Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions

Chapter · January 2010

DOI: 10.1007/978-3-642-14452-3_1

Roel Maes
Intrinsic-ID
38 PUBLICATIONS **1,314** CITATIONS

Physically Unclonable Functions: a Study on the State of the Art and Future Research Directions.

Roel Maes, Ingrid Verbauwhede

1 Introduction

The idea of using intrinsic random physical features to identify objects, systems and people is not new. Fingerprint identification of humans dates at least back to the nineteenth century [20] and led to the field of *biometrics*. In the eighties and nineties of the twentieth century, random patterns in paper and optical tokens were used for unique identification of currency notes and strategic arms [2, 49, 7]. A formalization of this concept was introduced in the very beginning of the twenty first century, first as physical one-way functions [38, 39], physical random functions [12] and finally as *physical(ly) unclonable functions* or PUFs[1]. In the years following this introduction, an increasing number of new types of PUFs were proposed, with a tendency towards more integrated constructions. The practical relevance of PUFs for security applications was recognized from the start, with a special focus on the promising properties of physical unclonability and tamper evidence.

Over the last couple of years, the interest in PUFs has risen substantially, making them a hot topic in the field of hardware security and leading to an expansion of published results. In this work we have made, to the best of our knowledge, an extensive overview of all PUF and PUF-like proposals up to date in an attempt to get a thorough understanding of the state of the art in this topic. Due to the wide variety of different proposals, the different measures used for assessing them and the different possible application scenarios, making an objective comparison between them is not a trivial task. In order

K.U.Leuven, ESAT/COSIC and IBBT

This work was supported by the IAP Program P6/26 BCRYPT of the Belgian State and by K.U.Leuven-BOF funding (OT/06/04). The first author's research is funded by IWT-Vlaanderen under grant number 71369.

[1] Note that there is a slight semantical difference between *physical* and *physically* unclonable functions. Further on in this work, we argue why the term *physically unclonable* is more fitting. For the remainder of this text, we will hence speak of PUFs as physically unclonable functions.

to generalize this and future overview attempts, we identify and concretize a number of properties on which different PUF proposals can be evaluated. In the process of listing the different PUFs and their properties, a number of interesting findings and future research and discussion topics will surface.

This chapter is structured as follows: after a necessary introduction in the basic PUF terminology in Sect. 2, an extensive and profound overview of all PUF and PUF-like proposals up to date is presented in Sect. 3. Based on the findings in this overview, we identify a number of fundamental PUF properties in Sect. 4 and assess them for popular PUF proposals. As a result of this comparison, we try to point out the necessary conditions for a construction to be called a PUF. After a brief overview of the basic PUF application scenarios in Sect. 5, we introduce and discuss a number of future research directions in Sect. 6. Finally, we present some concluding remarks in Sect. 7.

2 PUF Terminology and Measures

We introduce a number of commonly used terms and measures used in describing PUFs and their characteristics. We successively describe the challenge-response terminology in Sect. 2.1, the commonly used inter- and intra-distance measures in Sect. 2.2 and we already point out the problem of environmental effects and possible solutions in Sect. 2.3.

2.1 Challenges and Responses

From its naming it is clear that a PUF performs a functional operation, i.e. when queried with a certain input it produces a measurable output. We immediately stress that in most cases, a PUF is not a true *function* in the mathematical sense, since an input to a PUF may have more than one possible output. It is more appropriate to consider a PUF as a function in an engineering sense, i.e. a procedure performed by or acting upon a particular (physical) system. Typically, an input to a PUF is called a *challenge* and the output a *response*. An applied challenge and its measured response is generally called a *challenge-response pair or CRP* and the relation enforced between challenges and responses by one particular PUF is referred to as its *CRP behavior*. In a typical application scenario, a PUF is used in two distinct phases. In the first phase, generally called *enrollment*, a number of CRPs is gathered from a particular PUF and stored in a so-called *CRP database*. In the second phase or *verification*, a challenge from the CRP database is applied to the PUF and the response produced by the PUF is compared with the corresponding response from the database.

For some PUF constructions, the challenge-response functionality is implied by their construction, while for others it is less obvious and particular settings or parameters have to be explicitly indicated to act as the challenge. Also, since for most PUFs a number of post-processing steps are applied, it is not always clear at which point the response is considered. It is preferred to denote both challenges and responses as bit strings, however, this might involve some decoding and quantization, since the physically applied stimuli and measured effects are often analog quantities.

2.2 Inter- and Intra-distance Measures

The fundamental application of PUFs lies in its identification purposes. To that end, the concept of inter- versus intra-(class) distances was inherited from the theory about classification and identification. For a set of instantiations of a particular PUF construction, inter- and intra-distances are calculated as follows:

- For a particular challenge, the *inter-distance* between two different PUF instantiations is the distance between the two responses resulting from applying this challenge once to both PUFs.
- For a particular challenge, the *intra-distance* between two evaluations on one single PUF instantiation is the distance between the two responses resulting from applying this challenge twice to one PUF.

We stress that both inter- and intra-distance are measured on a pair of responses resulting from *the same challenge*. The distance measure which is used can vary depending on the nature of the response. In many cases where the response is a bit string, Hamming distance is used. Often the Hamming distance is expressed as a fraction of the length of the considered strings, and in that case one calls it *relative or fractional Hamming distance.*

The value of both inter- and intra-distance can vary depending on the challenge and the PUFs involved. For a particular type of PUF, the inter- and intra-distance characteristics are often summarized by providing histograms showing the occurrence of both distances, observed over a number of different challenges and a number of different pairs PUFs. In many cases, both histograms can be approximated by a gaussian distribution and are summarized by providing their means, respectively μ_{inter} and μ_{intra}, and when available their standard deviations, respectively σ_{inter} and σ_{intra}.

Observe that μ_{intra} expresses the notion of average *noise* on the responses, i.e. it measures the average reproducibility of a measured response with respect to an earlier observation of the same response. It is clear that we would like μ_{intra} as small as possible since this yields very reliable PUF responses. On the other hand, μ_{inter} expresses a notion of *uniqueness*, i.e. it measures the average distinguishability of two systems based on their PUF responses.

If the responses are bit strings, the best distinguishability one can achieve is if on average half of the bits differ, i.e., in case μ_{inter} is expressed as relative Hamming distance we would like it to be as close to 50% as possible. The practical use of both notions becomes clear when considering the use of the PUF for identification purposes as explained in Sect. 5.1 and a typical graphical representation is shown in Fig. 7. Note that the goals of minimizing both μ_{intra} and $|50\% - \mu_{inter}|$ can be opposing and finding an appropriate trade-off is often necessary.

2.3 Environmental Effects

Since producing a PUF response generally involves a physical measurement, there are a number of unwanted physical side-effects which could interfere. It was already pointed out in Sect. 2.2 that the same challenge applied to the same PUF does not necessarily produce the same response, giving rise to so-called intra-distance between PUF responses. This might be caused by completely random noise and measurement uncertainties which will inevitably have a random disturbing effect on the measurement. However, certain environmental factors also have a *systematic* effect on the response measurement, e.g. temperature or supply voltage in case of a PUF on an integrated circuit. Average intra-distances will probably increase when measurements are considered over (largely) varying environmental conditions. To enable a fair comparison between different results from literature, it is mentioned when μ_{intra} is obtained from measurements in a fixed or a variable environment[2].

Because environmental effects are systematic, techniques can be introduced to reduce their influence on the PUF responses. Possible options are:

- If the effects are partially linear and affect the whole device more or less equally, a differential approach can be taken,. By considering the relation (difference, ratio, ...) between two simultaneous measurements in stead of one single measurement, one obtains a much more robust measure. This technique was introduced in [12, 10] and is called *compensation.*
- The impact of environmental effects mainly depends on the exact implementation details of the PUF. Certain implementation strategies have a reduced environmental dependency [53]. Another option is to select the environmentally robust responses beforehand and ignoring the unstable ones [48].
- If PUF responses vary heavily over the range of an environmental factor, one can measure this factor with an independent on-board sensor and introduce different operation intervals, narrow enough to minimize the environmental effects within one interval [58].

[2] Whenever not explicitly mentioned, a fixed environment is assumed.

3 PUF Instantiations

In this section, we provide, to the best of our knowledge, a very thorough overview of all proposed instantiations of PUFs in literature up to now. We also take into account certain constructions which have not been labeled a PUF by their originators[3], but which we consider to possess certain PUF-like properties. We have divided this extensive list of PUFs into a number of categories, mainly based on their construction and operation principles. Note that not all proposals are discussed with the same amount of detail, mainly due to a lack of available literature or because some constructions are only mentioned for completeness. Also, within one section, the discussed proposals are sorted in no particular order.

In Sect. 3.1, we describe PUFs or PUF-like proposals whose basic operation is other than electronical. As will become clear, this includes a wide variety of different constructions. Sect. 3.2 lists a number of constructions consisting of electrical and/or electronic building blocks whose response generation is mainly based on analog measurements. Sects. 3.3 and 3.4, describe so-called digital *intrinsic PUFs*, i.e. PUFs which are embedded on an integrated circuit (IC), and of which the basic building blocks are regular digital primitives for the chosen manufacturing technology. This means that intrinsic PUFs are easy to construct, since they don't need any dedicated processing steps during manufacturing and no specialized or external equipment for their operation. For intrinsic PUFs, the measurement setup is often an inherent part of the PUF construction and is integrated on the chip. We discern two types of intrinsic PUFs, i.e. based on delay measurements (Sect. 3.3) and based on the settling state of memory elements (Sect. 3.4). To conclude, we list in Sect. 3.5 a number of conceptual constructions. Some of them are technically not really PUFs, but can be considered as closely related extensions, e.g. POKs, CPUFs and SIMPL systems. Others are true PUF proposals for which no concrete implementations have been realized, but which possess additional interesting properties distinguishing them from regular PUFs, e.g. quantum readout PUFs and reconfigurable PUFs.

3.1 Non-electronic PUFs

In this section, we give an overview of a number of constructions with PUF-like properties whose construction and/or operation is inherently non-electronic. However, very often electronic and digital techniques will be used

[3] Possibly because they were proposed before the name *PUF* had been coined, or they were introduced in fields other than cryptographic hardware, where the notion of PUFs has not yet been introduced. When the name of a PUF in the section headings is between quotation marks, it means that we have introduced this name in this work for simplicity and easy reference.

at some point anyway to process and store these PUFs' responses in an efficient manner. The common denominator *non-electronic* in this section hence only reflects the nature of the components in the system that contribute to the random structure which makes the PUF unique. It does not say anything about the measurement, processing and storage techniques which could be using electronics.

3.1.1 "Optical PUFs"

An early version of an unclonable identification system based on random optical reflection patterns, a so-called Reflective Particle Tag, was proposed in [49] well before the introduction of PUFs. They were used for the identification of strategic arms in arms control treaties.

Optical PUFs based on transparent media were proposed in [38, 39] as *physical one-way functions (POWF)*. The core element of their design is an optical token which contains an optical microstructure constructed by mixing microscopic ($500\mu m$) refractive glass spheres in a small ($10 \times 10 \times 2.54mm$) transparent epoxy plate. The token is radiated with a Helium-Neon laser and the emerging wavefront becomes very irregular due to the multiple scattering of the beam with the refractive particles. The *speckle pattern* that arises is captured by a CCD camera for digital processing. A Gabor hash is applied to the observed speckle pattern as a feature extraction procedure. The result is a string of bits representing the hash value. It is clear and was experimentally verified that even minute changes in the relative orientation of the laser beam and the token result in a completely different speckle pattern and extracted hash. The actual PUF functionality is then completed by a challenge which describes the exact orientation of the laser and the resulting Gabor hash of the arising speckle pattern as the response. The basic implementation and operation of an optical PUF is graphically represented by Fig. 1.

A number of experiments were performed in [38, 39] testing the characteristics of the constructed PUF. Four different tokens were tested using 576 distinct challenges. The inter- and intra-distance measures were evaluated for the obtained Gabor hashes. This resulted in an average inter-distance of $\mu_{inter} = 49.79\% (\sigma_{inter} = 3.3\%)$ and an average intra-distance of $\mu_{intra} = 25.25\% (\sigma_{intra} = 6.9\%)$. The information-theoretic security aspects of optical PUFs were further studied in [56, 52, 24]. Using the context-tree weighting method (CTW) [57], an average entropy content of 0.3 bit per pixel in the Gabor hash was estimated.

It is clear that the use of an optical PUF as described above is rather laborious due to the large setup involving a laser and a tedious mechanical positioning system. A more integrated design of an optical PUF, largely based on the same concepts, has been proposed in [10] and also in [51].

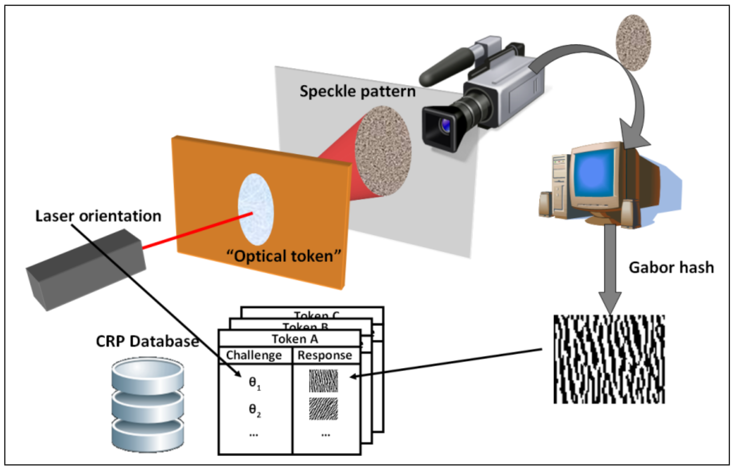


Fig. 1 Basic operation of an optical PUF.

3.1.2 "Paper PUFs"

What we call *paper PUFs* are in fact a number of proposals made in literature which basically consist of scanning the unique and random fiber structure of regular or modified paper. As with the optical PUF, also for paper PUFs there were a number of early proposals [2, 7] well before the introduction of the PUF concept, and they were mainly considered as an anti-counterfeiting strategy for currency notes. In [5], the reflection of a focused laser beam by the irregular fiber structure of a paper document is used as fingerprint of that document to prevent forgery. A similar approach is used in [6], but they explicitly introduce ultraviolet fibers in the paper during the manufacturing process which can be measured by a regular desktop scanner. They also introduce a method to strongly link the data on the document with the paper by using a combined digital signature of data and the paper's fingerprint which is printed on the document.

3.1.3 "CD PUFs"

In [17], it was observed that the measured lengths of lands and pits on a regular compact disk contain a random deviation from their intended lengths due to probabilistic variations during the manufacturing process. Moreover, this deviation is even large enough to be observed by monitoring the electrical signal of the photodetector in a regular CD player. This was tested for a large number of CDs and locations on every CD. After an elaborate quantization procedure, an average intra-distance of $\mu_{intra} = 8\%$ and an average inter-distance of $\mu_{inter} = 54\%$ on the obtained bit strings is achieved. Using the CTW method, an entropy content of 0.83 bit per extracted bit was estimated.

3.1.4 RF-DNA

A construction called radio-frequency- or RF-DNA was proposed in [8]. They construct a small $(25 \times 50 \times 3mm)$ inexpensive token comparable to the one used for the optical PUF, but now they place thin copper wires in a random way in a silicon rubber sealant. Instead of observing the scattering of light as with optical PUFs, they observe the near-field scattering of EM waves by the copper wires at other wavelengths, notably in the $5 - 6GHz$ band. The random scattering effects are measured by a prototype scanner consisting of a matrix of RF antennas. The entropy content of a single token is estimated to be at least 50000 bit.

3.1.5 "Magnetic PUFs"

Magnetic PUFs [25] use the inherent uniqueness of the particle patterns in magnetic media, e.g. in magnetic swipe cards. They are used in a commercial application to prevent credit card fraud [33].

3.1.6 Acoustical PUFs

Acoustical delay lines are components used to delay electrical signals. They convert an alternating electrical signal into a mechanical vibration and back. Acoustical PUFs [54] are constructed by observing the characteristic frequency spectrum of an acoustical delay line. A bit string is extracted by performing principle component analysis, and it is estimated that at least 160 bits of entropy can be extracted. The considered construction can constitute to an identification scheme with a false rejection rate of 10^{-4} and a false acceptance rate at most 10^{-5}.

3.2 Analog Electronic PUFs

In this section, we discuss a number of PUF constructions whose basic operation consists of an analog measurement of an electric or electronic quantity. This in contrast to the constructions in Sect. 3.1, where the measured quantity was inherently non-electronic, and to the proposals in Sects. 3.3 and 3.4, where the measurements are performed digitally, and hence without the need for analog primitives.

3.2.1 “V_T PUFs”

To the best of our knowledge, the first technique to assign a unique identification to every single instance of a regular integrated circuit, without the need for special processing steps or after-fabrication programming, was proposed in [30] and was called ICID. The operation principle is relatively simple. A number of equally designed transistors are laid out in an addressable array. The addressed transistor drives a resistive load and because of the effect of manufacturing variations on the threshold voltages (V_T) of these transistors, the current through this load will be partially random. The voltage over the load is measured and converted to a bit string with an auto-zeroing comparator. The technique was experimentally verified on 55 chips produced in $0.35\mu m$ CMOS technology. An average intra-distance under extreme environmental variations of $\mu_{intra} = 1.3\%$ was observed, while μ_{inter} was very close to 50%.

3.2.2 “Power Distribution PUFs”

In [19], a PUF was proposed based on the resistance variations in the power grid of a chip. Voltage drops and equivalent resistances in the power distribution system are measured using external instruments and it is again observed that these electrical parameters are affected by random manufacturing variability. Experimental results on chips manufactured in $65nm$ CMOS technology show $\mu_{inter} \approx 1.5\Omega$ and $\mu_{intra} \approx 0.04\Omega$ for the equivalent resistances.

3.2.3 Coating PUFs

Coating PUFs were introduced in [50] and consider the randomness of capacitance measurements in comb-shaped sensors in the top metal layer of an integrated circuit. In stead of relying solely on the random effects of manufacturing variability, random elements are explicitly introduced by means of a passive dielectric coating sprayed directly on top of the sensors. Moreover, since this coating is opaque and chemically inert, it offers strong protection against physical attacks as well. Measurement results on 36 produced chips, each with 31 sensors, show high randomness ($\mu_{inter} \approx 50\%$) and low noise ($\mu_{intra} < 5\%$), after quantization. An experimental security evaluation in [50] reveals that the coating PUF is also tamper evident, i.e. after an attack with a FIB the responses of the PUF are significantly changed. A more theoretical evaluation of coating PUFs was done in [55]. It was estimated that the entropy content of this PUF is approximately 6.6 bit per sensor. The basic implementation and operation of a coating PUF is shown in Fig. 2.

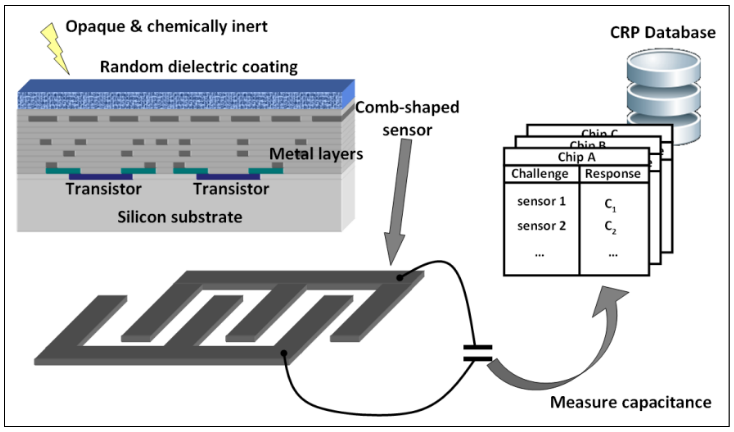


Fig. 2 Basic operation of a coating PUF. The upper left picture shows a schematic cross-section of a CMOS integrated circuit.

3.2.4 LC PUFs

An LC PUF [16] is constructed as a small ($\approx 1mm^2$) glass plate with a metal plate on each side, forming a capacitor, serially chained with a metal coil on the plate acting as an inductive component. Together they form a passive LC circuit which will absorb an amount of power when placed in a RF field. A frequency sweep reveals the resonance frequencies of the circuit, which depend on the exact values of the capacitive and inductive component. Due to manufacturing variations, this resonance peak will be slightly different for equally constructed circuits. As such, the LC PUF bares a resemblance to the coating PUF of Sect. 3.2.3 in that it measures the value of a capacitance, and to the RF-DNA of Sect. 3.1.4, in that it observes the wireless power absorption of a token during a frequency sweep over the RF field. Contrarily to RF-DNA, the LC PUF construction is intrinsically a (passive) electrical circuit and not a random arrangement of copper wire. Experimental data from 500 circuits presented in [16] shows a reproducibility of the resonance peak below $1MHz$ at a constant temperature and an entropy content between 9 and 11 bits per circuit.

3.3 Delay-based Intrinsic PUFs

In Sects. 3.1 and 3.2 a number of PUF and PUF-like proposals were discussed. They all basically start from an analog measurement of a random physical parameter, which is later quantized and can be used as an identifier of the whole system. In Sects. 3.3 and 3.4, *intrinsic PUFs* are discussed. Although no formal definition of an intrinsic PUF is provided in literature, we distinguish two prerequisites for a PUF to be called intrinsic:

1. The PUF, including the measurement equipment, should be fully integrated in the embedding device.
2. The complete PUF construction should consist of primitives which are naturally available for the manufacturing process of the embedding device.

The first condition implies that the device can query and read-out its own PUF without the need for external instruments and without the need for the challenge and response to leave the device. Note that some earlier discussed examples already meet this condition, e.g. the coating PUF or the integrated version of the optical PUF. The second condition implies that the complete PUF construction comes at virtually no additional overhead besides the space occupied by the PUF, i.e. no extra manufacturing steps or specialized components are required. This does not hold anymore for the coating PUF and the integrated optical PUF, since they both need highly specialized processing steps. A number of intrinsic PUFs have been proposed so far, all integrated on digital integrated circuits[4]. The big advantage of a PUF integrated on a digital chip is that the PUF responses can be used directly by other applications running on the same device. We distinguish two different classes, i.e. intrinsic PUFs based on digital delay measurements in this section and intrinsic PUFs based on settling memory elements in Sect. 3.4.

3.3.1 Arbiter PUFs

The initial proposal of an arbiter PUF was made in [29, 28]. The basic idea is to introduce a digital race condition on two paths on a chip and to have a so-called *arbiter* circuit decide which of the two paths *won* the race. If the two paths are designed symmetrically, i.e. with the same intended delay, then the outcome of the race is not fixed beforehand. During production of the chip, manufacturing variations will have an effect on the physical parameters determining the exact delay of each path, and causing a small random offset between the two delays. This leads to a random and possibly device-specific outcome of the arbiter and hence explains the PUF behavior of such a construction. If the offset is too small, the setup-hold time of the arbiter circuit will be violated and its output will not depend on the outcome of the race anymore, but be determined by random noise. This last phenomenon is called *metastability* of the arbiter and introduces noise in the PUF responses.

The initial design of [29, 28] uses so-called *switch blocks* to construct the two symmetrical paths and a latch or flip-flop to implement the arbiter circuit.

[4] Note that we do not use the term *silicon PUFs* in this work. It has been used to describe (a class of) PUFs which can be implemented on silicon digital integrated circuits and use the intrinsic manufacturing variability in the production process as a source of randomness. As such, they can be considered a particular case of intrinsic PUFs.

The switch blocks each have two inputs and two outputs and based on a parameter bit, they are connected *straight* or *switched*. Connecting a number of switch blocks in series creates two parameterizable delay lines feeding into the arbiter. The setting of the switch blocks will be the challenge of the PUF and the output of the arbiter the response. Note that the number of possible challenges is exponential in the number of switch blocks used. The basic arbiter PUF construction is schematically described in Fig. 3. This design was implemented on ASIC, chaining 64 switch blocks. Experimental validation on 37 chips shows $\mu_{inter} = 23\%$ and $\mu_{intra} < 5\%$, even under considerable variations of temperature and supply voltage. Equivalent tests on FPGA show much less unique randomness ($\mu_{inter} = 1.05\%, \mu_{intra} = 0.3\%$), probably due to the discrete routing constraints implied by the FPGA architecture.

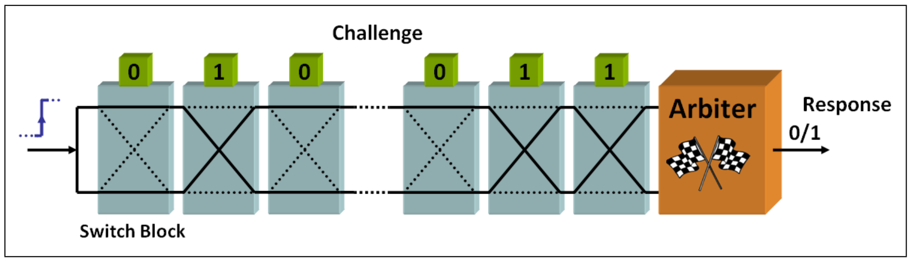


Fig. 3 Basic operation of an arbiter PUF.

Simultaneously with the introduction of delay based PUFs, it was recognized that digital delay is additive by nature, e.g. in case of the arbiter PUF from [29, 28], the delay of the chain of switch blocks will be the sum of the delays of the separate blocks. This observation leads to so-called *model-building attacks* [12, 10, 28, 29], i.e. one can build a mathematical model of the PUF which, after observing a number of CRP queries, is able to predict the response to an unseen challenge with relatively high accuracy. Such an attack was shown feasible for the basic arbiter PUF design in [28, 29, 13] using simple machine-learning techniques, achieving a prediction error of 3.55% after observing 5000 CRPs for the ASIC implementation, and a prediction error of 0.6% after observing 90000 CRPs for the FPGA implementation. All subsequent work on arbiter PUFs is basically an attempt to make model-building attacks more difficult, by introducing non-linearities in the delays and by controlling and/or restricting the inputs and outputs to the PUF.

Feed-forward arbiter PUFs [29] were a first attempt to introduce non-linearities in the delay lines. It is an extension to a regular arbiter PUF, where some challenge bits are not set by the user, but are the outcomes of intermediate arbiters evaluating the race at some intermediate point in the delay lines. This was equivalently tested on ASIC leading to $\mu_{inter} = 38\%$ and $\mu_{intra} = 9.8\%$. Note that the responses are much noisier, which is probably caused by the increased metastability since there are multiple arbiters involved. It was shown that the simple model-building attacks which succeeded in predicting the simple arbiter don't work any longer for this non-linear arbiter PUF. However, later results [34, 43] show that with more advanced modelling techniques it is still possible to build an accurate model for the feed-forward arbiter PUF, e.g. [43] achieves a prediction error of less than 5% after observing 49000 CRPs from a simulated design.

In [35], an elaborate attempt to construct a secure arbiter-based PUF on FPGA was discussed. They use an initial device characterization step to choose the optimal parameters for a particular instantiation and use the reconfiguration possibilities of FPGAs to implement this[5]. To increase randomness and to thwart model-building attacks, they use hard-to-invert input and output networks controlling the inputs and outputs to the PUF, although these are not shown cryptographically secure. By simulation, they show that this construction gives desirable PUF properties and makes model-building much harder. However, in [43] it was again shown that model-building of these elaborate structures might be feasible. They present a model of a slightly simplified structure as the one proposed in [35], which achieves a prediction error of 1.3% after observing 50000 CRPs from a simulated design.

Finally, a different approach towards model-building attacks for arbiter PUFs was taken in [37, 36, 18]. In stead of preventing the attack, they use the fact that a model of the PUF can be constructed relatively easy to their advantage. They adapt a Hopper-Blum style protocol [23] to incorporate a modelable arbiter PUF.

3.3.2 Ring Oscillator PUFs

Ring oscillator PUFs, as introduced in [12, 10], use a different approach towards measuring small random delay deviations caused by manufacturing variability. The output of a digital delay line is inverted and fed back to its input, creating an asynchronously oscillating loop, also called a *ring oscillator*. It is evident that the frequency of this oscillator is precisely determined by the exact delay of the delay line. Measuring the frequency is hence equivalent to measuring the delay, and due to random manufacturing variations on the delay, the exact frequency will also be partially random and device-dependent. Frequency measurements can be done relatively easy using digital components: an edge detector detects rising edges in the periodical oscillation and a digital counter counts the number of edges over a period of time. The counter value contains all the details of the desired measure and is considered the PUF response. If the delay line is parameterizable as with the basic arbiter PUF design, the particular delay setting is again considered the

[5] Note that there are different meanings given to the term *reconfigurable PUF*. The interpretation used in this work is the one described in Sect. 3.5.3 and is *not* directly related to the use of reconfigurable logic devices like FPGAs as meant in citeMKP09.

challenge. The basic building blocks of the simple ring oscillator construction are shown in Fig. 4.

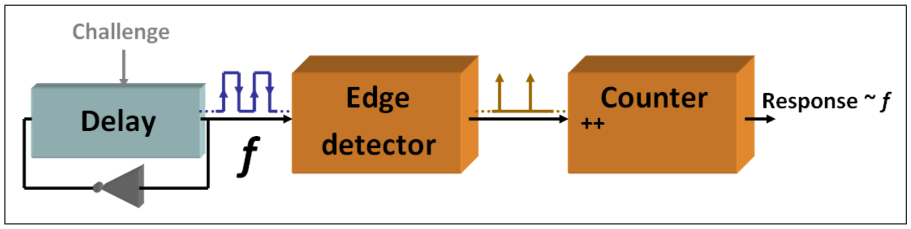
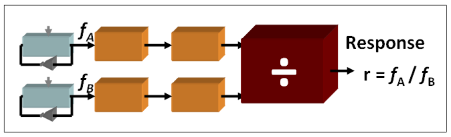


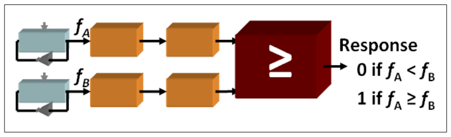
Fig. 4 Basic operation of a ring oscillator PUF.

As explained in Sect. 2.3, some environmental parameters might undesirably affect the PUF responses. In case of delay measurements on integrated circuits, the die temperature and the supply voltage heavily affect the exact delay. For arbiter PUFs, this effect was not so big since they implicitly perform a differential measurement by considering two parallel delay paths simultaneously. For ring oscillator PUFs, these effect are much larger and some sort of compensation is needed. In [12, 10], the proposed compensation technique is to divide the counter values of two simultaneously measured oscillations, which leads to much more robust responses. This compensation technique is shown in Fig. 5(a). They tested a ring oscillator PUF with division compensation on 4 FPGA devices obtaining $\mu_{inter} \approx 10 \cdot 10^{-3}$ and $\mu_{intra} \approx 0.1 \cdot 10^{-3}$ with measurements taken over a $25°C$ temperature interval. It was also shown that supply voltage variations increase μ_{intra} with another $0.003 \cdot 10^{-3}$ per mV variation. They use the same delay circuit as in the basic arbiter PUF design from [29, 28] which is hence also susceptible to model-building attacks. Moreover, it has been shown in [32] that in that case, there exists a high correlation, both between responses coming from the same challenge on different FPGAs and responses on the same FPGA coming from different challenges.

In [48], a slightly different approach was taken. The basic frequency measurement by counting rising edges is the same, but now a very simple and fixed delay circuit is used. A number of oscillators with the same intended frequency are implemented in parallel. The challenge to the PUF selects a pair of oscillators and the response is produced by comparing the two obtained counter values. This is a very simple and low-cost form of compensation and is shown in Fig. 5(b). Experiments on 15 FPGAs with 1024 loops per FPGA lead to $\mu_{inter} = 46.15\%$ and $\mu_{intra} = 0.48\%$. It has to be remarked that in order to obtain these results the authors used a technique called 1-out-of-8 masking, which considers only the most stable response bit from 8 loop pairs. This improves the reproducibility drastically and hence decreases μ_{intra}, but comes at the cost of a relatively large implementation overhed, i.e. 7 out of 8 loop pairs are unused.



(a) Ring oscillator PUF with division compensation.



(b) Ring oscillator PUF with comparator compensation.

Fig. 5

3.4 Memory-based Intrinsic PUFs

In this section we discuss another type of intrinsic PUFs, based on the settling state of digital memory primitives. A digital memory cell is typically a digital circuit with more than one logically stable state. By residing in one of its stable states it can store information, e.g. one binary digit in case of two possible stable states. However, if the element is brought into an unstable state, it is not clear what will happen. It might start oscillating between unstable states, or it might converge back to one of its stable states. In the latter case, it is observed that particular cells heavily prefer certain stable states over others. Moreover, this effect can often not be explained by the logic implementation of the cell, but it turns out that internal physical mismatch, e.g. caused by manufacturing variation, plays a role in this. For this reason, the stable settling state of a destabilized memory cell is a good candidate for a PUF response. We discuss different proposals from literature, based on different kinds of memory cells such as SRAM cells, data latches and flip-flops.

3.4.1 SRAM PUFs

SRAM PUFs were proposed in [14] and a very similar concept was simultaneously presented in [21]. SRAM or static random-access memory is a type of digital memory consisting of cells each capable of storing one binary digit. An SRAM cell, as shown in Fig. 6(a), is logically constructed as two cross-coupled inverters, hence leading to two stable states. In regular CMOS technology, this circuit is implemented with 4 MOSFETs, and an additional 2 MOSFETs are used for read/write access as shown in Fig. 6(b). For performance reasons, the physical mismatch between the two symmetrical halves of the circuit (each implementing one inverter) is kept as small as possible. It is not clear from the logical description of the cell at what state it will be right after power-up of the memory, i.e. what happens when the supply voltage comes up? It is observed that some cells preferably power-up storing a zero, others preferably power-up storing a one, and some cells have no real preference, but the distribution of these three types of cells over the complete memory is random. As it turns out, the random physical mismatch in the cell, caused by manufacturing variability, determines the power-up behavior. It forces a cell to zero or one during power-up depending on the sign of the mismatch. If the mismatch is very small, the power-up state is determined by stochastical noise in the circuit and will be random without a real preference.

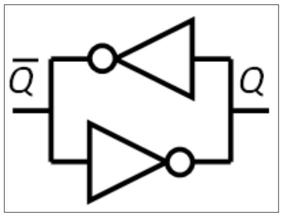
In [14], extensive experiments on SRAM PUFs were done. They collected the power-up state of 8190 bytes of SRAM from different memory blocks on different FPGAs. The results show an average inter-distance between two different blocks of $\mu_{inter} = 49.97\%$ and the average intra-distance within multiple measurements of a single block is $\mu_{intra} = 3.57\%$ for a fixed environment and $\mu_{intra} < 12\%$ for large temperature deviations. In [15], the authors estimate the entropy content of the SRAM power-up states to be 0.76 bit per SRAM cell. In [21, 22], the SRAM power-up behavior on two different platforms was studied. For 5120 blocks of 64 SRAM cells measured on 8 commercial SRAM chips, they obtained $\mu_{inter} = 43.16\%$ and $\mu_{intra} = 3.8\%$ and for 15 blocks of 64 SRAM cells from the embedded memory in 3 microcontroller chips, they obtained $\mu_{inter} = 49.34\%$ and $\mu_{intra} = 6.5\%$.

3.4.2 Butterfly PUFs

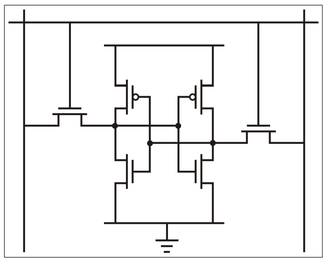
In [14], SRAM PUFs were tested on FPGAs. However, it turns out that in general this is not possible, since on the most common FPGAs, all SRAM cells are hard-reseted to zero directly after power-up and hence all randomness is lost. Another inconvenience of SRAM PUFs is that a device power-up is required to enable the response generation, which might not always be possible. To counter these two drawbacks, butterfly PUFs were introduced in [26]. The behavior of an SRAM cell is mimicked in the FPGA reconfigurable logic by cross-coupling two transparent data latches. The butterfly PUF cell construction is schematically shown in Fig. 6(d). Again, such a circuit allows two logically stable states. However, using the clear/preset-functionality of the latches, an unstable state can be introduced after which the circuit converges back to one of the two stable states. This is comparable to the convergence for SRAM cells after power-up, but without the need for an actual device power-up. Again, the preferred stabilizing state of such a butterfly PUF cell is determined by the physical mismatch between the latches and the cross-coupling interconnect. It must be noted that due to the discrete routing options of FPGAs, it is not trivial to implement the cell in such a way that the *mismatch by design* is small. This is a necessary condition if one wants the random mismatch caused by manufacturing variability to have any effect. Measurement results from [26] on 64 butterfly PUF cells on 36 FPGAs yield $\mu_{inter} \approx 50\%$ and $\mu_{intra} < 5\%$ for large temperature variations.

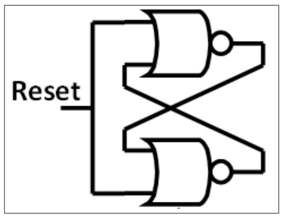
3.4.3 "Latch PUFs"

What we call a latch PUF is an IC identification technique proposed in [47] which is very similar to SRAM PUFs and butterfly PUFs. In stead of cross-coupling two inverters or two latches, two NOR-gates are cross-coupled as shown in Fig. 6(c), constituting to a simple NOR-latch. By asserting a reset signal, this latch becomes unstable and again converges to a stable state depending on the internal mismatch between the electronic components. Equivalently to SRAM PUFs and butterfly PUFs, this can be used to build a PUF. Experiments on 128 NOR-latches implemented on 19 ASICs manufactured in $0.130\mu m$ CMOS technology yield $\mu_{inter} = 50.55\%$ and $\mu_{intra} = 3.04\%$.

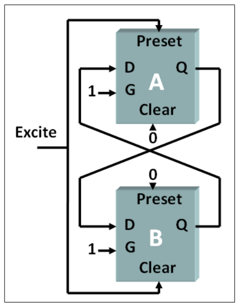


(a) Logical circuit of an SRAM (PUF) cell.


(b) Electrical circuit of an SRAM (PUF) cell in standard CMOS technology.



(c) Logical circuit of a latch (PUF) cell.



(d) Schematical circuit of a butterfly PUF cell.

Fig. 6 Comparison of the implementation of different memory-based PUF cells.

3.4.4 Flip-flop PUFs

Equivalently to SRAM PUFs, the power-up behavior of regular flip-flops can be studied. This was done in [31] for 4096 flip-flops from 3 FPGAs and gives $\mu_{inter} \approx 11\%$ and $\mu_{intra} < 1\%$. With very simple post-processing consisting of 1-out-of-9 majority voting, these characteristics improve to $\mu_{inter} \approx 50\%$ and $\mu_{intra} < 5\%$.

3.5 PUF Concepts

In the final section of this extensive overview, we discuss a number of proposed concepts which are closely related to PUFs. Some of them are generalizations or even modes of operation of PUFs. Others are actual PUF proposals for which no working implementation has been provided and whose feasibility remains yet unconfirmed.

3.5.1 POKs: Physically Obfuscated Keys

The concept of a physically obfuscated key or POK has been introduced in [10] and has been generalized to physically obfuscated algorithms in [4]. The basic notion of a POK is that a key is permanently stored in a physical way in stead of a digital way, which makes it hard for an adversary to learn the key by a probing attack. Additionally, an invasive attack on the device storing the key should destroy the key and make further use impossible, hence providing tamper evidence. It is clear that POKs and PUFs are very similar concepts and it has already been pointed out in [10] that POKs can be built from (tamper-evident) PUFs and vice versa.

3.5.2 CPUFs: Controlled PUFs

A controlled PUF or CPUF, as introduced in [11], is in fact a mode of operation for a PUF in combination with other (cryptographic) primitives. A PUF is said to be controlled if it can only be accessed via an algorithm which is physically bound to the algorithm in an inseparable way. Attempting to break the link between the PUF and the access algorithm should preferably lead to the destruction of the PUF. There are a number of advantages in turning a PUF into a CPUF:

- A (cryptographic) hash function to generate the responses of the PUF can prevent chosen-challenge attacks, e.g. to make model-building attacks more difficult. However, for arbiter PUFs it has been shown that model-building attacks work equally well for randomly picked challenges.

- An error-correction algorithm acting on the PUF measurements makes the final responses much more reliable, reducing the probability of a bit error in the response to virtually zero.
- A (cryptographic) hash function applied on the error-corrected outputs effectively breaks the link between the responses and the physical details of the PUF measurement. This makes model-building attacks much more difficult.
- The hash function generating the PUF challenges can take additional inputs, e.g. allowing to give a PUF multiple personalities. This might be desirable when the PUF is used in privacy sensitive applications to avoid tracking.

It is clear that turning a PUF into a CPUF greatly increases the security. A number of protocols using CPUFs were already proposed in [11] and more elaborate protocols were discussed in [46]. It must be stressed that the enhanced security of a CPUF strongly depends on the physical linking of the PUF with the access algorithms which can be very arbitrary and might be the weak point of a CPUF.

3.5.3 Reconfigurable PUFs

Reconfigurable PUFs or rPUFs were introduced in [27]. The basic idea behind an rPUF is that it extends the regular CRP behavior of a PUF with an additional operation called *reconfiguration*. This reconfiguration has as effect that the complete CRP behavior of the PUF is randomly and preferably irreversibly changed, hence leading to a completely new PUF. The authors of [27] propose two possible implementations of rPUFs where the reconfiguration mechanism is an actual physical reconfiguration of the randomness in the PUF. One is an extension of optical PUFs, were a strong laser beam briefly melts the optical medium, causing a rearrangement of the optical scatterers, which leads to a completely new random CRP behavior. The second proposal is based on a new type of non-volatile storage called phase change memories. Writing to such a memory consists of physically altering the phase of a small cell from cristaline to amorphous or somewhere in between, and it is read out by measuring the resistance of the cell. Since the resistance measurements are more accurate than the writing precision, the exact measured resistances can be used as responses, and rewriting the cells will change them in a random way. Both proposals are rather exotic at this moment and remain largely untested. A third option is actually a logical extension of a regular PUF. By fixing a part of a PUF's challenge with a fuse register, the PUF can be reconfigured by blowing a fuse, which optimally leads to a completely changed CRP behavior for the challenge bits controlled by the user. However, the irreversibility of such a *logical* rPUF might be questionable, since the previous CRP behavior is not actually gone, but just blocked. Possible applications enabled by rPUFs are key-zeroization, secure storage in untrusted memory and prevention of *downgrading*, e.g. of device firmware.

3.5.4 Quantum Readout PUFs

Quantum readout PUFs were proposed in [45] and present a quantum extension to regular PUFs. It is proposed to replace the regular challenges and responses of a PUF with quantum states. Because of the properties of quantum states, an adversary cannot intercept challenges and responses without changing them. This leads to the advantage that the read-out mechanism of the PUF does not need to be trusted anymore, which is the case for most regular non-quantum PUFs. Up to now, the feasibility of this proposal has not been practically verified. Moreover, it is unclear if presently existing PUFs can be easily extended to accept and produce quantum states as challenges and responses.

3.5.5 SIMPL Systems and PPUFs

A number of attempts to use PUFs as part of a public-key-like algorithm have been proposed. SIMPL systems were proposed in [41] and are an acronym for *SImulation Possible but Laborious*. Two potential implementations of such a system are discussed in [42]. A very similar concept was proposed in [3] as Public PUFs or PPUFs. Both SIMPL systems and PPUFs rely on systems (PUFs) which can be modeled, but for which evaluating the model is laborious and takes a detectable longer amount of time than the evaluation of the PUF itself.

4 PUF Properties

After the extensive overview of the wide variety of different PUF proposals in Sect. 3, it becomes clear that the notion of a physically unclonable function will be hard to capture in one single closed definition. Previous attempts at defining a PUF are often too narrow, excluding certain PUFs, or too broad, including other things than PUFs, and mostly ad hoc, i.e. giving an informal description of the perceived qualities of the proposed construction. Moreover, in many of these attempts, properties are included which are not even validated but just assumed. In this work, we will not yet attempt to come up with a more complete or formal definition of PUFs. In stead, we will first look deeper into proposed PUF properties in Sect. 4.1 and check different PUF proposals against them in Sect. 4.2. Finally, we try to detect a least common subset of necessary properties for a construction to be called a PUF in Sect. 4.3.

4.1 Property Description

Here, we will list the most important properties which we selected from different definition attempts and/or identified in the PUF proposals. Although we do not completely formalize the discussed properties, we give a hint towards a possible formalization and try to make the descriptions as clear as possible to avoid ambiguity in this and future works.

To simplify the property description, we start from a very basic classification for a PUF as a *physical challenge-response procedure*. Note that already this implicitly assigns two properties to PUFs, i.e. an instantiation of a PUF cannot merely be an abstract concept but it is always (embedded in) a physical entity, and a PUF is a procedure (not strictly a function) with some input-output functionality. Since these properties are fundamental and are immediately clear from the construction for every PUF proposal up to now, we will not discuss them further. For brevity, we use the notation $\Pi : \mathcal{X} \to \mathcal{Y} : \Pi(x) = y$ to denote the challenge-response functionality of a PUF Π.

We begin by listing seven regularly occurring properties identified from multiple attempted PUF definitions and give a concise but accurate description of what we mean by them. We immediately note that these are not completely formal properties, but a hint towards a more formal description is given. In fact, the informal parts of the property descriptions are clearly marked in sans-serif font. A more elaborate discussion on each of these properties follows directly below.

1. *Evaluatable:* given Π and x, it is easy to evaluate $y = \Pi(x)$.
2. *Unique:* $\Pi(x)$ contains some information about the identity of the physical entity embedding Π.
3. *Reproducible:* $y = \Pi(x)$ is reproducible up to a small error.
4. *Unclonable:* given Π, it is hard to construct a procedure $\Gamma \neq \Pi$ such that $\forall x \in \mathcal{X} : \Gamma(x) \approx \Pi(x)$ up to a small error.
5. *Unpredictable:* given only a set $\mathcal{Q} = \{(x_i, y_i = \Pi(x_i))\}$, it is hard to predict $y_c \approx \Pi(x_c)$ up to a small error, for x_c a random challenge such that $(x_c, \cdot) \notin \mathcal{Q}$.
6. *One-way:* given only y and Π, it is hard to find x such that $\Pi(x) = y$.
7. *Tamper evident:* altering the physical entity embedding Π transforms $\Pi \to \Pi'$ such that with high probability $\exists x \in \mathcal{X} : \Pi(x) \neq \Pi'(x)$, not even up to a small error.

We now discuss all seven properties in more detail:

1. Whether or not a PUF is evaluatable can be interpreted very broadly. From a theoretical perspective, easy can mean that we want the evaluation to be possible within polynomial time and effort. From a practical perspective, it means that we want the evaluation to induce as little overhead as possible, e.g. in the restricted timing, area, power and energy constraints of an integrated chip. Also note that if a PUF is evaluatable, it is already implied that the PUF is *constructible* to begin with. It is clear that all PUF proposals which provide experimental results are constructible and at least theoretically evaluatable. If the overhead of their evaluation is also practically considered feasible depends on the application.
2. Regarding the description of the uniqueness property, there can still be some ambiguity about the meaning of *information* and *identity*. We look at this in an information theoretic sense. If a well-defined set or population of PUF instantiations is considered, the information contained in a PUF response $\Pi(x)$ relates to the partition one can make in the population based on this response. Consecutive responses allow for smaller and smaller partitions of the population until optimally a partition with a single PUF instantiation remains, in which case the considered set of CRPs *uniquely* identifies the PUF in the population. Based on the size of the population and the characteristics of the PUF responses, such a unique identification might or might not be possible. One possible measure of uniqueness which is provided in most experimental results is the inter-distance histogram, summarized by its average value μ_{inter}.
3. The reproducibility property is clear from its description. The responses to different evaluations of the same challenge x on the same PUF Π should be close in the considered distance metric. For experimental results, this is mostly measured by the intra-distance histogram and summarized by its average value μ_{intra}. Reproducibility is the property which distinguishes PUFs from true random number generators (TRNGs).
4. As is clear from its name, unclonability is the core property of a PUF. The provided description is relatively obvious, however, there are many details to be taken into consideration. Firstly, note that the clone Γ is described as a procedure, but not necessarily a *physical* procedure, since we explicitly distinguish between *physical* and *mathematical* unclonability. If it is hard to come up with a physical entity containing another PUF $\Pi_\Gamma \neq \Pi$[6] such that $\forall x : \Pi_\Gamma(x) \approx \Pi(x)$, we say that Π is *physically unclonable*. Note that the hardness of producing a physical clone even holds for the manufacturer of the original PUF Π and is for that reason also called *manufacturer resistance*. If it is difficult to come up with an (abstract) mathematical procedure f_Γ such that $\forall x : f_\Gamma(x) \approx \Pi(x)$, we say that Π is *mathematically unclonable*. Note that physical and mathematical unclonability are fundamentally different properties since a construction can be easy to clone physically but not mathematically or vice versa. In

[6] By "$\Pi_\Gamma \neq \Pi$" here we mean that Π_Γ and Π are (embedded in) physically distinct entities.

order to be truly unclonable, Π needs to be both physically and mathematically unclonable. Again, the hardness of cloning can be considered from a theoretical and a practical point of view. Practically, cloning can be very hard or infeasible. Demonstrating theoretical unclonability on the other hand is very difficult. The only known systems which can be proven to be theoretically unclonable are based on quantum physics.

5. Unpredictability is in fact a relaxed form of unclonability. If one can correctly predict the outcome of a PUF for a random challenge, only from observing a set of CRPs, it is easy to build a mathematical clone if one has access to the full PUF. Hence, predictability implies mathematical clonability and hence clonability.
6. One-wayness is a classical property coming from cryptography. We include it since the earliest definition of PUFs describes them as a physical variant of one-way functions [38].
7. Over time, a number of notions were proposed in literature regarding tampering and security against tampering. Under tampering, we understand making permanent changes to the integrity of a physical entity. We distinguish between tamper proof systems, i.e. systems for which tampering does not reveal any useful information and tamper evident systems, i.e. systems for which tampering may be possible but leaves indelible evidence. We call a PUF tamper evident if tampering with the physical entity embedding the PUF with high probability changes the CRP behavior of the PUF.

4.2 Property Check

In this section, we will check a number of PUF proposals against all seven properties identified and discussed in Sect. 4.1. The proposals we consider are basically all proposed digital intrinsic PUFs for which concrete implementation details are available, and two well studied non-intrinsic PUFs. However, we believe that the conclusions of this study in Sect. 4.3 can be generalized to all discussed PUF proposals from Sect. 3. We begin by summarizing the most important implementation details and experimental results for the discussed PUFs in Table 1.

To draw some sensible conclusions, we have to compare these PUF proposals with some non-PUF reference cases. We check against the following three reference cases which we describe in a challenge-response-like style for easy comparison with PUFs:

- A *true random number generator*. The single challenge is the request for a random number. The response is a random number extracted from a stochastical physical process.

- A very simple *RFID-like identification protocol.* The single challenge is the request for identification. The response is an identifier string which was hard-programmed in the device by the manufacturer.
- A *public key signature scheme.* A challenge is a message string. A response is signature on that message generated using a private key which was hard-programmed by the device manufacturer.

The result of this study is shown in matrix format in Table 2. Note that we explicitly distinguish between physical and mathematical unclonability since we consider them fundamentally different notions.

4.3 Least Common Subset of PUF Properties

Looking at Table 2, we spot two properties, i.e. evaluatability and uniqueness, which hold for all discussed PUFs, and all reference cases! This means that these are necessary properties for a PUF, but they are certainly not sufficient, since they also allow programmed identifiers, public key signatures and TRNGs. A third necessary property, reproducibility, excludes TRNGs. Finally, the core property of physical unclonability completely distinguishes the PUF proposals from the reference cases based on a hard-programmed unique identifier or key. We remark that this observation elegantly justifies the naming of the primitives studied in this work, i.e. *physically unclonable functions.*

Drawing further conclusions from Table 2, we notice that mathematical unclonability is an unachievable property for most of these *naked* PUFs. However, mathematical unclonability can be greatly improved by turning these PUFs into Controlled PUFs as described in Sect. 3.5.2, e.g. to prevent exhaustive read-out and model-building. One-wayness does not seem to be a good PUF property since no single PUF turns out to be truly one-way. Even for optical PUFs, which were originally introduced as *physical one-way functions* in [38], this property is unclear. Finally, although widely believed to be one of the main advantages of PUF technology, tamper evidence was only experimentally verified for the (non-intrinsic) optical and coating PUFs.

Table 1 Overview of the most important implementation details and experimental results for the PUFs discussed in Sect. 4.2.

PUF	**Randomness**	**Challenge**	**Response**	**Experiment**	μ_{inter}	μ_{intra}	**Entropy**	**Tamper evident?**	**Model-building?**
Optical PUF [38]	Explicitly introduced	Laser orientation	Gabor hash of speckle pattern	576 CRPs on 4 tokens	49.79%	25.25%	0.3 bit/pixel [24]	Yes [38]	Difficult [52, 56]
Coating PUF [50]	Explicitly introduced	Sensor selection	Quantized capacitance measurement	31 CRPs on 36 ASICs	$\approx 50\%$	$< 5\%$	6.6 bit/sensor [55]	Yes [50]	Impossible*
Basic Arbiter PUF [29, 28]	Implicit Manuf. Var.	Delay line setting	Arbiter decision	10000 CRPs on 37 ASICs	23%	$< 5\%$ ‡	?	?	$\epsilon = 3.55\%$ for $q = 5000$ † [29]
Feed-forward Arbiter PUF [29]	Implicit Manuf. Var.	Delay line setting	Arbiter decision	10000 CRPs on 37 ASICs	38%	9.8% ‡	?	?	$\epsilon < 5\%$ for $q = 50000$ † [43]
Basic Ring Oscillator PUF [12, 10]	Implicit Manuf. Var.	Delay line setting	Division compensated counter values	many CRPs on 4 FPGAs	$\approx 1\%$	$\approx 0.01\%$ ‡	?	?	Same as basic arbiter PUF
Ring Oscillator PUF with comparator [48]	Implicit Manuf. Var.	Loop pair selection	Counter values comparison with 1-out-of-8 masking	1024 loops on 15 FPGAs	46.14%	0.48% ‡	?	?	Impossible*
SRAM PUF [14]	Implicit Manuf. Var.	SRAM address	Power-up state of addressed SRAM cell	65440 CRPs on different FPGAs	49.97%	$< 12\%$	0.76 bit/cell [15] ‡	?	Impossible*
Latch PUF [47]	Implicit Manuf. Var.	Latch selection	Settling state of destabilized NOR-latch	128 CRPs on 19 ASICs	50.55%	3.04%	?	?	Impossible*
Butterfly PUF [26]	Implicit Manuf. Var.	Butterfly cell selection	Settling state of destabilized cell	64 CRPs on 36 FPGAs	$\approx 50\%$	$< 6\%$ ‡	?	?	Impossible*

$*$ We assess model-building to be impossible for these PUFs, since the physical random elements contributing to different CRPs are to a very large extent independent from each other.
† ϵ is the prediction error after a model building attack using q (observed) queries.
‡ For these PUFs, the average observed intra-distance includes at least some minimal environmental variations.

Table 2 Property matrix of different PUF proposals. ✓ = proposal meets the property. × = proposal does not meet the property. **!** = proposal meets the property under certain conditions. **?** = it remains untested whether the proposal meets the property.

	Optical PUF	Coating PUF	Basic Arbiter PUF	Feed-forward Arbiter PUF	Basic Ring Oscillator PUF	Comparator Ring Oscillator PUF	SRAM PUF	Latch PUF	Butterfly PUF	TRNG	Simple ID protocol	Public key signature
Evaluatable	✓[1]	✓[1]	✓	✓	✓	✓	✓[2]	✓	✓	✓	✓	✓
Unique	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓[3]	✓[3]
Reproducible	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓
Physically Unclonable	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×[4]	×[4]
Mathematically Unclonable	✓	×[5]	×[6]	×[6]	×[6]	×[5]	×[5]	×[5]	×[5]	✓	×[7]	×[7]
Unpredictable	✓	✓	**!**[8]	**!**[8]	**!**[8]	✓	✓	✓	✓	✓[9]	×	✓
One-way	**?**	×[10]	×[11]	×[11]	**?**	×[11]	×[11]	×[11]	×[11]	×	×	×
Tamper Evident	✓	✓	**?**	**?**	**?**	**?**	**?**	**?**	**?**	**?**	×[12]	×[12]

1. Requires extra manufacturing steps and/or external measurement equipment.
2. Requires device power-up.
3. Requires explicit hard-programming of unique identifier or key.
4. Physically cloning a hard-programmed identifier or key is easy.
5. For these PUFs, a mathematical clone can be easily created by exhaustively reading out every CRP.
6. For these PUFs, a mathematical clone can be created by a model-building attack.
7. An adversary who knows the identifier/private key can easily forge a valid identification/signature.
8. These PUFs become increasingly easier to predict when an adversary learns more CRPs.
9. Unpredictability is a key requirement for a good TRNG.
10. Because these PUFs have so few challenges, a random challenge will with non-negligible probability invert a PUF response.
11. Because the output of these PUFs is basically one bit, a random challenge will with probability $\approx 50\%$ invert a PUF response.
12. If there is no additional tamper protection provided, hard-programmed identifiers and keys are not tamper evident.

5 PUF Application Scenarios

As a final overview part of this work, we briefly present the three classes of application scenarios which we envision for PUFs, i.e. system identification in Sect. 5.1, secret key generation in Sect. 5.2 and hardware entangled cryptography in Sect. 5.3.

5.1 System Identification

Because of their physical unclonability property, using PUFs for identification is very interesting for anti-counterfeiting technologies. PUF responses can be used directly for identification very similarly as in a biometrical identification scheme. During an enrollment phase, a number of CRPs from every PUF from the population are stored in a database, together with the identity of the physical system embedding the PUF. During identification, the verifier picks a random CRP from the CRPs stored in the database for the presented system and challenges the PUF with. If the observed response is close enough to the response in the database, the identification is successful, otherwise it fails. In order to prevent replay attacks, each CRP should be used only once for every PUF and has to be deleted from the database after the identification.

The threshold used to decide on a positive identification depends on the separation between the intra-distance and inter-distance histograms. If both histograms do not overlap, an errorless identification can be made by placing the threshold somewhere in the gap between both histograms. If they do overlap then setting the threshold amounts to making a trade-off between false-acceptance rate (FAR) and false-rejection rate (FRR). The determination of the FAR and FRR based on the overlap of the inter- and intra-distance histograms is shown in Fig. 7. The optimal choice, minimizing the sum of FAR and FRR, is achieved by setting the threshold at the intersection of both histograms, but other trade-offs might be desirable for specific applications. Additionally, it is obvious that a unique identification is only possible with high probability if the response contains enough entropy with regards to the population size[7].

5.2 Secret Key Generation

Intrinsic PUFs in integrated circuits have interesting properties for use in secret key generation and storage. Since the key is generated from intrinsic

[7] A response containing n bits of entropy optimally allows for a unique identification in a population with an average size of $2^{\frac{n}{2}}$ because of the birthday paradox.

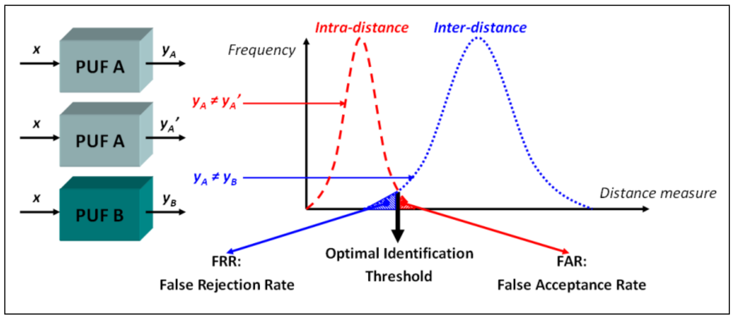


Fig. 7 Details of basic PUF based system identification. Shown is the inter- and intra-distance distribution and the determination of the FAR and the FRR based on the optimal identification threshold.

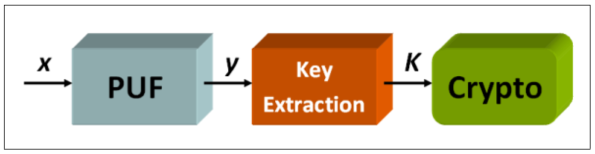
randomness introduced by inevitable manufacturing variability, no explicit key-programming step is required, which simplifies key distribution. Moreover, since this randomness is permanently fixed in the (sub-)microscopical physical details of the chip, no conventional non-volatile key memory is required. This also offers additional security against probing attacks and possibly other side-channel attacks, since the key is not permanently stored in digital format, but only appears in volatile memory when required for operation. Finally, possible tamper evidence of the PUF can be used to provide tamper proof key storage.

For cryptographic algorithms, uniformly random and perfectly reliable keys are required. Since PUF responses are usually noisy and only contain a limited amount of entropy, they cannot be used as keys directly. An intermediate processing step is required to extract a cryptographic key from the responses. This is a problem known in information theory as secret key extraction from close secrets and is generally solved by a two-phase algorithm. During the initial *generation* phase, the PUF is queried and the algorithm produces a secret key together with some additional information often called *helper data*. Both are stored in a secure database by the verifier, but not on the device. In the *reproduction* phase, the verifier presents the helper data to the algorithm which uses it to extract the same key from the PUF as in the generation step. In that way, the device containing the PUF and the verifier have established a shared secret key. It is possible to construct these algorithms such that the key is perfectly secret, even if the helper data is observed, i.e. the helper data can be publicly communicated from the verifier to the device. Practical instances of these algorithms have been proposed, e.g. in [9].

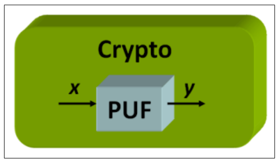
5.3 Hardware Entangled Cryptography

A recently introduced application scenario transcends the generation of secret keys from PUFs for use in existing cryptographic primitives. In stead, it fully integrates the PUF in the primitive itself, leading to so-called hardware entangled cryptographic primitives. No key is generated anymore, but the secret element of the primitive is the full unique CRP-behavior of the PUF instantiation in the embedding device. The fundamental difference between classical cryptography with a PUF-based key and hardware entangled cryptography is conceptually depicted in Fig. 8. The first result [1] based on this principle proposes a PUF-based block cipher and shows that it is possible to prove regular security notions for this construction based on reasonable assumptions for the PUF.

Hardware entangled cryptographic primitives are basically keyless, i.e. not at any point in the algorithm a secret digital key is stored in memory, neither in non-volatile nor volatile memory. Not only does this offer full security against non-volatile memory attackers, as was already the case for PUF-base secret key generation, but additionally it largely prohibits volatile memory attackers from learning anything useful. In this view, hardware entangled cryptography is closely related to the field of *provable physical security*, see e.g. [40].



(a) Classical cryptography with PUF-based secret key generation.



(b) Hardware entangled PUF-based cryptography.

Fig. 8 Schematic comparison of cryptography with PUF-based key generation and hardware entangled cryptography.

6 PUF Discussions and some Open Questions

After the overview and study of PUF instantiations, properties and application scenarios, respectively in Sects. 3, 4 and 5, we touch upon some discussion points and open questions. The field of physically unclonable functions has grown a lot over the last couple of years and is still expanding. In this section, we try to point out some interesting future research directions.

6.1 Predictability versus Implementation Size

From Tables 1 and 2, it is clear that certain PUFs suffer from predictability due to model-building attacks after a relatively small number of CRPs have been observed. This is especially the case for delay-based PUFs such as the arbiter PUF, whereas most memory-based PUFs are reasonably considered to withstand model-building attacks since their responses are based on independent random elements. On the other hand, all memory-based PUFs, and also other PUFs such as the coating PUF and the comparator-based ring oscillator PUF, suffer from another disadvantage. Their implementation size grows exponentially with the desired length of their challenges. In other words, for these PUFs the number of possible CRPs scales linearly with their size, whereas for the modelable arbiter PUF this scales exponentially. This means that for both types of digital intrinsic PUFs, the number of *unpredictable CRPs* is limited to an amount which is at best polynomial in the size of the PUF. For arbiter-like PUFs, this limitation is due to model-building attacks, whereas for memory-based PUFs it is simply because of the limited number of available CRPs.

This is a peculiar observation since it is not clear whether this is a sign of an underlying physical bound on the number of unpredictable CRPs which is obtainable for any intrinsic PUF, or whether this is merely a result of the particular PUF constructions which have been prosed thus far. Moreover, it limits the possible application scenarios, since an intrinsic PUF with a super-linear or even an exponential amount of unpredictable CRPs could lead to stronger security assumptions. From a physical point of view one could say that since the amount of (thermodynamical) entropy in a physical system is at best polynomial in the size of the system, the number of truly independent random CRPs for a single PUF can never be exponential in the PUF's size. However, for many cryptographical applications we aim for computational rather than perfect measures of security, i.e. even if the true entropy content is limited there still could be a large number of *computationally unpredictable* CRPs. In other words, model-building could be possible in theory but infeasible in practice. Further study on this topic, both from a theoretical and a practical point of view, is definitely recommended.

6.2 Formalization of PUF Properties

The properties which we studied for a number of PUF proposals in Sect. 4 were described informally in Sect. 4.1. In order to make strong claims on the security of PUFs and PUF applications, it is necessary to come up with a formalized version of these property descriptions. This formalization will act as a convenient interface between the people involved in the practical implementation of a physically unclonable function and the people designing

PUF-based security primitives and applications. The actual PUF designers should validate their constructions against the proposed properties and further focus on making them as efficient as possible. Application developers can build upon the specified properties without having to worry about the physical details of the underlying PUFs and they can use the formal nature to prove strong security notions for their proposals. Especially for the further development of hardware entangled crypto primitives, the need for a formal description of PUF qualities seems inevitable.

We acknowledge that for some of the properties discussed in Sect. 4.1, coming up with a formal definition is far from trivial. Especially the more practical properties, i.e. physical unclonability and tamper evidence, will be hard to fit into a theoretical framework. Moreover, even from a practical point of view it is not yet exactly clear what these properties stand for. With regard to tamper evidence, further experiments on intrinsic PUFs are highly recommended in order to get a better feeling of its feasibility. Physical unclonability, although considered to be the core property of PUFs, is for the moment a rather ad-hoc assumption primarily based on the apparent hardness of measuring and controlling random effects during manufacturing processes. However, for a number of intrinsic PUF proposals, it is not clear how difficult this assumption is in reality. Further research into these topics is required.

6.3 Reporting on PUF Implementation Results

In the growing body of literature on the implementation of PUFs, extensively summarized in Sect. 3, a number of different concepts and figures are used to demonstrate the practicality and security of the proposed constructions, some more useful than others. We remark that this poses a possible risk to objectivity. Firstly, without the proper argumentation it becomes rather subjective to assess one's PUF based on one's own proposed measures. Secondly, a wide variety of measures makes it difficult to objectively compare different PUF proposals. For these two reasons, it is important to agree upon a number of standardized measures which can assess the practical and security-related characteristics of differently constructed PUFs in an objective manner. For some characteristics, this is closely related to a further formalization of different PUF properties as discussed in Sect. 6.2.

We briefly discuss a number of used concepts which we consider to be important for comparison of different PUF proposals.

- *Sample size.* Before even touching upon concrete characteristics, we point out that the sample size used to estimate these characteristics is important, i.e. the number of distinct devices, the number of distinct challenges, the number of distinct measurements of every response, etc. Up to now, most works were conscientious in mentioning the used devices and the size of the sample population that was tested. However, to the best of our knowledge, for none of the PUF proposals a statistical analysis was performed pointing out the confidence level on the estimated characteristics. For further formal analysis of PUFs, this will be of increasing importance.

- *Inter- and intra-distance histograms.* The importance of both inter- and intra-distance as a measure for respectively the uniqueness and the noise present in a PUF measurement has been pointed out a number of times earlier in this work. Luckily, these two measures are almost always provided in the body of literature, making at least a partially objective comparison between early proposals possible. However, a number of remarks have to be made. Firstly, these histograms are often assumed to be approximately gaussian and are summarized by their average μ, and sometimes their standard deviation σ. It is important to validate this gaussian approximation with a statistical test. Moreover, it is highly advised to always mention both μ *and* σ since this at least allows to calculate the optimal FAR and FRR for system identification applications (Sect. 5.1). Secondly, although inter-distance gives a good *feeling* of the uniqueness of a response, it can not be used to assess the actual independent entropy present. Some PUFs, e.g. the basic arbiter PUF, which have reasonably large inter-distances suffer from predictability due to the dependence between their responses.
- *Entropy estimations.* To overcome this last problem, a number of works proposing PUFs provide an estimate of the actual entropy present in a large number of PUF responses. Two methods which are used to do this are testing the compressibility of the response strings, mostly using the context-tree weighting method [57], and running standardized randomness tests such as the Diehard test and the NIST test [44]. We remark that due to the limited length of the available responses, both methods generally offer only a low level of confidence on their outcome. In particular for the randomness tests, which are in fact designed to test the apparent randomness in the output of *pseudo*-random number generators, it is not clear whether the passing of these tests is of any significance for PUFs. Finally, we point out that both methods only estimate the independent entropy within one single PUF, i.e. how much uncertainty does an adversary have about the outcome of an unseen response, even if he has learned all other responses from that PUF. However, for a PUF to be secure, it also has to be unpredictable given responses from other PUFs. This last notion is not assessed by the considered entropy estimates.
- *Environmental influences.* As discussed in Sect. 2.3, PUF responses are subject to unwanted physical influences from their direct environment. For a PUF to be used in a practical application, these effects have to be rigorously quantified in order to prevent unforeseen failures. For intrinsic PUFs on integrated circuits, at least the influence of the die temperature and the supply voltage on the PUF's responses should be studied.
- *CRP yield and predictability.* In Sect. 6.1 we discussed the remarkable observation that for all proposed intrinsic PUFs up to now, the number of

unpredictable CRPs is limited to an amount polynomial in the size of the PUF. To assess the usefulness of a particular PUF in some applications, it is important to know how many unpredictable response bits one can optimally obtain from a PUF of a given size. This requires a further study of predictability in PUF responses.

- *Implementation cost and efficiency.* It is evident that, in order to be of any practical use, PUFs and PUF-based applications should be as cost-effective as possible. Measuring the implementation and operation cost in terms of size, speed and power consumption is an exercise which should be made for any hardware implementation and is not limited to PUFs.
- *Tamper evidence.* A number of remarks concerning the tamper evidence property of PUFs were already discussed in Sect. 4. We point out again that any claims or assumptions regarding the tamper evidence of a particular PUF construction only make sense if they are backed up by an experimental validation. A detailed description of the performed tampering experiments and the resulting effect on the PUF's CRP behavior is invaluable in that case. To the best of our knowledge, such practical results only exist for optical PUFs [38] and coating PUFs [50].

7 Conclusion

In this chapter, we tried to cover the complete field of PUF constructions up to date. From this overview, it becomes clear that a physically unclonable function is not a rigorously defined concept, but a collection of functional constructions which meet a number of naturally identifiable qualities such as uniqueness, physical unclonability and possibly tamper evidence. A more concrete and comparative study of these properties for different PUFs leads to a common subset of necessary conditions centered around the core property of physical unclonability. This study also reveals some blind spots and open questions which point to a number of interesting future research directions. From a theoretical point of view, a further formalization of the identified properties is necessary to enable the development of strong PUF-based security primitives, notably hardware entangled cryptography. On a practical level, more concrete and standardized characteristics need to be adapted and verified in order to make objective decisions possible, both in the design of new PUFs and their applications. This will lead to more competitive results on a fair basis, which naturally advances the state-of-the-art research in this field.

References

1. Armknecht, F., Maes, R., Sadeghi, A.R., Sunar, B., Tuyls, P.: Memory leakage-resilient encryption based on physically unclonable functions. In: M. Matsui (ed.) Advances in Cryptology - ASIACRYPT 2009, *Lecture Notes in Computer Science*, vol. 5912, pp. 685–702. Springer-Verlag, Tokyo,Japan (2009)
2. Bauder, D.: An anti-counterfeiting concept for currency systems. Tech. Rep. PTK-11990, Sandia National Labs, Albuquerque, NM (1983)
3. Beckmann, N., Potkonjak, M.: Hardware-based public-key cryptography with public physically unclonable functions pp. 206–220 (2009)
4. Bringer, J., Chabanne, H., Icart, T.: On physical obfuscation of cryptographic algorithms. In: INDOCRYPT '09: Proceedings of the 10th International Conference on Cryptology in India, pp. 88–103. Springer-Verlag, Berlin, Heidelberg (2009)
5. Buchanan, J.D.R., Cowburn, R.P., Jausovec, A.V., Petit, D., Seem, P., Xiong, G., Atkinson, D., Fenton, K., Allwood, D.A., Bryan, M.T.: Forgery: 'fingerprinting' documents and packaging. Nature **436**(7050), 475 (2005)
6. Bulens, P., Standaert, F.X., Quisquater, J.J.: How to strongly lonk data and its medium: the paper case. In: IET Information Security (to appear) (2010)
7. Commission on Engineering and Technical Systems (CETS): Counterfeit Deterrent Features for the Next-Generation Currency Design. The National Academic Press (1993). Appendix E
8. Dejean, G., Kirovski, D.: Rf-dna: Radio-frequency certificates of authenticity. In: CHES '07: Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems, pp. 346–363. Springer-Verlag, Berlin, Heidelberg (2007)
9. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)
10. Gassend, B.: Physical Random Functions. Master's thesis, MIT, MA, USA (2003)
11. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Controlled physical random functions. In: ACSAC '02: Proceedings of the 18th Annual Computer Security Applications Conference, p. 149. IEEE Computer Society, Washington, DC, USA (2002)
12. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Silicon physical random functions. In: ACM Conference on Computer and Communications Security, pp. 148–160. ACM Press, New York, NY, USA (2002)
13. Gassend, B., Lim, D., Clarke, D., van Dijk, M., Devadas, S.: Identification and authentication of integrated circuits: Research articles. Concurr. Comput. : Pract. Exper. **16**(11), 1077–1098 (2004)
14. Guajardo, J., Kumar, S.S., Schrijen, G.J., Tuyls, P.: FPGA intrinsic PUFs and their use for IP protection. In: Cryptographic Hardware and Embedded Systems Workshop, *LNCS*, vol. 4727, pp. 63–80 (2007)
15. Guajardo, J., Kumar, S.S., Schrijen, G.J., Tuyls, P.: Physical unclonable functions and public-key crypto for FPGA IP protection. In: Field Programmable Logic and Applications, 2007, pp. 189–195 (2007)
16. Guajardo, J., Škorić, B., Tuyls, P., Kumar, S.S., Bel, T., Blom, A.H., Schrijen, G.J.: Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions. Information Systems Frontiers **11**(1), 19–41 (2009)
17. Hammouri, G., Dana, A., Sunar, B.: CDs Have Fingerprints Too. In: CHES '09: Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems, pp. 348–362. Springer-Verlag, Berlin, Heidelberg (2009)
18. Hammouri, G., Öztürk, E., Birand, B., Sunar, B.: Unclonable lightweight authentication scheme. In: ICICS, pp. 33–48 (2008)
19. Helinski, R., Acharyya, D., Plusquellic, J.: A physical unclonable function defined using power distribution system equivalent resistance variations. In: DAC '09: Proceedings of the 46th Annual Design Automation Conference, pp. 676–681. ACM, New York, NY, USA (2009)

20. Herschel, Sir William J.: The origin of finger-printing. Oxford University Press (1916)
21. Holcomb, D.E., Burleson, W.P., Fu, K.: Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In: Proceedings of the Conference on RFID Security (2007)
22. Holcomb, D.E., Burleson, W.P., Fu, K.: Power-up sram state as an identifying fingerprint and source of true random numbers. IEEE Trans. Comput. **58**(9), 1198–1210 (2009)
23. Hopper, N., Blum, M.: A secure human-computer authentication scheme. Tech. rep. (2000)
24. Ignatenko, T., Schrijen, G.J., Škorić, B., Tuyls, P., Willems, F.M.J.: Estimating the secrecy rate of physical uncloneable functions with the context-tree weighting method. In: Proc. IEEE International Symposium on Information Theory 2006, pp. 499–503. Seattle, USA (2006)
25. Indeck, R.S., Muller, M.W.: Method and apparatus for fingerprinting magnetic media (1994). US Patent No. 5365586
26. Kumar, S., Guajardo, J., Maes, R., Schrijen, G.J., Tuyls, P.: Extended abstract: The butterfly PUF protecting IP on every FPGA. In: Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on, pp. 67–70 (2008)
27. Kursawe, K., Sadeghi, A.R., Schellekens, D., Tuyls, P., Škorić, B.: Reconfigurable physical unclonable functions – enabling technology for tamper-resistant storage. In: 2nd IEEE International Workshop on Hardware-Oriented Security and Trust - HOST 2009, pp. 22–29. IEEE, San Francisco,CA,USA (2009)
28. Lee, J.W., Lim, D., Gassend, B., Suh, G.E., van Dijk, M., Devadas, S.: A technique to build a secret key in integrated circuits for identification and authentication application. In: Proceedings of the Symposium on VLSI Circuits, pp. 176–159 (2004)
29. Lim, D.: Extracting Secret Keys from Integrated Circuits. Master's thesis, MIT, MA, USA (2004)
30. Lofstrom, K., Daasch, W.R., Taylor, D.: IC Identification Circuit Using Device Mismatch. In: In Proceedings of ISSCC 2000, pp. 372–373 (2000)
31. Maes, R., Tuyls, P., Verbauwhede, I.: Intrinsic pufs from flip-flops on reconfigurable devices. In: 3rd Benelux Workshop on Information and System Security (WISSec 2008). Eindhoven,NL (2008)
32. Maes, R., Tuyls, P., Verbauwhede, I.: Statistical analysis of silicon puf responses for device identification. In: Workshop on Secure Component and System Identification (SECSI 2008). Berlin,DE (2008)
33. MagneTek(R): MagnePrint(R). `http://www.magneprint.com/`
34. Majzoobi, M., Koushanfar, F., Potkonjak, M.: Testing techniques for hardware security. In: Test Conference, 2008. ITC 2008. IEEE International, pp. 1–10 (2008)
35. Majzoobi, M., Koushanfar, F., Potkonjak, M.: Techniques for design and implementation of secure reconfigurable pufs. ACM Trans. Reconfigurable Technol. Syst. **2**(1), 1–33 (2009)
36. Ozturk, E., Hammouri, G., Sunar, B.: Physical unclonable function with tristate buffers. In: IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008., pp. 3194–3197 (2008)
37. Öztürk, E., Hammouri, G., Sunar, B.: Towards robust low cost authentication for pervasive devices. In: PERCOM '08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications, pp. 170–178. IEEE Computer Society, Washington, DC, USA (2008)
38. Pappu, R.S.: Physical one-way functions. Ph.D. thesis, Massachusetts Institute of Technology (2001)
39. Pappu, R.S., Recht, B., Taylor, J., Gershenfeld, N.: Physical one-way functions. Science **297**, 2026–2030 (2002)
40. Pietrzak, K.: Provable Security for Physical Cryptography. Survey talk at WEWORC'09 (2009)

41. Rührmair, U.: Simpl systems: On a public key variant of physical unclonable functions. Cryptology ePrint Archive, Report 2009/255 (2009)
42. Rührmair, U., Chen, Q., Lugli, P., Schlichtmann, U., Martin Stutzmann, G.C.: Towards Electrical, Integrated Implementations of SIMPL Systems. Cryptology ePrint Archive, Report 2009/278 (2009)
43. Rührmair, U., Sölter, J., Sehnke, F.: On the foundations of physical unclonable functions. Cryptology ePrint Archive, Report 2009/277 (2009)
44. Rukhin, A., Soto, J., Nechvatal, J., Barker, E., Leigh, S., Levenson, M., Banks, D., Heckert, A., Dray, J., Vo, S., Smid, M., Vangel, M., Heckert, A., Dray, J., Iii, L.E.B.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22 (2001)
45. Škorić, B.: Quantum readout of physical unclonable functions: Remote authentication without trusted readers and authenticated quantum key exchange without initial shared secrets. Cryptology ePrint Archive, Report 2009/369 (2009)
46. Škorić, B., Makkes, M.X.: Flowchart description of security primitives for controlled physical unclonable functions. Cryptology ePrint Archive, Report 2009/328 (2009)
47. Su, Y., Holleman, J., Otis, B.: A 1.6pj/bit 96% stable chip-id generating circuit using process variations. In: Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International, pp. 406–611 (2007)
48. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Design Automation Conference, pp. 9–14. ACM Press, New York, NY, USA (2007)
49. Tolk, K.: Reflective particle technology for identification of critical components. Tech. Rep. SAND-92-1676C, Sandia National Labs, Albuquerque, NM (1992)
50. Tuyls, P., Schrijen, G.J., Škorić, B., van Geloven, J., Verhaegh, N., Wolters, R.: Read-proof hardware from protective coatings. In: Cryptographic Hardware and Embedded Systems Workshop, *LNCS*, vol. 4249, pp. 369–383. Springer (2006)
51. Tuyls, P., Škorić, B.: Physical unclonable functions for enhanced security of tokens and tags. In: ISSE 2006 Securing Electronic Busines Processes, pp. 30–37 (2006)
52. Tuyls, P., Škorić, B., Stallinga, S., Akkermans, A.H.M., Ophey, W.: Information-theoretic security analysis of physical uncloneable functions. In: Financial Cryptography and Data Security (2005)
53. Vivekraja, V., Nazhandali, L.: Circuit-level techniques for reliable physically uncloneable functions. In: HOST '09: Proceedings of the 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 30 –35 (2009)
54. Vrijaldenhoven, S.: Acoustical Physical Uncloneable Functions. Master's thesis, Technische Universiteit Eindhoven, the Netherlands (2005)
55. Škorić, B., Maubach, S., Kevenaar, T., Tuyls, P.: Information-theoretic analysis of capacitive physical unclonable functions. Journal of Applied Physics **100**(2), 024902 (2006)
56. Škorić, B., Tuyls, P., Ophey, W.: Robust key extraction from physical uncloneable functions. In: Applied Cryptography and Network Security (ACNS) 2005, *LNCS*, vol. 3531, pp. 407–422. Springer (2005)
57. Willems, F.M.J., Shtarkov, Y.M., Tjalkens, T.J.: The context tree weighting method: Basic properties. IEEE Transactions on Information Theory **41**, 653–664 (1995)
58. Yin, C.E., Qu, G.: Temperature-aware cooperative ring oscillator PUF. In: HOST '09: Proceedings of the 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 36–42. IEEE Computer Society, Washington, DC, USA (2009)