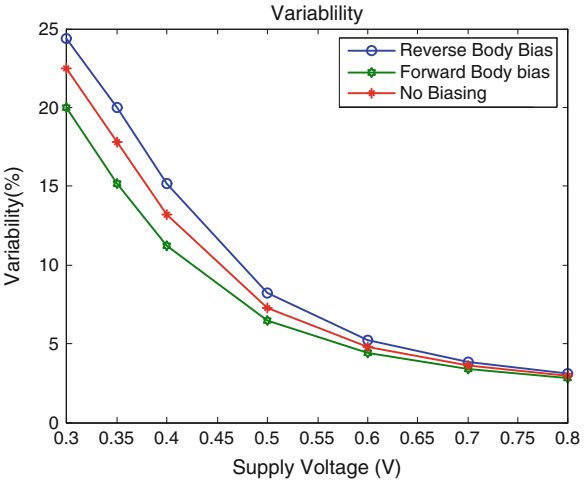

Fig. 5 Scaling of coefficient variation with supply voltage and body bias

ever, lowering the supply voltage increases the sensitivity of the circuit to process variation. This has been shown in Fig. 5 (the middle line with body bias of 0 V), which shows the CV of a ring oscillator with respect to operating voltage. The graph was obtained through Monte Carlo SPICE simulation of a ring oscillator using the setup explained previously in this chapter.

In recent years, it has been shown that the supply voltage of a CMOS circuit can be scaled even further, to voltages below the threshold voltage, which is called subthreshold operation. Various subthreshold circuits have been successfully designed, fabricated, and tested to prove the effectiveness and viability of subthreshold operation [30–32]. However, the sensitivity of the circuit to process variation increases drastically in this region. As can be seen in Fig. 5, the CV increases at a slow but steady pace as we reduce the voltage from nominal voltage to the threshold voltage, around 500 mV. However, around this point the circuit starts to show significant increase in susceptiblity toward process variation. The reason behind this is that below the threshold voltage, transistors are not switching as usual and rely heavily on leakage current for charging and discharging the load capacitance. Leakage current is more affected by process variation, which results in overall higher CV in subthreshold region. This effect is considered a drawback of subthreshold operation in general designs. However, we believe this effect can be employed to our advantage when designing a PUF circuit.

3.2.2 Body Bias Voltage

Figure 6 identifies the source, drain, gate, and bulk contacts of the PMOS and NMOS transistors in a CMOS circuit. Reverse body biasing (RBB) is the process of raising the voltage of the PMOS N-wells with respect to supply voltage or lowering the voltage of the substrate relative to ground. In a forward body bias (FBB) config-

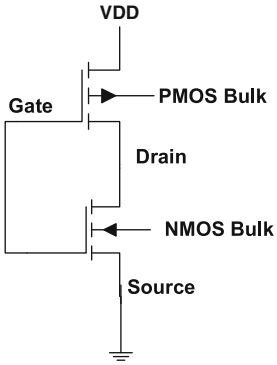

Fig. 6 A CMOS inverter with substrate nodes identified

uration, the PMOS is biased with a voltage lower than supply voltage or the voltage of the NMOS substrate is made negative relative to ground. Traditionally, RBB [33] is employed to reduce the leakage current of the circuit, thereby reducing its leakage power. But this configuration makes the design more susceptible to inherent process variations and decreases its performance. Forward body biasing on the other hand has been used for increasing the frequency of operation and making the design more tolerant toward process variation. Figure 5 shows the effect of body bias on a ring oscillator's CV.

3.3 Effect of Operating Voltage and Body Bias on PUF

Figure 7 presents the scaling of uniqueness with a varying supply voltage for a circuit with three different levels of body bias: zero, forward, and backward. It can be seen that reverse body biasing results in higher uniqueness in a PUF design. However, the effect of reducing operating voltage is much more pronounced and return of using reverse body bias is almost insignificant in subthreshold voltages. It can be concluded from this graph that operating the PUF in subthreshold voltages, which is a relatively cheap technique and requires a very small amount of hardware overhead, is a very effective approach in improving uniqueness of a PUF. This can be explained by the fact that population of frequencies of ROs in subthreshold have a much higher CV compared to a population in nominal voltage. In fact, in the subthreshold region, the circuit's sensitivity toward process variation is so high that the value of uniqueness tends to reach the theoretical maximum of 50%.

Of course, increasing a circuit's sensitivity towards process variation may also increase its sensitivity toward other sources such as variations of temperature and operating voltage. This may reduce the stability of the PUF design. Figure 8 shows the stability of the design as the temperature is varied between -15 and $+85°C$. Initially, decreasing the operating voltage of the circuit decreases the Stability under temperature variations. This is caused by RO frequencies "crossing" each other, which affects the C/R characteristic of the PUF [34]. However, below 500 mV the stability recovers. This is explained as follows. By reducing the voltage, the cir-

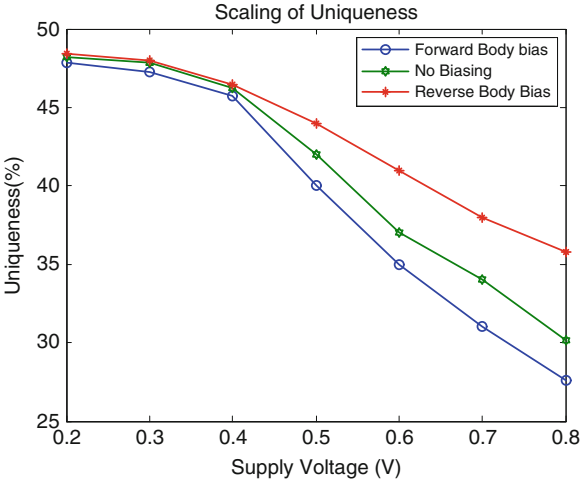

Fig. 7 Scaling of uniqueness with supply voltage for three different body bias levels

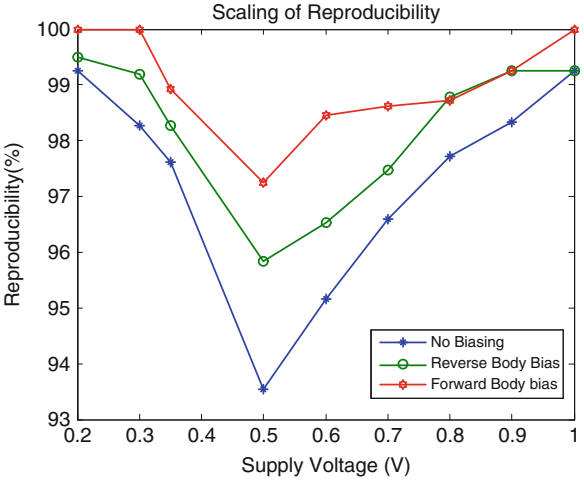

Fig. 8 A CMOS inverter with substrate nodes identified

cuit is exponentially more sensitive toward process variations (Fig. 7) exceeding its sensitivity toward temperature. Therefore, the "crossing" of RO frequencies can be suppressed, and the effects of temperature can be avoided.

4 Architecture-Level Optimization of PUF

Architecture-level optimizations on a PUF aim to optimize PUF quality factors, such as uniqueness and reliability, while at the same time minimizing the circuit cost. An architecture-level optimization distinguishes itself from a circuit-level optimiza-

tion in that it considers the architecture of the entire PUF. Therefore, it is sensitive to the spatial organization of the PUF. Architecture-level optimizations can be used to address two quality issues in PUF design: compensation of environmental effects and compensation of correlated process manufacturing variations (PMV). We describe each of these two aspects.

4.1 Compensation of Environmental Effects

The impact of environmental effects can be addressed at architectural level through redundancy. First, we should observe that the effect of a temperature change or a voltage change on a PUF is only problematic when the quantization of the response bits has a low signal-to-noise ratio. In this case, the relative magnitude of the effect of PMV is similar or smaller than the effect of environmental variations. As a result, changes in environmental conditions have a dominant effect on the PUF output.

Lim describes a redundant ring oscillator PUF as follows [13]. Each ring-oscillator is replicated four or eight times, thereby creating several redundant oscillators. Each set of redundant oscillators is one group. Next, instead of comparing individual ring oscillator frequencies, one will compare groups of ring oscillators. For each comparison, one ring oscillator is chosen from each group such that their frequency difference is maximal. This will ensure that the contribution of PMV is maximized, and hence will create a stable response.

The disadvantage of redundancy is increased architecture cost. We developed a simple, economical implementation of the redundancy technique for ring oscillators [35]. Figure 9a illustrates a configurable ring oscillator. In this design, each stage of the ring oscillator is configurable and can select one of two inverters. The design in the figure has three stages and therefore can implement eight different ring oscillators. Of particular interest for -ield-programable gate array designs is that this structure can be efficiently implemented. Figure 9a illustrates the mapping of this configurable ring oscillator in a single configurable logic block (CLB) of a Xilinx FPGA. The configurable ring oscillator occupies the same configurable area as a non-configurable ring oscillator, which means that this redundancy technique comes for free in an FPGA.

The configurable ring oscillator is used as follows. When, in a given PUF design, two oscillators A and B need to be compared, we will select A's and B's configuration such that their difference is maximal. Figure 9b illustrates the effect of this simple strategy on the PUF reliability as a function of circuit voltage and temperature, obtained over a population of five different XC3S500E FPGA's. By adaptively choosing a configuration that maximizes the frequency different, the reliability remains almost perfect. In contrast, when having a non-adaptive strategy that fixes a single, fixed configuration, the reliability is as below 70% for voltage variations and below 90% for temperature variations. Note that the overall reliability of the fixed configuration can be worse than the reliability of a single case because different environmental conditions can affect different PUF response bits.

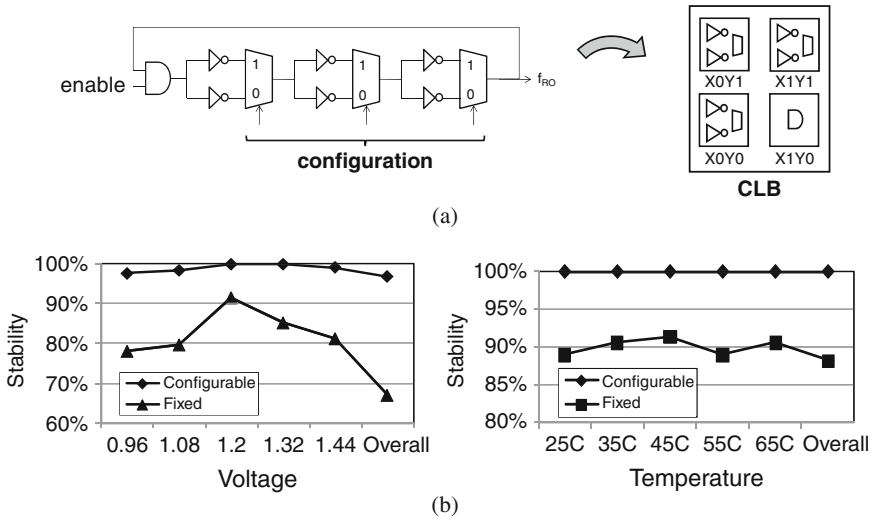

Fig. 9 (**a**) Configurable ring oscillator that maps in a single CLB; (**b**) impact of temperature and voltage on the reliability of the resulting PUF. These results were obtained over a population of 5 Spartan XC3S500E FPGA's

Another recently proposed scheme to compensate of environmental effects is the use of so-called cooperative ring oscillators [34]. In this case, a ring oscillator group is adaptively constructed, depending on the response of ring oscillators. This approach also provides significant area savings. However, since the group-forming data depends on the environmental parameter (such as temperature), it may reveal details about the internal PUF structure and therefore this data must be adequately protected.

4.2 Compensation of Correlated Process Variations

Architectural techniques are also useful to address correlated process variations. In this case, we will make use of our knowledge on the spatial organization of the PUF.

First, we briefly clarify *correlated* PMV. Figure 10a shows 256 ring oscillators arranged as a 16-by-16 matrix in an FPGA fabric. These 256 oscillators form a ring oscillator-based PUF, and the comparison of their frequencies will lead to the response bits R. If one would observe the oscillator frequencies of a chip with *ideal* PMV, the oscillator frequencies would look like Fig. 10b. In this case, the challenge/response scheme can select any two oscillators for comparison, and the comparison outcome would depend on the PMV only. In a real FPGA, however, one can observe a spatial dependency of the average ring oscillator frequency, as illustrated in Fig. 10c. This is caused by various effects, including the detailed structure of the power grid, the irregularities of the reconfigurable fabric due to hard macro's, and the systematic intra-die variations in the FPGA chip.

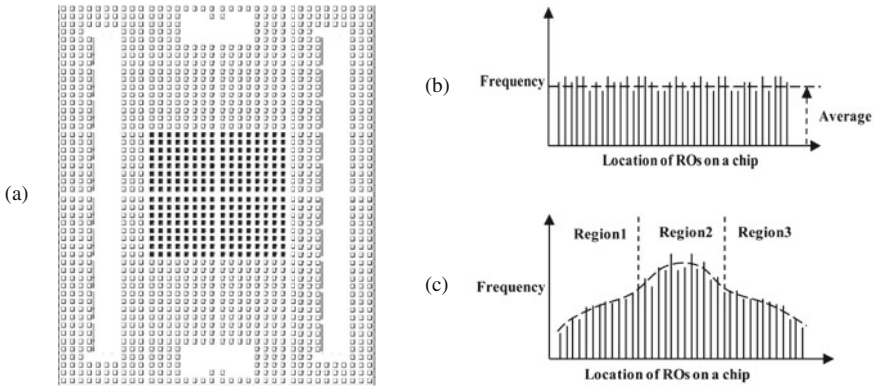

Fig. 10 **(a)** 256 ring oscillators arranged as a 16-by-16 matrix in a Spartan 3ES500 FPGA **(b)** Oscillator frequencies of an FPGA with ideal PMV **(c)** Spaital dependency of the average ring oscillator frequency of a real FPGA

Correlated PMV degrade the entropy that can be extracted from the FPGA. Considering Fig. 10c, it is clear that the comparison of ring oscillators, located far apart from each other, is likely to be biased due to systematic variations. If we need to minimize the effect of correlated variations, we should only compare frequencies of ring oscillators that are close together. Note that this simple strategy will reduce the amount of entropy as well [35].

An alternate strategy to cancel correlated variations is to make use of a common centroid technique [36]. In this case, redundant ring oscillators are used to establish a common average, and then the differential on this average is analyzed. For example, for a group of four ring oscillators with frequency A, B, C, D, the differential $(A + D) - (C + B)$ will be evaluated.

To summarize, architecture-level optimization is complementary to circuit-level optimization, and it can be used to remove environmental effects as well as correlated process variations. In the next section, we focus on the design and implementation of the identity mapping step using a test statistic and non-parametric statistics.

5 Identity Mapping and Testing

Recall that the design of a PUF scheme includes three steps: sample measurement, identity mapping, and quantization. The previous sections described strategies for sample measurement using circuit-level and architecture-level techniques. In this section, we describe a new approach to identity mapping, based on a new test statistic (TS). We will demonstrate that this TS improves over known schemes in terms of extracted entropy. Further, we will also show that the nonlinear nature of the TS provides protection against reverse engineering.

Complementary to the TS, we also propose a non-parametric hypothesis test to evaluate it, and we demonstrate experimental results obtained from five different

FPGA. We wrap up the chapter with an overview of pending challenges in Identity Mapping and Testing.

5.1 Statistical Preliminaries

We first define some notations and introduce a statistical model to explain our approach in detail. Let f_{ijl} be the frequency value for the lth measurement of the jth RO in chip i, where $i = 1, \ldots, n$, $j = 1, \ldots, m$, and $l = 1, \ldots, r$. Using these frequency data, we consider the following statistical model which is expressed as a function of unknown parameters and an additive error term,

$$f_{ijl} = f_{ij} + \varepsilon_{ijl},$$

where f_{ij} is the fixed unknown mean frequency for the ith chip and the jth RO and ε_{ijl} is a random measurement error following unknown distribution.

5.1.1 Bootstrapping the RO frequency

Since f_{ij} is unknown parameter, we need to estimate f_{ij} without assuming any distribution of ε_{ijl}. For this kind of situation, the bootstrapping approach [37] can be applicable. The bootstrapping approach is a resampling-based approach; we start from one sample which represents the unknown population from which it was drawn and then create many re-samples by repeatedly sampling with replacement. Each re-sample is the same size m as the original random sample. Using each re-sample, we estimate f_{ij} which minimizes the least square estimation criterion

$$\text{argmin}_{f_{ij}} \sum_{ijl} \varepsilon_{ijl}^2 = \text{argmin}_{f_{ij}} \sum_{ijl} (f_{ijl} - f_{ij})^2.$$

The bootstrap distribution of a f_{ij} collects its values from many re-samples. The bootstrap distribution gives information about the sampling distribution. The procedure of bootstrapping approach is summarized in Fig. 11.

Using the bootstrap distribution, we can estimate f_{ij} and also obtain a confidence interval of f_{ij} which indicates the precision with which the parameter is estimated. Because bootstrap approach can be used without assuming distribution of ε_{ijl}, it is called a *non-parametric approach*. This approach is especially useful in situation where less is known with small sample size.

5.1.2 Hypothesis Testing

Statistical test plays a key role in experimental research and aids in problem-solving and decision-making processes. Statistical test enables us to draw inferences about a phenomenon or process of interest. These inferences are obtained by using TS to draw conclusions about postulated models of the underlying data generat-

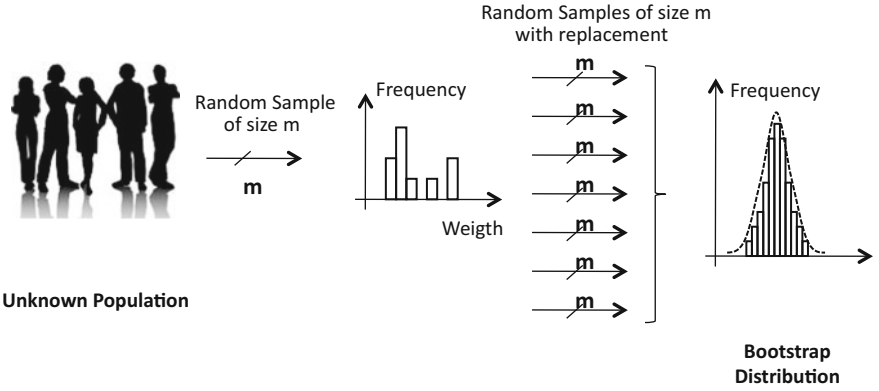

Fig. 11 The procedure of bootstrap approach

ing mechanism. To do this, we need to specify statistical hypotheses which are statements about theoretical models or about probability or sampling distributions. There are two hypotheses that must be specified in any statistical testing procedure: the *null* hypothesis and *alternative* hypothesis. In the context of a statistically designed experiment, the null hypothesis, denoted H_0, defines hypothesis of no change or experimental effect and alternative hypothesis, denoted H_a, indicates change or experimental effect. One hypothesis of interest in our study is whether chips are different from each other. Thus one set of hypotheses of interest in comparison of all chips is H_0: *the distribution of all chips are the same* vs H_a: *the distribution of all chips are different*. This set of hypotheses is equivalent to that H_0: *the distribution of any two chips are the same* vs H_a: *the distribution of some two chips are different*. To decide whether or not we reject H_0, we need to make a decision rule which is constructed by TS. Hence, for our testing, we develop a new TS and propose to use its distribution as a unique on-chip fingerprint in this chapter.

For testing our hypothesis, H_0: *the distribution of any two chips are the same* vs H_a: *the distribution of some two chips are different*, we use two statistical methods. One is bootstrap-based confidence interval and the other is Kolmogorov–Smirnov non-parametric test [38].

The first method which we implemented is the bootstrap-based confidence interval. For each chip, we calculate Q values (The Q value is the result of our proposed identity mapping step and will be explained in Sect. 5.2). Next, we obtain the difference between two Q values of two chips which we want to test whether they are different. Using bootstrapping approach, we obtain the distribution of the difference between two chips and calculate a bootstrap-based confidence interval which is 95% percentile-based confidence interval [2.5%, 97.5%] for the mean of difference between two Q values. We then obtain these confidence intervals for all possible pairs of chips. If the confidence interval does not include 0, we make a decision of rejecting H_0 which means that there are statistical evidence that any two chips are different.

The second method which we used is Kolmogorov–Smirnov test. It is used whether two underlying distributions differ. Since this test does not specify what that common distribution is, it is a non-parametric test. It is a form of minimum distance estimation used as a non-parametric test of equality of two distributions. The Kolmogorov–Smirnov statistic quantifies the distance between two distribution of Q values obtained from two chips. If the distance is larger than a critical values of the Kolmogorov distribution, we make a decision of rejecting H_0 which supports that there are statistical evidence that two chips are different.

5.2 A New Test Statistic: Q

5.2.1 Motivation

The need for a new TS can be motivated by means of an example. Let us say that we are evaluating a ring oscillator PUF design. For a given challenge, we select four ring oscillators. The response R needs to be derived from the resulting frequencies. Assume that two different chips return the following four frequencies:

$$\begin{aligned} \text{chip}_1 &= (5, 2, 1, 3), \\ \text{chip}_2 &= (9, 3, 2, 5). \end{aligned}$$

These chips are clearly different. We will show, however, that these chips are indistinguishable using the conventional approach. Indeed, the conventional approach compares the magnitude of the ring oscillator frequencies with one another, and thus builds the response R based on the *rank* of each ring oscillator frequency. The rank is the relative order of a number in a list. The frequency ranks of both chips are given as follows:

$$\begin{aligned} \text{chip}_1 &= (4, 2, 1, 3), \\ \text{chip}_2 &= (4, 2, 1, 3). \end{aligned}$$

In this case, the rank list of both chips is the same. The conventional identity mapping approach will therefore treat these two chips as identical, while they are clearly not the same! We will therefore develop a TS which can look across the frequency rank, and which directly considers the frequency values in terms of their distance. Each distance is then evaluated using a nonlinear power function, and the distribution of the resulting distance is used as the chip response R. Before giving a formal derivation, we illustrate this approach with an example.

First, we derive all frequency distances in each list. These are defined based on the creation of frequency subsets. The two chips have the following subsets:

$$\begin{aligned} \text{chip}_1 = \{&(5, 2), (5, 1), (5, 3), (2, 1), (2, 3), (1, 3), \\ &(5, 2, 1), (5, 2, 3), (5, 1, 3), (2, 1, 3), (5, 2, 1, 3)\}, \end{aligned}$$

$$\begin{aligned}\text{chip}_2 = \{&(9,3),(9,2),(9,5),(3,2),(3,5),(2,5),\\ &(9,3,2),(9,3,5),(9,2,5),(3,2,5),(9,3,2,5)\}.\end{aligned}$$

Next, we evaluate the set of distances between the frequency subsets. For example, an Euclidean metric for tuples and triples of frequencies would be as follows:

$$\begin{aligned}d(f_1,f_2) &= (f_1-f_2)^2,\\ d(f_1,f_2,f_3) &= (f_1-f_2)^2+(f_2-f_3)^2+(f_1-f_3)^2.\end{aligned}$$

This way, the set of distances for each chip leads to a distribution defined as the *Q value*:

$$\begin{aligned}Q_{\text{chip}_1} &= (9,16,4,1,1,4,26,14,24,6,36),\\ Q_{\text{chip}_2} &= (36,49,16,1,4,9,86,56,74,14,119).\end{aligned}$$

A kernel density plot of these two distributions is shown in Fig. 12. This plot shows that one distribution is sharper than the other, which suggests that the two distributions are different. Note that the Q value is not the actual bitpattern that defines the response R. The bitpattern is only obtained after the quantization step. In the following sections, we provide a formal derivation of the Q-value definition and explain the ideas that define it.

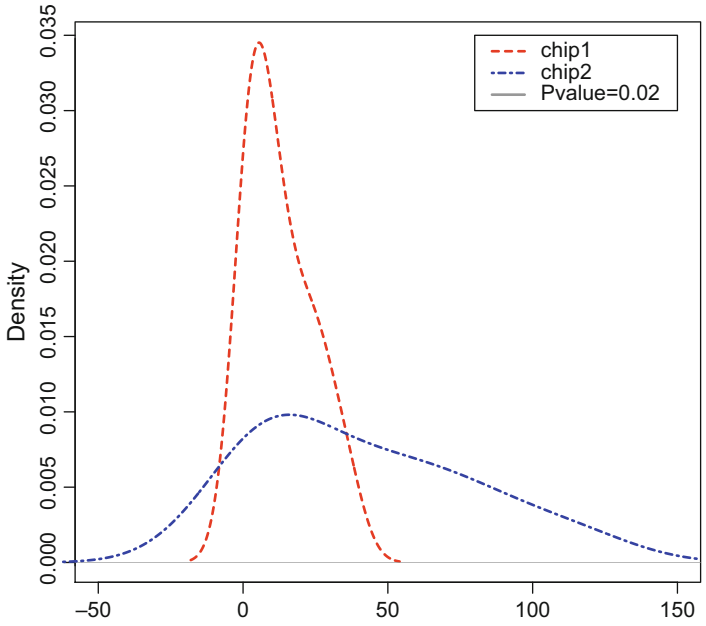

Fig. 12 Distribution of two chips in motivating example

5.2.2 Proposed Q Test Statistic

We define the sample spaces $S_2, \ldots, S_k$ which are the sets of possible outcomes of a particular experiment and random variables $Q_2, \ldots, Q_k$ which are functions mapping from the sample space to a real number (R), where $k \leq m$.

Let S_2 be the sample space which contains all possible frequency pairs. The S_3 is the sample space for all possible frequency triples. Then finally we have $S_k = \{(f_{i1}, f_{i2}, \ldots, f_{ik})\}$ for all possible k frequency values. Denoting $||\cdot||$ as the distance, let Q_k be a random variable which is a nonlinear function mapping from S_k to R,

$$Q_k : S_k \rightarrow R \text{ such that } Q_k(f_{i1}, f_{i2}, f_{i3}, \ldots, f_{ik}) = \sum_{t=1}^{\binom{k}{2}} w_{i12} \cdot ||f_{i1} - f_{i2}||^q.$$

The weight factor w_{i12} can include additional information on the design, such as the Euclidian distance between the two ROs under consideration. q may be any real number, e.g., $q = 1/2, 2$ for Euclidian distance. The choice of k depends on m, the number of RO's in the chip. While the ideal value of k is m, the construction of Q_k can be computationally expensive when m is large. In that case, we can reduce k while ensuring that the entropy of the resulting Q_k still remains large. Once q and k are chosen, our Q value is defined as $Q = (Q_2, \ldots, Q_k)$. Q can be quantized as needed to create a suitable response space. For example, we can use (1% percentile, 3% percentile, 5% percentile,...,99% percentile) of Q as the identifier of each chip, or we can use five summary values (minimum, 25% percentile, 50% percentile, 75% percentile, maximum) as the identifier. This Q value is a natural extension of the traditional method based on ordering RO frequencies. Hence, we propose this Q value for TS.

The distribution of Q values can be obtained using bootstrapping approach. We create many re-samples by repeatedly sampling with replacement. Using each re-sample, we calculate Q values. The bootstrap distribution of a Q collects its values from many re-samples. The distribution of TS can be used as a unique on-chip fingerprint.

5.2.3 Entropy of the Q Test Statistic

We now show that our approach based on the Q value has a larger entropy compared to the traditional approach based on rank. The most obvious effect of increased entropy is that the wordlength of the response R can grow. Thus, for a given PUF architecture, the probability of aliasing reduces, which means that the C/R space is more efficiently used. The entropy of the proposed approach, $H(Q)$, is obtained from $H(Q_2)$,...,$H(Q_k)$ as follows. Since all possible subsets are ${}_mC_2 = \binom{m}{2}$ in Q_2, the probability of choosing one set with equal probability is $1/{}_mC_k$. Hence, the entropy is

$$H(Q_2) = \sum_{e=1}^{{}_mC_2} \left\{ -\frac{1}{{}_nC_2} \log \left(\frac{1}{{}_mC_2} \right) \right\} = \log {}_mC_2.$$

Similarly, we can calculate $H(Q_3) = \log {}_mC_3$, ..., $H(Q_k) = \log {}_mC_k$. Then the entropy $H(Q)$ is

$$\begin{aligned} H(Q) &= \sum_{e=1}^{{}_mC_2\ {}_mC_3 \cdots {}_mC_k} \left\{ -\frac{1}{{}_mC_2\ {}_mC_3 \cdots {}_m\ C_k} \right\} \log \left(\frac{1}{{}_mC_2\ {}_mC_3 \cdots {}_mC_k} \right) \\ &= \log {}_mC_2 + \cdots + \log {}_mC_k = H(Q_2) + \cdots + H(Q_k) \\ &= \log_2 ({}_mC_2 \cdot {}_m\ C_3 \cdots {}_m\ C_k) \end{aligned}$$

On the other hand, the entropy of a traditional RO-based PUF is $\log {}_mC_2$. The entropy for an arbited-based PUF is given by its challenge space 2^m. Hence, the probability choosing one of them with equal probability is $1/2^m$. The entropy $H(T_d)$ of an arbiter-based approach is therefore

$$H(T_d) = \sum_{e=1}^{2^m} \left(-\frac{1}{2^m} \right) \log_2 \left(\frac{1}{2^m} \right) = \log_2 2^m = m.$$

The comparison between traditional approach and our proposed approach is summarized in Table 1. This result clearly shows that our proposed approach yields a larger entropy for the PUF.

5.3 Experimental Results

We have implemented five instances of an FPGA-based PUFs which contain 128 ROs. We obtained 25 frequency measurements for each RO. For given $q = 1$ and $k = 2$, we obtained Q values and used them as a TS value. We then test the hypothesis whether the distributions of Q values are the same among chips. Using Kolmogorov–Smirnove test, all chips are distinguished well. The KS result is summarized in the Table 2 which suggests that there is strong statistical evidence that all chips are quite different because all p values for all possible comparison between any two chips are very small. A p value is the probability of obtaining a value for a test statistic that is as extreme as or more extreme than the observed value, assuming the null hypothesis is true.

In order to confirm this result, we compared bootstrap-based 95% confidence intervals of the mean of the difference between Q values of two chips, which is

Table 1 Summary of entropy comparison

Traditional approach	Entropy	Comparison	Our proposed approach
Ring oscillation	$\log_2({}_mC_2)$	$<$	$\log_2({}_mC_2\ {}_mC_3 \cdots {}_m\ C_k)$
Delay-based approach	$\log_2(2^m)$	$<$	$\log_2({}_mC_2\ {}_mC_3 \cdots {}_m\ C_k)$
Example, $m = 4$,	$\log_2(2^4)$	$<$	$\log_2\left(2^4 \cdot \frac{3}{2}\right) = \log_2({}_4C_2 \cdot {}_4\ C_3)$

Table 2 The result of Kolmogorov–Smirnov test

Chip #	Chip #	p Value of KS test	Are two chips significantly different?
1	2	1.11e-07	Yes
	3	3.75e-09	Yes
	4	1.45e-06	Yes
	5	4.35e-34	Yes
2	3	2.30e-19	Yes
	4	3.97e-13	Yes
	5	1.24e-38	Yes
3	4	3.55e-02	Yes
	5	1.95e-16	Yes
4	5	9.94e-15	Yes

p Value is the probability of obtaining a value for a test statistic that is as extreme as or more extreme than the observed value, assuming the null hypothesis is true.

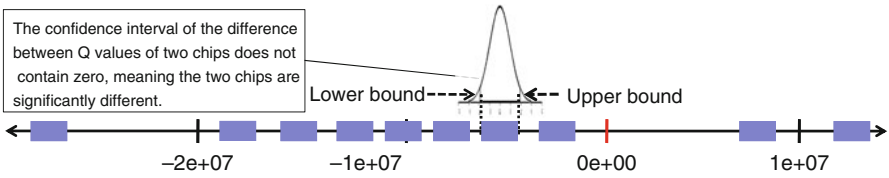

Fig. 13 95% bootstrap-based confidence interval for the mean of the difference between Q values of two chips

shown in Fig. 13. In this figure, each band represents the confidence interval. All intervals are far from 0, which strongly suggests that all chips are significantly different from each other.

5.4 Compensation of Environmental Effects

The proposed approach has three advantages compared to traditional methods. First, it is a nonlinear mapping, and it is therefore more effective against reverse engineering. Second, it is able to increase the entropy of the PUF which makes the PUF lifetime longer. Third, it is very flexible to be extended easily to control several source of variations such as temperature, voltage, and aging. For example, let us assume that f_{ijl} is measured at temperature T_{ijl}, voltage V_{ijl}, and aging A_{ijl}. Then we can model as

$$f_{ijl} = f_{ij} + \beta_1 T_{ijl} + \beta_2 \mathrm{V}_{ijl} + \beta_3 A_{ijl} + \varepsilon_{ijl}, \ \ l = 1, \ldots, r.$$

Using bootstrapping approach we can estimate f_{ij}, $\beta_1, \beta_2, \beta_3$. We then adjust frequency, $f^*_{ijl} = f_{ijl} - \hat{\beta}_1 - \hat{\beta}_2 V_{ij} - \hat{\beta}_3 A_{ij}$. Using this adjusted frequency value we can construct Q value with weight. This way, we can control the effect of several sources of variation. The resulting identifier will have a higher accuracy, and the testing can be done with higher confidence.

5.5 Open Challenges

Our present efforts address the following open challenges.

1. For a desired level of entropy, we need to select the optimal number of RO that minimizes the overall cost.
2. Q can be constructed in many ways, based on selection of measurements, the choice of q and k, and the conversion into an identifier. We need to find what exact format is the best feasible one as well as the optimal one.
3. We need to develop a testing technique to evaluate the distribution of the difference of Q values among chips. Since Q is a multidimensional quantity, not a scalar number, we also need to find the optimal testing approach. Although our experiment result shows that the distribution of the difference of Q values distinguishes all chips very well using both Kolmogorov–Smirnove statistic and bootstrap-based confidence interval, we need to investigate what testing approaches will be optimal.
4. To implement the identity mapping step, the Q test statistic formula needs to be implemented in hardware. We will develop efficient architectures to do this, based on efficient signal processing architectures. We will also evaluate the possibility of using on-chip embedded software post-processing.

6 Conclusions

The design and implementation of reliable and efficient PUF covers many different aspects, including circuit-level optimization, architecture-level optimization, and statistical analysis. Through our research, we find that a cross-disciplinary approach is important to cover this very large design space. For example, by employing subthreshold circuits, we can increase the sensitivity of the design to process manufacturing variations. By using clever redundancy at architecture level, we can then compensate any non-desirable sensitivities to environment variables (such as to temperature and operating voltage). Finally, using an appropriate test statistic, we can harvest the entropy in a statistically optimal way. Clearly, this type of design relies on a range of skills rather than a point specialty. We are currently developing prototypes of the ideas described in this chapter, using FPGA as well as ASIC technology.

Acknowledgments This work was supported in part by the Institute for Critical Technology and Applied Science (ICTAS) and the National Science Foundation with grant no. CNS-0964680.

References

1. Trusted Computing Group, TCG Trusted Network Connect - Federated TNC, 2009. http://www.trustedcomputinggroup.org/resources/federated_tnc_version_10_revision_26
2. D.D. Hwang, P. Schaumont, K. Tiri, I. Verbauwhede, Securing embedded systems. IEEE Security and Privacy, **4**(2), 40–49 (2006)

3. P.H. Kvam, B. Vidakovic, *Nonparametric Statistics with Applications to Science and Engineering* (Wiley-Interscience, Hoboken, NJ, 2007)
4. Y. Dodis, L. Reyzin, A. Smith, in *Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data*. Proceedings of EUROCRYPT'04 on Advances in Cryptology, Lecture Notes in Computer Science, vol. 3027 (Springer, Berlin, Heidelberg, 2004), pp. 523–540
5. P. Tuyls, B. Skoric, T. Kevenaar, *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting* (Springer-Verlag New York, Inc., Secaucus, NJ, 2007)
6. R. Maes, P. Tuyls, I. Verbauwhede, in *Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs*. Cryptographic Hardware and Embedded Systems - CHES 2009, Lausanne, Switzerland, 6–9 Sept 2009 (Springer Verlag, Berlin, Heidelberg, New York)
7. C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, P. Tuyls, in *Efficient Helper Data Key Extractor on FPGAs*. Cryptographic Hardware and Embedded Systems - CHES 2008, Washington, DC, USA, 10–13 Aug 2008 (Springer Verlag, Berlin, Heidelberg, New York) pp. 181–197
8. Y. Su, J. Holleman, B. Otis, in *A 1.6pj/bit 96% Stable Chip-ID Generating Circuit Using Process Variations*. Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International, Feb 2007, pp. 406–611
9. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, P. Tuyls, in *Extended Abstract: The Butterfly PUF Protecting IP on Every FPGA*. IEEE International Workshop on Hardware-Oriented Security and Trust, 2008. HOST 2008, Anaheim, CA, USA, 9 June, 2008, pp. 67–70
10. D.E. Holcomb, W.P. Burleson, K. Fu, Power-up SRAM state as an identifying fingerprint and source of true random numbers. IEEE Trans. Comput. **58**(9), 1198–1210 (Sept 2009)
11. G.E. Suh S. Devadas, in *Physical Unclonable Functions for Device Authentication and Secret Key Generation*. DAC '07: Proceedings of the 44th Annual Design Automation Conference (ACM, New York, NY, 2007), pp. 9–14
12. E. Ozturk, G. Hammouri, B. Sunar, in *Physical Unclonable Function with Tristate Buffers*. IEEE International Symposium on Circuits and Systems, 2008 (ISCAS 2008), Seattle, WA, 18–21 May 2008 (IEEE, Piscataway, NJ, 2008), pp. 3194–3197
13. D. Lim, J. Lee, B. Gassend, G. Suh, M. van Dijk, S. Devadas, Extracting secret keys from integrated circuits. IEEE Trans. VLSI Syst. **13**(10), 1200–1205 (Oct 2005)
14. R. Helinski, J. Plusquellic, Measuring power distribution system resistance variations. IEEE Trans. Semicond. Manuf. **21**(3), 444–453 (Aug 2008)
15. R. Helsinki, D. Acharyya, J. Plusquellic, in *A Physical Unclonable Function Defined Using Power Distribution System Equivalent Resistance Variations*. Proceedings of the 46th Design Automation Conference (DAC '09), San Francisco, CA, USA (ACM, New York, NY, 2009), pp. 676–681
16. M. Majzoobi, F. Koushanfar, M. Potkonjak, Techniques for design and implementation of secure reconfigurable PUFs. ACM Trans. Reconfigurable Technol. Syst. **2**(1), 1–33 (2009)
17. R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, Physical one-way functions. Science **297**(5589), 2026–2030 (2002)
18. J.D.R. Buchanan, R.P. Cowburn, A.V. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D.A. Allwood, M.T. Bryan, Forgery: 'fingerprinting' documents and packaging. Nature **436**(7050), 475–475 (2005)
19. G. Hammouri, A. Dana, B. Sunar, in *CDs Have Fingerprints Too*. Cryptographic Hardware and Embedded Systems - CHES 2009 (Springer, Heidelberg, 2009), pp. 348–362
20. G. DeJean, D. Krovski, in *RF-DNA: Radio-Frequency Certificates of Authenticity*. Cryptographic Hardware and Embedded Systems - CHES 2007 (Springer, Heidelberg, 2007), pp. 346–363
21. F. Kousanfar, A. Candore, O. Kocabas, in *Robust Stable Radiometric Fingerprinting for Wireless Devices*. IEEE International Workshop on Hardware Oriented Security and Trust 2009 (HOST 2009), San Francisco, CA, USA, July 2009, pp. 43–49

22. S. Jana, S.P. Nandha, M. Clark, S.K. Kasera, N. Patwari, S. Krishnamurty, in *On the Effectiveness of Secret Key Extraction Using Wireless Signal Strength in Real Environments*.. Proceedings of the ACM Sigmobile International Conference on Mobile Computing and Networking (MOBICOM), Beijing, 20–25 September 2009
23. B. Skoric, S. Maubach, T. Kevenaar, P. Tuyls, Information-theoretic analysis of capacitive physical unclonable functions. J. Appl. Phys. **100**(2), 024902 (2006).
24. B. Skoric, G.-J. Schrijen, W. Ophey, R. Wolters, N. Verhaegh, J. van Geloven, Experimental hardware for coating PUFs and optical PUFs. in *Security with Noise Data*, ed. by P. Tuyls, B. Skoric, T. Kevenaar (Springer, New York, NY, 2008)
25. P. Gupta, A.B. Kahng, in *Manufacturing-Aware Physical Design*. ICCAD '03: Proceedings of the 2003 IEEE/ACM International Conference on Computer-Aided Design (IEEE Computer Society, Washington, DC, 2003), p. 681
26. N. Shah, R. Samanta, M. Zhang, J. Hu, D. Walker, in *Built-In Proactive Tuning System for Circuit Aging Resilience*. IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, Cambridge, MA, USA, 1–3 October 2008, pp. 96–104
27. P. Lee, M. Kuo, P. Ko, C. Hu, *BERT - Circuit Aging Simulator (CAS)*. Technical Report UCB/ERL M90/2, EECS Department, University of California, Berkeley, 1990
28. W. Wang, V. Reddy, B. Yang, V. Balakrishnan, S. Krishnan, Y. Cao, in *Statistical Prediction of Circuit Aging Under Process Variations*. Custom Integrated Circuits Conference, 2008. CICC 2008. (IEEE, Piscataway, NJ, Sept 2008), pp. 13–16
29. UMC Foundry, http://www.umc.com. Accessed 11/2009
30. A. Wang, A. Chandrakasan, A 180-mv subthreshold FFT processor using a minimum energy design methodology. IEEE J. Solid-State Circuits **40**(1), 310–319 (Jan 2005)
31. V. Sze, R. Blazquez, M. Bhardwaj, A. Chandrakasan, in *An Energy Efficient Sub-Threshold Baseband Processor Architecture for Pulsed Ultra-Wideband Communications*. Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, vol. 3, Toulouse, 2006
32. C.H.I. Kim, H. Soeleman, K. Roy, Ultra-low-power DLMS adaptive filter for hearing aid applications. IEEE Trans.VLSI Syst. **11**(6), 1058–1067 (Dec 2003)
33. J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, V. De, in *Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage*. Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International, vol. 1, San Francisco, CA, USA, 2002, pp. 422–478
34. C.E. Yin, G. Qu, in *Temperature-Aware Cooperative Ring Oscillator PUF*. IEEE International Workshop on Hardware-Oriented Security and Trust, 2009. HOST '09, San Francisco, CA, USA, July 2009, pp. 36–42
35. A. Maiti, P. Schaumont, in *Improving the Quality of a Physical Unclonable Function Using Configurable Ring Oscillators*. 19th International Conference on Field Programmable Logic and Applications (FPL 2009), 2009
36. H. Yu, P.H.W. Leong, M. Glesner, H. Hinkelmann, L. Moller, P. Zipf, in *Towards a Unique FPGA-Based Identification Circuit Using Process Variations*. Proceedings of the 19th International Conference on Field Programmable Logic and Applications 2009 (FPL09), September 2009
37. B. Efron, R.J. Tibshirani, *An Introduction to the Bootstrap* (Chapman & Hall, London, England,1993)
38. I.M. Chakravarti, R.G. Laha, J. Roy. *Handbook of Methods of Applied Statistics*, vol. I (Wiley, New York, NY, 1967), pp. 392–394