# Security based on Physical Unclonability and Disorder

Ulrich Rührmair[†,1], Srinivas Devadas[2], Farinaz Koushanfar[†,3]

[†]These two authors have equally contributed to this book chapter

[1]Computer Science, Technische Universität München
[2]Electrical Engineering and Computer Science, Massachusetts Institute of Technology
[3]Electrical and Computer Engineering, Rice University

**Abstract.** Identification, authentication, and integrity checking are important tasks for ensuring the security and protection of valuable objects, devices, programs, and data. The utilization of the microscopic, random and unclonable disorder of physical media for such security tasks has recently gained increasing attention. Wherever applicable, the harnessing of disorder can lead to intriguing advantages: First, it can avoid the permanent storage of digital secret keys in vulnerable hardware, promising to make the resulting systems more resilient against invasive and malware attacks. Second, random physical disorder has the natural feature of being very hard to clone and to forge: Fully controlling the micro- and nanoscale fabrication variations in physical media is extremely difficult and, even if possible, prohibitively expensive. Third, utilization of the natural disorder and entropy in physical systems can sometimes enable cryptographic protocols whose security does not rest on the usual unproven number-theoretic assumptions like factoring and discrete log, creating an alternate foundation for cryptography. Physical Unclonable Functions or PUFs are perhaps the best known representative of this new class of "disordered" cryptoprimitives, but there are also others. In this chapter, we provide a classification for past and ongoing work in physical disorder based security alongside with security analyses and implementation examples. We will also outline some open problems and future research opportunities in the area.

## 1 Introduction

Since the number of networked smart objects, programs, and data is constantly increasing, there is an equally growing demand to ensure the security and reliability of these units. Since they are pervasive in our daily lives, this issue has become a significant societal challenge. One central task lies in realizing secure and reliable identification, authentication, and integrity checking of these systems.

Traditional security methods based on secret digital keys often do not provide adequate solutions for this purpose. One major point of vulnerability relates to their hardware implementations and key storage: A whole host of attacks for extracting, estimating, or cloning secret keys that are stored digitally in non-volatile memory have been developed and reported over the past several years. The situation is especially problematic for embedded and mobile low power devices with a small form factor, where the adversaries can often gain full and direct access to the device. For many FPGA-based reconfigurable devices, which are increasingly growing in market share, the permanent storage of secret keys can be a problem: Integrating secure non-volatile memory (NVM) on FPGAs incurs additional costs and fabrication overhead and, thus, it is often not included. Therefore, keys have to either be stored in external memory, where they are highly vulnerable, or an additional back-up battery to power on-chip volatile storage must be used, which increases cost and system complexity. We refer interested readers to Chapter 6 of this book for a full discussion of FPGA vulnerabilities and security.

Over recent years, an alternative security approach has therefore emerged, which is based on the inherent, hard-to-forge and unique disorder of physical objects. It constitutes a promising alternative which can address the standing challenges of classical security that were described above. Two major classes of disorder-based security systems that have been proposed are *Unique Objects (UNOs)* and *Physical Unclonable Functions (PUFs)*. A Unique Object is a physical system that, upon measurement by an external apparatus, exhibits a small, fixed set of inimitable analog properties that are not similar to any other objects. It shall be impossible to intentionally fabricate a second object with the same properties, even if the properties and exact structure of the original object are known. Such properties can be referred to as the "fingerprint" of a unique object for obvious reasons. We discuss several media that exhibit such unique disorder, including paper, fibers, magnetic disks, radiowave scatterers, and optical tokens.

PUFs are the second important class of disordered systems that can be employed for reliable identification, authentication, key storage, and other security tasks. The term and acronym "PUF" for denomination of this class first appeared in [1]. In a nutshell, a PUF

is a disordered physical system $S$ that, when interrogated by a *challenge (or input, stimulus)* denoted by $C_i$, generates a unique device *response (or output)* denoted by $R_{C_i}$. This response shall depend on the applied challenge and on the specific disorder and device structure of the PUF. The unclonability requirement in the PUF definition is that it should be intractable for an adversary with physical access to create a physical or software clone of a PUF.

Both the challenge-response pairs of PUFs and the fingerprints of Unique Objects have the purpose of uniquely identifying any device with high probability. In order to realize this in practice, we need stable repeated measurements, and must be able to cope with noise and varying operational conditions. In such scenarios, error correcting codes may be used to ensure the desired operability and robustness of the system [2–7]. Other options are averaging or calibrating the device's operational conditions [8, 9].

Two important metrics that are typically applied to categorize the uniqueness and robustness of PUF responses and UNO fingerprints are *inter-device* and *intra-device* distances. Inter-device distance is often quantified as the average Hamming distance between the responses to the same challenge obtained from two different PUFs/UNOs, or the average distance between the fingerprints of two unique objects measured in the same conditions. Intra-device distance is the average Hamming distance between the responses to the same challenge applied at different times and environmental conditions to the same PUF/UNO, or the average distance between the repeatedly measured fingerprint(s) of a unique object. Ideal PUFs and UNOs should lead to large inter-device and small intra-device distances. Another key requirement for PUFs and unique objects is the entropy of the resulting responses or fingerprints. The entropy quantifies the number of independent IDs that can be generated by the same device architecture.

Despite the similarities between UNOs and PUFs, there are several important differences between them that distinguish these two security primitives (and their subclasses) from each other. This chapter provides a conceptual categorization and summary of the field of physical disorder based cryptography and security, also termed *physical cryptography* in [10]. Whenever applicable, the concepts are interleaved with examples from the contemporary literature and im-

plementation details. A number of surveys on the PUF subject are already existent, for example, [11] and a recent book with several chapters dedicated to PUFs [12]. We will cite these review articles whenever applicable, and emphasize that the concepts in this chapter are complementary to the contemporary literature in this area.

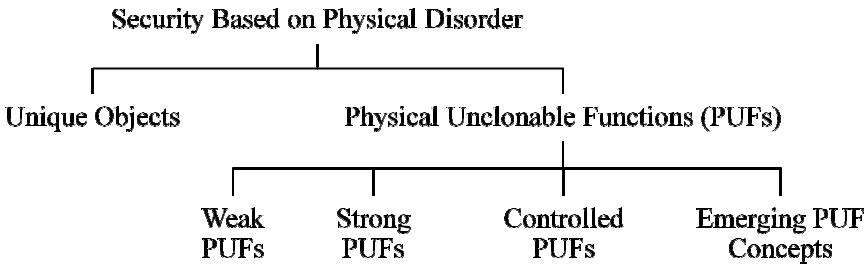


**Fig. 1.** Organization of the Chapter.

**Organization of this chapter.** Figure 1 gives an overview of the classes of physical disorder based security tokens discussed in this chapter. Each of the discussed subjects are shown as a branch in the chart. The next section reviews UNOs including paper-based (fiber-based) fingerprints, magnetic signatures, and RF-based Certificates of Authenticity. Section 3 discusses the weak PUF class including Physically-Obfuscated Keys, SRAM-PUFs, and butterfly PUFs. Strong PUFs are the subject of Section 4. Examples of PUF structures that can provide building blocks for Strong PUFs include optical PUFs, arbiter PUFs, XOR arbiter PUFs, and analog cellular arrays. Emerging PUF designs and research challenges are presented in Section 6. We conclude the chapter in Section 8.

## 2 Unique Objects

Extracting an objects's fingerprint based on its random physical disorder has been exploited for more than three decades. In absence of an established common term, we call this class of structures "Unique Objects (UNOs)".

A Unique Object is a physical entity that exhibits a small, fixed set of unique analog properties (the "fingerprint") upon being measured by an external equipment. It should be possible to measure the fingerprint quickly and preferably by an inexpensive device. The "fingerprint" should be specific to the object such that it is practically infeasible to find or build another instance of the object with the same specs, even if the object's fingerprint and its detailed structure are known (see also [13]).

More precisely, a Unique Object and its fingerprint should meet the following properties:

1. *Disorder.* The fingerprint should be based on the unique disorder of the physical object.
2. *Operability.* The fingerprint should be adequately stable over time, and must be robust to aging, environmental conditions, and repeated measurements. It must be possible to fabricate other instances of the measurement equipment with similar characterization capability. The measurement and characterization cost and time should be practically and economically viable.
3. *Unclonability.* It should be prohibitively expensive or impractical for any entity (including the manufacturer) to produce another object that presents the same unique fingerprint characteristics when queried by the measurement device.

Figure 2 demonstrates the scenario. It is assumed that each Unique Object in the figure has an unclonable fingerprint that is specific to it. Also, it is assumed that both measurement equipment are able to characterize the object's fingerprint at the desired level of resolution and accuracy. In other words, the UNO shall be the unique and unclonable part of the system, while the measurement device can be mass-produced with the same functionality.

### 2.1 History and Examples of Unique Objects

Using biometrics for fingerprinting dates back to the $19^{th}$ century. Although human hand fingerprints and other biometric entities are closely related to Unique Objects, a discussion of biometrics fingerprinting is outside the scope of this chapter. We refer the interested readers to comprehensive books on the subject [14, 15].

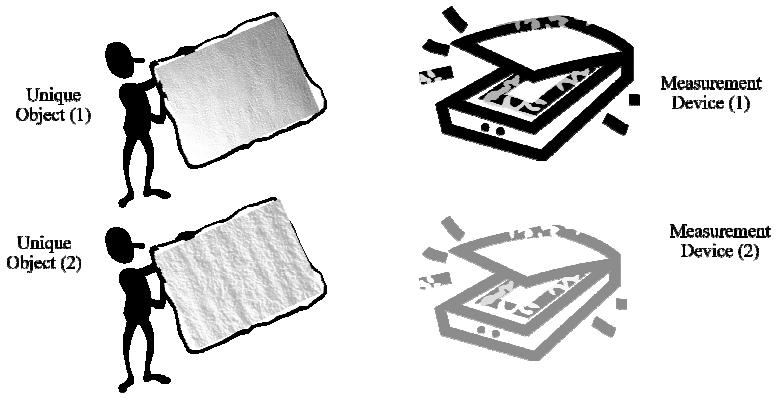


**Fig. 2.** Two Unique Objects (based on paper structures in this example), and two fingerprint measurement devices. The cloning of a Unique Object should be prohibitively costly, while it should be possible to mass-manufacture large numbers of measurement devices that can characterize the fingerprints at the desired level of accuracy.

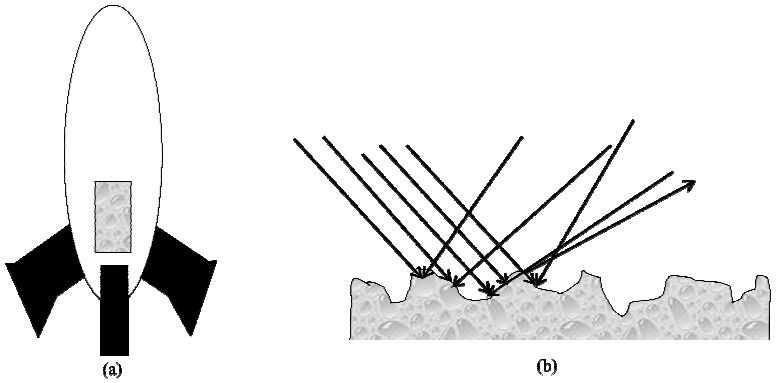


**Fig. 3.** (a) A thin, random layer of light scattering particles sprayed on the missiles. (b) Illuminating the surface from different angles would generate different inference patterns.

**Sprayed random surfaces.** Perhaps the earliest reported usage of Unique Objects for security was proposed by Bauder during the cold war for use in nuclear weapons treaties [16, 17]. To tag the nuclear missiles unforgeably, a thin, random layer of light scattering particles was sprayed onto the missiles. The layer was illuminated from various angles, and images of the resulting interference patterns were recorded by an inspector. On later inspections, the interference patterns could be measured anew, and compared to the record of the inspector. An example is shown in Figure 3.

The scheme was assumed secure even if an adversary would know the (relatively few) illumination angles and the resulting patterns used by the inspector, and even if he had access to the unique layer for a long time period and could investigate its structure. Even under these circumstances, it was presumed infeasible to produce a second layer which generated the same speckle pattern. Of course, if an adversary knows all illumination angles and the resulting patterns used by the inspector, this system cannot be used for remote authentication, since an adversary can merely replay back the digitized responses/images upon receiving a challenge. Furthermore, the scheme can only be used by an inspector who carries trusted measurement equipment and uses it on the Unique Object directly, which was presumably the usage model of the system during the cold war.

**Fiber-based random surfaces.** Other early implementations of Unique Objects were based on randomly distributed fibers in solid-state objects, for example, paper fibers in banknotes [18], or metallic fibers in a thin substrate on bank cards measured by a magnetic reader [19]. A seminal reference that discusses Unique Objects from a more fundamental cryptographic perspective is [20], which was later extended by [21]. [20] is, to our knowledge, the first academic source that suggests the combined use of Unique Objects and digital signatures to create offline verifiable labels.

Several seeds laid in this early work were followed up in later research. Firstly, surface irregularities in paper or other materials were further investigated by [22–28]. The authors of [23, 24] create unforgeable postal stamps and document authenticators from paper irregularities and digital signatures; [22] shows that the paper

surface fingerprints are robust to soaking and a number of other transformations;

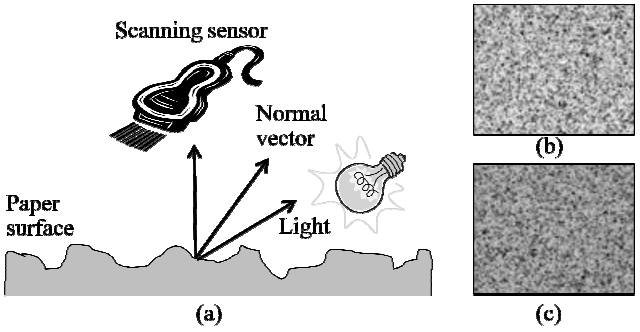


**Fig. 4.** (a) Scanner produces different images of the paper surface based on the page orientation. The light seen at the sensor depends on the angle between the surface normal and the light source; (b) A region of the document can be scanned from top-to-bottom; and (c) The same document region can be scanned from left-to-right. The 3D texture can be estimated by combining (b) and (c) (Figure inspired by studies in [28]).

[26] investigates the use of surface-based labels in developing countries and rural regions; [27] deals with database and error correcting algorithms for surface-based identification; and [28] offers a detailed treatment centering around inexpensive measurement methods for paper surfaces by commodity scanners as demonstrated in Figure 4. The complex reflective behavior of surfaces has even led to commercially available security systems [29] [30].

Secondly, the use of randomly disordered fibers contained in a fixing matrix was described in [31–34]. [32, 33] use light conducting optical fibers, and measure the individual light transfer via these fibers into various spatial segments of the matrix. Each instance is created as a collection of fibers randomly positioned in an object by a transparent gluing material that permanently fixes the fibers' positions. Readout of the random structure of a fiber-based note is performed using the following fact: if one end of a fiber is illuminated, the other end will also be lit as shown in Figure 5.

[31] employs randomly distributed metal particles, and measures their scattering behavior by a near-field read-out. [34] pours ultraviolet fibers into a paper mixture and measures the optical response.

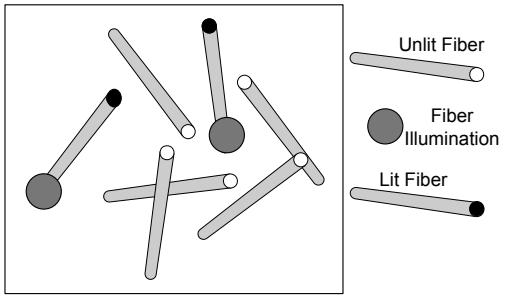


**Fig. 5.** Examples of randomly placed, fixed-length fibers. The square demonstrates the embedding substrate. Three fibers lit by spot illumination light as described in [33].

**Unique Objects and digital rights management.** An observation that further propelled the field was that common data carriers such as (again) paper, but also ICs, CDs, and DVDs can have unique features. Sometimes their unique properties arise just in the very process of writing or imprinting data onto them. One early reference in this context is [35]. There, the unique irregularities in CD-imprinting (such as individual height, shape and length of the bumps) are used to secure the CD's digital content that is stored in exactly these bumps. Conceptually the same suggestion is made in [36] and [37], yet at a much greater level of scientific detail. For more information on optical media (CD) fingerprints, we refer interested readers to [12]. The irregularities in letters printed on paper have been suggested to secure paper documents in [38]. Finally, several methods for uniquely identifying and authenticating chips will be described in the remainder of this chapter.

**Other implementations of Unique Objects.** Other studies proposed novel classes of unique structures, and can be best categorized according to the employed read-out technique. A number of Unique Objects with radio wave read-out in the (relative) far field were suggested in the commercial sector [39–44]. For most of them, doubts have been raised with respect to their unclonability [13]. Another radio wave approach measures the unique RF signals created by higher harmonic oscillations [45]. Unique Objects with magnetic read-out have been investigated in [46]. Alternative optical concepts that dig deeper into physics and utilize more complex effects and structures

have been suggested in [47]. They include photonic crystals and resonant energy transfer between optically-active particles. Surprisingly, there is little work on Unique Objects with electrical read-out, even though this would promise particularly simple and inexpensive measurement. One recent source is [10], where the unique current-voltage characteristics of imperfect diodes are exploited. Finally, even DNA-based approaches have been suggested [48], and made it to the marketplace some time ago [49].

Finally, the question of error correction of the measured unique signals is treated en passant in most of the above publications, including [27, 28, 32, 34]. References solely dedicated to error correction include [50–52].

### 2.2 Protocols and Applications of Unique Objects

The natural application of Unique Objects is to label items of value (such as commercial products, bank cards, banknotes, passports, access cards, etc.) in an unforgeable manner. Proving authenticity is particularly useful as losses due to counterfeiting of digital goods and physical objects amount to worldwide losses in a three-digit billion dollar range [53]. Two basic approaches can be applied.

(i) In one classic and straightforward approach, the Unique Object is physically attached to the item it protects, or consists of a measurable unique characteristic of the protected item itself. The Unique Object's fingerprint is stored in a central database. When authentication is needed, the object's fingerprint is measured and compared to the stored value in the database. The requirements for this protocol include existence of a central database and an authenticated online connection to the database.

(ii) An alternative approach has been pioneered, to our knowledge, in [20], and has been termed Certificate of Authenticity (COA) in [31]. Again, the Unique Object is physically attached to the protected item (or is a measurable unique characteristic of the protected item itself). In addition to the Unique Object, complementary information is stored directly on the item, for example via a printed barcode. The information includes a numerical encoding of the fingerprint, error-correcting codes [52], item-related

information $I$ (such as the origin of the item), and most importantly, a digital signature of the fingerprint and $I$. In order to verify the validity of the label/the item, a verification device does the following: It reads the complementary information from the item, and verifies the validity of the digital signature by use of a public verification key stored in the device. Secondly, it measures the fingerprint of the label/the item by a trusted measurement apparatus, and verifies if it matches the fingerprint given and signed in the complementary information.

The advantage of the approach (ii) is that it does not need a connection to a database and that it can work offline. Neither the label nor the testing apparatus needs to contain secret information of any sort. This leads to the strong asset that neither tampering with a label nor with any of the widespread testing apparatuses can lead to secret key extraction and a global failure of the system through one extracted key. The measurement apparatus has to be trusted to work correctly.

Variants and combinations of the two basic protocols above have been proposed in Unique Objects literature, e.g., [35–37].

### 2.3 Security Features

**No secret information in hardware.** The most striking feature of Unique Objects is that they contain no piece of information that must remain secret, and which would have to be protected by costly and laborious means. Their security rests not on the assumption that some key or some other information about their structure remains secret; rather, it is built on the hypothesis that it is infeasible to build a clone of the object even if all its internal details are known. It is based directly on the limitations of current nano-scale fabrication methods.

Furthermore, a COA can even be verified for validity without possession of any secret keys; any verifying party merely must hold a public key to check the digital signature contained in the COA. This allows the widespread distribution of labels and testing apparatuses, without risking a global security break through secret key compromise in either the labels or apparatuses, which is significant.

The only secret key that is required can be stored at the authority that creates the signatures, where it can usually be protected much better. The authenticated communication required in classic protocol (i) above can be established by typical cryptographic methods. At the same time, parties using the system must rely on the integrity of the measurement apparatus. This implies that remote authentication to a central authority by an untrusted terminal is not possible, and therefore limits applicability of Unique Objects.

**Structural sensitivity as a security benchmark.** One critical measure for the security of Unique Objects is their structural sensitivity: How much are the output signal and the unique measured fingerprints of the object affected if we change its inner structure slightly, by a factor of $\delta$, say? This parameter determines the level of exactness that an adversary has to reproduce the Unique Object in order to remain undetected. It can be employed as a benchmark to rank competing candidates of Unique Objects.

**Attacks on Unique Objects.** The main attack on Unique Objects is refabrication or cloning. It is not necessary to rebuild the original with perfect precision; merely, a second structure needs to be fabricated that generates the same measured fingerprint as the original from the view of the measurement apparatus. This structure could in principle have a totally different size, lengthscale, or appearance; it might even be a smart, reactive system that artificially synthesizes the correct response. Note that purely numerical modeling attacks such as the ones executed on PUFs [54] are pointless and not applicable to Unique Objects. Such attacks can help a fraudster to numerically predict the (numerical) response of a PUF to a randomly chosen challenge. But in case of UNOs, the attacker is assumed to know these responses anyway; his task lies in fabricating a clone that produces the same analog response upon measurement with an external apparatus that he/she cannot influence. This is foremost a physical manufacturing challenge, not a question of modeling.

**Quantum systems vs. Unique Objects.** Quantum systems, such as polarized photons, were among the first systems whose inherent

physical features have been suggested for security systems [55] [56]. It is long known that the laws of quantum physics forbid the cloning of a quantum system with an unknown state, for example of a photon with an unknown polarization. Could a polarized photon hence be interpreted as a specific object according to our definition, with its unique property being the polarization angle? This is not the case: One condition of the Unique Object definition is that the adversary knows the unique properties of a Unique Object. But once the polarization of the photon is known, many photons with the same polarization state can be generated. Unique Objects thus relate on a different type of unclonability than quantum systems.

## 3 Weak Physical Unclonable Functions (Weak PUFs)

One class of Physical Unclonable Functions based on inherent device variations are Weak PUFs. They exploit the disordered, unique, internal structure of the underlying fabric as a non-volatile memory for storing the secret keys. In an ideal case, the volatile keys generated by Weak PUFs upon power-up cannot be determined by external and invasive attacks due to construction or tamper-proof properties of the pertinent structure. Weak PUFs are also known under the name of Physically Obfuscated Keys (POKs) [2].

The term *Weak PUF* was coined in [57] to refer to PUFs with a limited number of challenge-response pairs (CRPs) in contrast to Strong PUFs that contain many CRPs. The following specification has it in greater detail.

1. *Challenge-Response Pairs.* A Weak PUF can be interrogated by one (or a very small number of) fixed challenge(s) $C_i$, upon which it generates response(s) $R_{C_i}$ that depends on its internal physical disorder.
2. *Key Derivation.* The response(s) $R_{C_i}$ from a Weak PUF is (are) exploited by the device for deriving a standard digital key that can be used for security applications.
3. *Practicality and operability.* The generated response $R_{C_i}$ should be sufficiently stable and robust to environmental conditions and multiple readings.

**Weak PUFs vs. UNOs.** It is important and necessary to differentiate Weak PUFs from Unique Objects: Applications of Unique Objects require an adversarial model where Eve has time to inspect all features of the Unique Object, and will often know its internal structure and unique fingerprint. Furthermore, these unique properties are measured by an external apparatus. The exact opposite holds for Weak PUFs: Their responses are measured internally, and the derived key is kept secret in the embedding hardware.

### 3.1 History and Implementation Examples

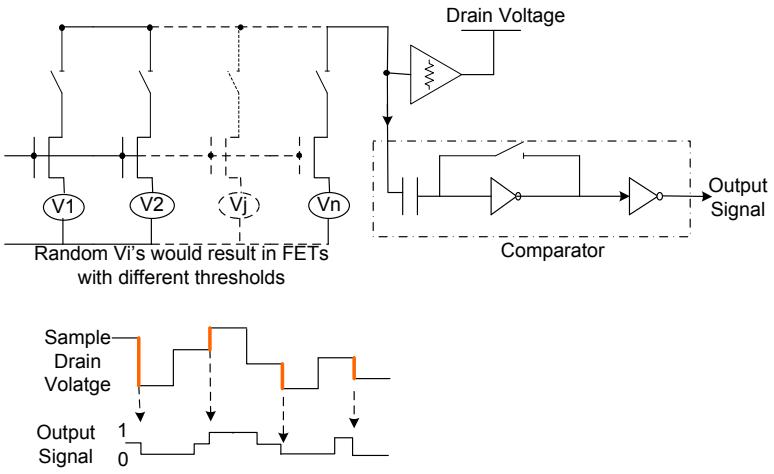


**Fig. 6.** Array of ICID transistors producing a sequential random voltage proposed in [58].

**ICID PUFs.** ICID is the first proposed and designed circuit structure for generating a Weak PUF (or *random chip ID*) based on process variations [58]. They devised an array of addressable MOSFETs (shown in Figure 6), with common gate and source and sequentially selected drains driving a resistive load. Because of device threshold voltage mismatches (resulting from process variation) the drain currents are randomly different. Therefore, at each die, a unique sequence of random voltages would be generated at the load. ICID exploits these unique sequences of random but repeatable voltages

to construct unique identification. In $0.35\mu m$ technology, the authors reported about 10% false positive and false negative results for repeating random bits on their test circuits. Identification capability can be improved by increasing the bit length.

**Physically Obfuscated Keys (POKs).** Under the name of a Physically Obfuscated Key (POK), Gassend proposed a type of Weak PUF that was built from the first integrated Strong PUF (see Figure 8 in Section 3.2 for the architecture, and see Section 4 for Strong PUF) [2]. The POK/Weak PUF would only utilize one (or a small subset) of all possible challenges for a Strong PUF. This allows using them exactly as a digital key that is more resistant to physical attack, because it extracts its information from a complex physical system.

**SRAM-based PUFs.** A commonly used candidate structure for a Weak PUF exploits the positive feedback loop in an SRAM or an SRAM-like structure. If a write operation is used, the cross-coupled device starts transitioning to the inserted value, and the transition is sped up by the positive feedback loop in the structure. When no write operation is in place and the system is not in any of the states (initial power up), the inherent small transistor threshold mismatches and thermal and shot noise trigger the positive feedback loop so the state would be in one of its two possible stable points (0 or 1). The effects of common mode process variations including lithography, and common mode noise (e.g., substrate temperature and supply fluctuations) is similar on the differential device structure and does not strongly impact the transition.

The idea of fingerprinting of semiconductor integrated circuits using SRAM was originally suggested in a 2002 patent, but no experimental implementation data were included [59]. The work in [60] constructed a custom-built array of SRAM-like cells that generated random values based on threshold mismatches in $0.13\mu m$ technology (Figure 7). Their experiments have shown a close to uniform distribution of the bits (close to Normal distribution of Hamming distances) and more than 95% bit stability. The work in [61] showed that initialization of SRAM can produce a physical fingerprint for

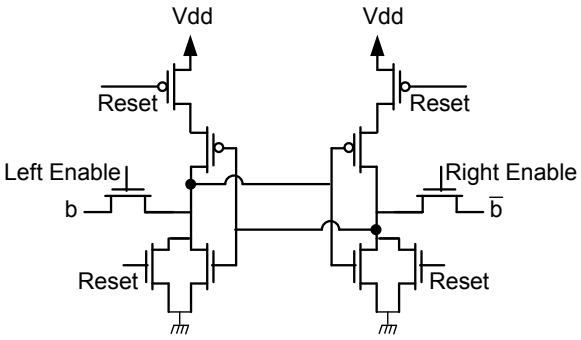


**Fig. 7.** The positive feedback loop created by the cross-coupled NOR (NAND) gates is used for storing a 0 or a 1 in SRAM-like memory structures.

each chip. They have also shown that the fingerprints can pass the standard NIST randomness tests for runs [61]. The authors in [57] also exploit the initial state of the SRAMs in an FPGA to extract IDs based on differential device mismatches. They coined the term *intrinsic PUF* to refer to structures that do not need additional circuitry for embedding the PUF.

Since not all FPGAs have SRAMs built in them, the work in [5] proposed *butterfly PUFs* based on reconfiguring the FPGA cells to construct two back-to-back NAND gates (in positive feedback mode) similar to the SRAM structure. Note that the Butterfly PUF cannot be considered intrinsic, since it should be custom configured the same way as any other logic circuitry can be made on a reconfigurable fabric.

**Coating PUFs.** Another construction of a Weak PUF is a *coating PUF* that provides a practical implementation of read-proof hardware. A read-proof hardware device has the property that once constructed, no outside entity can read (extract) information on the data stored in the device. The authors in [62] introduced coating PUFs as form of a protective coating that can be sprayed on the IC and cover its surface. The coating is composed of a matrix material doped with random dielectric particles (i.e., different kinds of particles of random shape, size and location with a relative dielectric constant differing from the coating matrix's dielectric constant). The

top metal layer of the IC contains an array of sensors that are used to measure the local capacitance values of the coating.

One central property of a Coating PUF is their purported tamper sensitivity: It is assumed that any tampering with the coating (such as invasive penetration, or removal from the covered IC) strongly and irrecoverably changes its properties. [62] has positively evaluated the resilience of coating PUFs against some optical and invasive attacks.

**Resistive PUFs.** Another instance of silicon-based PUFs are based on power distribution and resistance variation of chips that have appeared in recent literature [63, 64].

### 3.2 Protocols, Applications, and Security

**Secret key generation and storage.** Weak PUFs provide a method for secret key generation and storage based on random disordered physical medium fluctuations. Therefore, any security protocol that leverages the storage of a secret key can utilize a Weak PUF in its flow. To our knowledge, the earliest security protocols and IP protection applications based on Weak PUFs/POKs were presented in [2, 65]. Other protocols and applications including metering and RFID protection based on Weak PUFs were presented in [62, 66, 57, 5, 67].

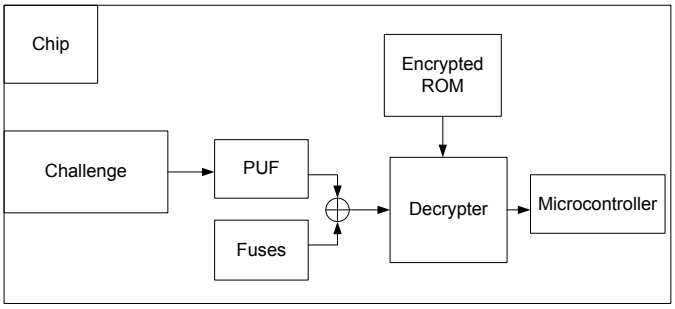


**Fig. 8.** A POK built by using a Strong PUF proposed in [2].

**IP protection application.** Weak PUFs were proposed for protecting hardware IP and ICs against piracy. A proposed system for

protecting programmable hardware IP against piracy is shown in Figure 8 taken from [2]. Assume that the design is a microcontroller with a compression algorithm stored in ROM. A Strong PUF is hardwired with other functions on the chip to generate a $k$-bit key $K$ that is the same for all chips to mitigate the cost. The challenges to the PUF are also hardwired to be fixed. That PUF response is combined with the contents of burned on-chip fuses through an exclusive-or operation to produce $K$. A decoder uses $K$ to decrypt the ROM content. By selecting the fuse bits one can generate the same decrypting key $K$ on all chips. The response never leaves the chip during the decryption operation. Even if the state of all the fuses are discovered, the key would remain secret.

**Secure processor.** Suh [65] describes how a Weak PUF can be embedded in a secure processor which then can be used for applications such as certified execution and software licensing. In one design, the weak PUF is used to generate a seed for a public/private key pair. The seed and private key are never exposed and the public key is published and certified by a certification authority. In another, the seed is used as a symmetric key to encrypt a secondary symmetric key that is known to the user of the processor. Again, the seed remains unknown, and is only used to encrypt a given secondary key and decrypt the secondary key for internal use in secure execution.

**Active IC metering.** Another usage of Weak PUFs was for active IC metering that protects the hardware against foundry piracy (overbuilding) [66]. Here, the functional specification of the design in the finite state machine (FSM) domain was modified. The alteration was such that an exponential number of states were added to the design with a low overhead. Hiding a state in the large state-space of the FSM was later shown to be an instance of a provably obfuscatable general output multi-point function. It was shown that the transitions from the hidden state cannot be found by having access to the layout and even access to the register's contents that store the state. Upon fabrication at the foundry, based on the Weak PUF's response, the design would be in one of the hidden obfuscatable states that is called a locked state. This locked state can be read out by everybody, but the passkeys to the functional (unlocked) state can only

be provided by the original designer who has access to the modified FSM.

**Security analysis.** Weak PUFs are commonly attributed three advantages:

(1) They are harder to read-out than standard digital keys that are stored permanently in non-volatile memory (NVM) since keys only exist when the chip is powered.
(2) They can possess some natural tamper sensitivity, meaning that any tampering with the device, or even with the hardware system that embeds the PUF, alters the physical features of the device and the key derived from them.
(3) They save on costs, since they avoid the process steps necessary to include NVM in hardware systems.

Some of these assets must be analyzed further. Let us start with advantage (1): Weak PUFs clearly avoid the *long-term* presence of *digital* keys in NVM. But the security of a Weak PUF based hardware still depends on the secrecy of a single digital key derived from the Weak PUF, which is present for at least a short period after its derivation from the PUF's responses. This creates a single digital point of failure for the system. Weak PUFs furthermore cannot alleviate the permanent presence of secret information in the hardware in general: If an adversary knows the disorder or fabrication mismatches that determine the responses of the Weak PUF, he may simulate and derive these responses.

Further, Weak PUF based hardware may suffer from similar weak spots as other systems built on standard binary keys. Side channel or emanation analysis may be possible; and since the device will apply some standard cryptoprimitives to $K$, its security will thus depend on the same unproven computational assumptions as any classical system built on digital keys.

Regarding the above asset (3), it must be stressed that error correcting information is vital for Weak PUFs; any single bit flips make the system not applicable any more. This necessitates the use of accompanying error correcting information, which must be stored in NVM of some form. To the asset of Weak PUFs, this storage can

be external and/or public; further, it need not be implemented in the hardware that contains the Weak PUF.

## 4 Strong Physical Unclonable Functions (Strong PUFs)

Immediately after the introduction of Weak PUFs or POKs, a second class of PUFs was put forward [68, 69, 1, 70]. They have later often been referred to as Strong PUFs, for example in [57, 71, 72].

In a nutshell, a Strong PUF is a disordered physical system with a very complex input-output behavior that depends on its disorder. The system must allow very many possible inputs or challenges, and must react with outputs or responses that are a function of the applied challenge and of the specific disorder present in the system. The input/output behavior should be so complex that it cannot be imitated numerically or by any other device.

More specifically, a Strong PUF is a disordered physical system $S$ with the following features:

1. *Challenge-Response Pairs.* The Strong PUF can be interrogated by challenges $C_i$, upon which it generates a response $R_{C_i}$ that depends on its internal physical disorder and the incident challenge. The number of CRPs must be very large; often (but not always) it is exponential with respect to some system parameter, for example with respect to the number of components used for building the PUF.
2. *Practicality and operability.* The CRPs should be sufficiently stable and robust to environmental conditions and multiple readings.
3. *Access mode.* Any entity that has access to the Strong PUF can apply multiple challenges to it and can read out the corresponding responses. There is no protected, controlled or restricted access to the PUF's challenges and responses.
4. *Security.* Without physically possessing a Strong PUF, neither an adversary nor the PUF's manufacturer can correctly predict the response to a randomly chosen challenge with a high probability. This shall hold even if both parties had access to the Strong PUF at an earlier time for a significant period, and could make any

reasonable physical measurements on the PUF, including (but not limited to) determination of many CRPs.

The definition above is more qualitative than quantitative in order to remain intuitive; a more formal and thorough definition can be found in [72].

**Unique vs. Weak vs. Strong.** While a Unique Object must always possess an external and a Weak PUF always an internal measurement equipment, this is left open for Strong PUFs; both variants are possible and have been realized in practice (see [68, 69] for an optical PUF with an external and [1, 73] for an electrical PUF with an internal measurement apparatus). Unique Objects require a trusted measurement apparatus whereas Strong PUFs once "bootstrapped" (cf. Section 4.2) can be remotely authenticated with an untrusted measurement apparatus. Another difference between Strong PUFs and Unique Objects lies in the exact adversarial model and the relevant security properties: While the adversary's aim in the case of Unique Objects lies in physically fabricating a clone device with the same properties, his goal in the case of Strong PUFs is to learn how to predict the input/output behavior of the Strong PUF. The latter is a mixture of numerical assumptions and physical hypotheses. This fact does not exclude that the same structure can be used as a Unique Object and as a Strong PUF under different read-out schemes, for example; but not every Unique Object is a Strong PUF and vice versa.

Weak PUFs possess only a small number of fixed challenges, whereas Strong PUFs have a very large number of challenges. In Weak PUFs, the responses remain secret and internal. To the contrary, Strong PUFs allow free querying of their responses.

### 4.1 History and Examples of Strong PUFs

**Optical PUF.** The first implementation of a Strong PUF has been suggested in [68] [69], albeit under the different name of a physical one-way function. It consists of a transparent plastic token which contains a large number of randomly distributed glass spheres as shown in Figure 9. We call this implementation an *optical PUF*. An individual, unclonable token is illuminated under different angles

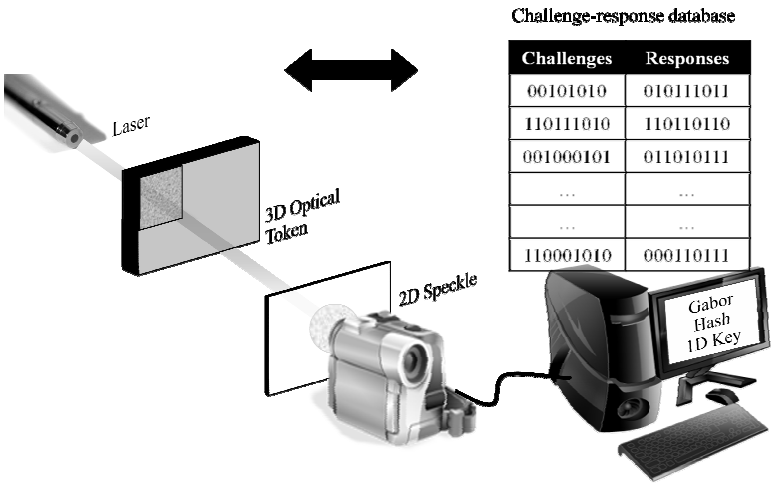


**Fig. 9.** A 3D inhomogeneous transparent plastic token being optically challenged (illuminated under different angles and points of incidents) and produces an output in form of an interference pattern. The output is hashed to produce a 2D image, which is in turn filtered by a multiscale Gabor transform to produce a 1D key as proposed in [69].

and points of incidence (which are regarded as the challenges of the system), and produces an interference pattern, which is considered the response of the system. We draw the reader's attention to the similarity in Figures 3 and 9. The main difference is a usage one: optical PUFs are assumed to have a large number of challenges, and a secret set of challenge-response pairs is stored in a central database. Thus, optical PUFs can be remotely authenticated.

This construction is presumably secure (no attacks are known to this date), but the measurement apparatus is external and relatively large, potentially leading to practicality issues and stability problems when the token is measured by different apparatuses at different locations.

**Arbiter PUF.** Almost simultaneously to optical PUFs, the first integrated electrical Strong PUFs including "Arbiter PUFs" were put forward in [1] [73]. [1] is also the first publication that uses the term PUF as a common abbreviation for the expressions Physical Random Function and Physical Unclonable Function. Unlike optical

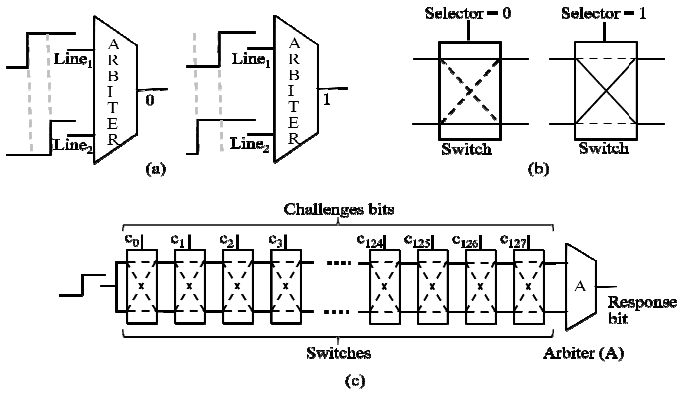


**Fig. 10.** (a) Demonstration of an arbiter's operation: the relative time of signal arrival at $\text{Line}_1$ and $\text{Line}_2$ would determine the value of the output bit; (b) Demonstration of a selector's operation: the selector bit would decide if the top and bottom lines continue in the same order, or they switch places; (c) An arbiter PUF with 128 challenge bits $c_0, \ldots, c_{127}$ applied as the selectors to the switches. The switch selectors dynamically configure two parallel paths with random delay differences that would form the response generated by the arbiter [74]).

PUFs, silicon PUFs do not require external measurement equipment. They are based on the runtime delay variations in electrical circuits.

In one implementation, an electrical signal is split into two parallel signals, which race against each other through a sequence of $k$ electrical components, for example, $k$ multiplexers. This architecture is shown in Figure 10. As shown in the figure, the challenges are applied to the selectors of the multiplexers. The exact signal paths are determined by these challenge bits $b_1, \ldots, b_k$ applied at the multiplexers. At the end of the $k$ components, an arbiter element decides which of the two signals arrived first and correspondingly outputs a zero or a one, which is regarded as the system's response.

It was clear from the beginning that these first electrical candidates were prone to modeling attacks as mentioned in [1]. Attacks using machine learning algorithms have been carried out, see Section 4.2. In these attacks, the adversary collects many challenge-response pairs (CRPs), and uses them to derive the runtime delays occurring in the subcomponents of the electrical circuit. Once they are known, simple simulation and prediction of the PUF becomes possible, breaking its security. One reason why these attacks worked so well lies in the fact that plain Arbiter PUFs have relatively simple

linear models, in which the delay of each of the two signals can be approximated as the linear sum of the delays in the subcomponents. This makes standard machine learning algorithms applicable to the problem.

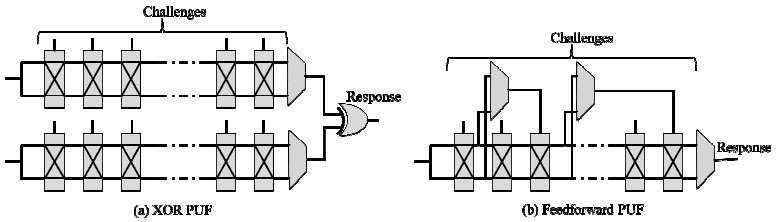


**Fig. 11.** (a) An arbiter PUF wtih added XORing of two arbiter outputs; (b) Feedforward PUF.

**Variants of the Arbiter PUF.** The above issues naturally led to the introduction of non-linear electrical PUFs, for example, XOR arbiter PUFs, Lightweight Secure PUFs and Feedforward Arbiter PUFs [11, 75, 76, 74]. In an XOR arbiter PUF, multiple arbiter outputs are XOR'ed to form a response. In Figure 11(a), an example is shown where two arbiter outputs are XOR'ed. In the Feedforward Arbiter PUF, the output of intermediate multiplexer(s) on the signal paths are input to so called Feedforward arbiter(s). The Feedforward arbiter output is then fed to the input of another multiplexer forward on the signal path. In Figure 11(b), an example of a Feedforward arbiter structure is shown. All of the aforementioned structures employ the basic Arbiter PUF architecture, but refine its architecture by introducing additional, non-linearities. These structures showed a significantly higher resilience against machine learning attacks, but still could be attacked up to a certain level of size and complexity [77, 54].

Arbiter PUFs and their variants have been shown to have small and stable integrated electrical implementations and have been commercialized [78].

**Legacy PUFs.** [80] has proposed using the ICs' timing path signatures that are unique for each state-of-the-art CMOS chip (because

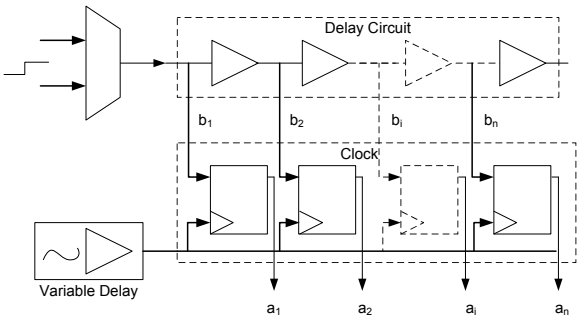


**Fig. 12.** The glitch PUF architecture samples the glitches on the path and the arrival of a glitch compared to a clock signal generates the response bits in the FFs [79].

of process variations) as a PUF. The work in [81, 82] has shown that all ICs that are fabricated in new CMOS process nodes that contain nontrivial process variations have a unique signature that can be extracted using noninvasive methods by the structural side channel tests such as IDDT, IDDQ, or delay tests. They have shown a unified gate-level characterization of the signatures for all side-channels that could be used as a compact representation. It was shown that statistical signal processing methods can be adopted for ensuring rapid and robust characterization [83–87]. The interesting aspect of this line of work is that the signatures are intrinsic to all legacy ICs, and there is no need for insertion of additional circuits or structures by the manufacturer or other parties who are interested in verifying the chip's authenticity by its specific signature. Therefore, it can be readily used for digital rights management of integrated circuits in the supply chain and for anti-counterfeiting protection.

**Analog PUF family.** New, recent suggestions for Strong PUFs have tried to exploit the analog characteristics of electrical signals, such as in analog cellular arrays [88]. The system suggested in [88] imitates optical wave propagation in an electrical cellular non-linear network, transferring the known complexity of optical PUFs into electrical circuits. Another non-linear electrical suggestion is [79] that is based on the nonlinear propagation of glitches on a logic path. Figure 12 demonstrates the architecture of the glitch PUF system, where the glitches based on the delay difference between the signal path and the clock signal are stored in the response FFs.

Finally, integrated optical PUFs have been proposed [89], but their security seems suspect if merely linear scattering media are used (see appendix of [90]).

### 4.2 Protocols, Applications, and Security

**Protocols and applications.** The archetypical application of Strong PUFs is the identification and authentication of hardware systems (or other security tokens such as credit cards) [69] [68] [1]. The corresponding protocols are usually run between a central authority (CA) and a hardware/token carrying a Strong PUF $S$. One assumes that the CA had earlier access to $S$, and could establish a large, secret list of challenge-response-pairs (CRPs) of $S$ using a trusted external measurement apparatus in the case, for example, of an optical PUF. This step is usually referred to as bootstrapping. Whenever the hardware, possibly at a remote location, wants to identify itself to the CA at some later point in time, the CA selects some CRPs at random from this list, and sends the challenges contained in these CRPs to the hardware in the clear. The hardware applies these challenges to $S$, and sends the obtained responses to the CA, also in the clear. If these responses closely match the pre-recorded responses in the CRP-list, the CA believes the identity of the hardware. Note that each CRP can only be used once, whence the CRP-list shrinks over time, and needs to be large. As noted above, an exact match is not required, and a certain level of noise in the responses can be tolerated.

Another application that has been mentioned is key exchange or key establishment based on Strong PUFs [69]; a formal protocol has been given in [89]. However, it has been shown in [91] that such key exchange protocols can suffer from problems regarding their forward secrecy and their repeated use for session key exchange. [91] also proposed a new type of PUF that can fix this issue, called erasable PUFs. We note that this new type of erasable PUFs is different than the earlier FPGA PUFs that could be configured and erased for each authentication session [77, 92].

The above protocols give Strong PUFs broad cryptographic applicability. They can be employed for any application which requires

the above cryptographic tasks, often without storing explicit digital keys in the hardware containing the PUF.

**Security features and Attacks.** Attacks on Strong PUFs will either try to build a physical clone, i.e., a second physical system that behaves indistinguishably from the original PUF, or a digital clone, i.e., a computer algorithm that imitates the PUF's challenge-response behavior.

It has been rightfully stressed in early publications on Strong PUFs [68, 69] that they can avoid the classical, well-known number-theoretic assumptions in cryptographic protocols. But is the security of Strong PUFs entirely free of computational assumptions, and can it merely be built on their internal entropy and randomness? It is known that the maximal amount of randomness or entropy in a physical system is polynomially bounded in the size of the system [93, 71, 72]. This implies that the overall number of possible challenges of many PUFs is larger than their entropy. In particular, this observation necessarily holds for any PUFs with an exponential number of challenges.

An adversary therefore often merely needs to gather a small subset of all CRPs of a Strong PUF to obtain (at least indirect) knowledge about all CRP-relevant information/entropy contained in the PUF. Once he has gathered such a subset, it is merely a computational assumption that he cannot derive an internal PUF model from it which allows PUF prediction. For example, he could set up a system of (in-)equations from the CRP subset, whose variables describe the inner PUF structure. If he can solve this system of (in-)equations efficiently, he can break the PUF. The hypothesis that he will not be able to do so is just another type of unproven computational assumption.

This perhaps surprising observation is not just a theoretical concern. The modeling attacks presented in [2, 75, 94–96, 71, 77, 54] are practical, and prove the basic feasibility and effectiveness of such attacks. They also exhibit that such attacks reach their limits when the involved computations become too complex; for example, the authors of [54] could not attack XOR arbiter PUFs with $k > 6$ XORs because the complexity grew exponentially in $k$.

In other words, the security of many Strong PUFs is dependent on the underlying computational assumptions. In favor of Strong PUFs, it must be said that these assumptions are independent of the classical number-theoretic assumptions such as the factoring or discrete logarithm function, and that Strong PUFs can help to establish an independent basis for cryptography and security. Furthermore, they have other security advantages, as discussed in the remainder of this section. The only subtype of Strong PUFs whose security is strictly independent of computational assumptions are SHIC PUFs [10, 97, 98]. The price they pay for this feature is an intrinsically slow readout speed and a comparably large area consumption.

This brings us to another central point related to Strong PUF security. Strong PUFs avoid the use of explicit digital keys in hardware. But do they avoid the presence of secret information in hardware in general? Once the internal configuration of a Strong PUF has become known, an adversary will almost always be able to predict and hence break the PUF. To illustrate our point, consider the Arbiter PUF and its variants: Once the internal runtime delays have become known, the structure can be fully predicted and broken. Therefore Strong PUFs, just like classical cryptosystems, often depend on the assumption that some internal information remains secret. In their favor, this information is arguably hidden better than if stored explicitly in the form of a digital key. $F$ will usually be known to the adversary and efficiently computable.

**Security benchmarks.** Natural security benchmarks for Strong PUFs must evaluate the complexity of their challenge-response behavior and their resilience against modeling attacks. To this end, various measures have been proposed: (i) Theoretical analysis of the overall internal entropy of the PUF [69]. (ii) Theoretical analysis of the entropy / information-theoretic independence of the CRPs [99–101]. (iii) Empirical, statistical analysis of large CRP sets by statistical tools and compression algorithms [95, 102, 103]. (iv) Empirical analysis by assessment of machine learning curves over instances of increasing size and complexity [95, 103].

Let us briefly discuss these approaches. One downside of (i) is that it usually does not consider the CRP-relevant entropy, but the general entropy of the system, which is often very much larger. (ii)

is a suitable measure. On the downside, it can be difficult to derive theoretically, and does not take into account computational aspects. (iii) and (iv) are easy to apply and generic tools, but do not provide definite security guarantees. (iii) does not require an generic model of the PUF (such as the linear additive delay model for arbiter PUFs), while method (iv) needs such a model before it can be applied.

# 5 Controlled Physical Unclonable Functions (CPUFs)

## 5.1 Specification of Controlled PUFs

Let us start by specifying the notion of a Controlled PUF: A Controlled Physical Unclonable Function (CPUF) is a PUF that has been bound with an algorithm in such a way that it can only be accessed through a specific Application Programming Interface (API).

The main problem with (uncontrolled) Strong PUFs is that anybody can query the PUF for the response to any challenge. To engage in cryptography with a PUF device, a user who knows a CRP has to use the fact that only he and the device know the response to the user's challenge. But to exploit that fact, the user has to tell the device his challenge so that it can get the response. The challenge has to be told in the clear because there is no key yet. Thus a man in the middle can hear the challenge, get the response from the PUF device and use it to spoof the PUF device.

Clearly the problem in this attack is that the adversary can freely query the PUF to get the response to the user's challenge. By using a CPUF in which access to the PUF is restricted by a control algorithm, this attack can be prevented. The API through which the PUF is accessed should prevent the man-in-the-middle attack we have described without imposing unnecessary limitations on applications.

## 5.2 History and Implementation

CPUFs can perform all operations that a Strong PUF can perform. While the details of various CPUF APIs are beyond the scope of this paper, useful APIs have been developed [70, 104] that satisfy the following properties:

1. *Access Control.* Anybody who knows a CRP that nobody else knows, can interact with the CPUF device to obtain an arbitrary number of other CRPs that nobody else knows. Thus users are not limited to using a small number of digital outputs from the PUF. Moreover, if one of these new CRPs was revealed to an adversary, transactions that use the other CRPs are not compromised. This is analogous to key management schemes that use session keys derived from a master key.
2. *Secret Sharing.* Anybody can use a CRP that only they know to establish a shared secret with the PUF device. Having a shared secret with the PUF device enables a wide variety of standard cryptographic primitives to be used.
3. *Control Algorithm.* The control algorithm is deterministic. Since hardware random number generators are sensitive and prone to attack, being able to avoid them is advantageous.
4. *Cryptographic Primitive.* The only cryptographic primitive that needs to be built into the control algorithm is a collision resistant hash function. All other cryptographic primitives can be updated during the lifetime of the CPUF device.

By selecting an appropriate API, a CPUF device can be resistant to protocol attacks. With careful design, Optical and Silicon PUFs can be made in such a way that the chip containing the control logic is physically embedded within the PUF: the chip can be embedded within the bubble-containing medium of an Optical PUF, or the delay wires of a Silicon PUF can form a cage on the top chip layer. This embedding should make probing of the control logic considerably more difficult, as an invasive attacker will have to access the wires to be probed without changing the response of the surrounding PUF medium.

The PUF and its control logic have complementary roles. The PUF protects the control logic from invasive attacks, while the control logic protects the PUF from protocol attacks. This synergy makes a CPUF far more secure than either the PUF or the control logic taken independently. Figure 13 demonstrates an example architecture of how a controlled PUF can be used for improving a PUF. A random hash function is placed before the PUF to prevent the adversary from doing a PUF chosen challenge attack. So a model-

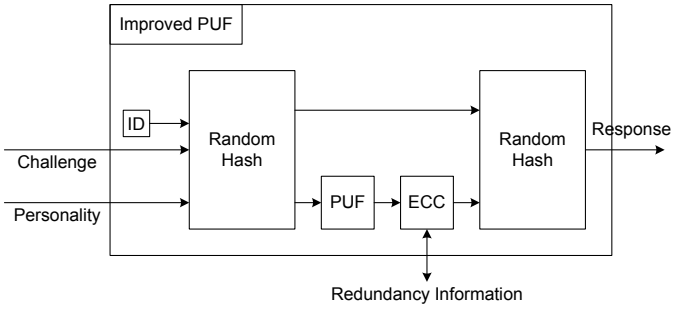


**Fig. 13.** An example architecture for a controlled PUF proposed in [70].

building adversary is prevented from selecting challenges that allow him to extract the PUF parameters. To ensure response consistency, an Error Correcting Code (ECC) is used. An output random hash function is used to decorrelate the response from the actual physical measurements, and therefore rendering a model-building adversary's task even harder.

### 5.3 Protocols, Applications, and Security

Because there is no algorithmic way to tie together all the keys produced by a device, the device will have to take an active part in protocols like certificate verification, that would not usually need any device involvement. This limitation is offset by a decreased vulnerability to invasive attacks.

There are many applications for which CPUFs can be used, and we give two examples here. Other applications can be imagined by studying the literature on secure coprocessors, in particular [105]. We note that the general applications for which this technology can be used include all the applications today in which there is a single symmetric key on a chip.

A bank could use certified execution to authenticate messages from PUF smartcards. This guarantees that the message the bank receives originated from the smartcard. It does not, however authenticate the bearer of the smartcard. Some other means such as a PIN number or biometrics must be used by the smartcard to determine if its bearer is allowed to use it. If the privacy of the smartcard's message is a requirement, then the message can also be encrypted.

A second application is for computers that implement private storage [106–112]. A program wishing to store encrypted data in untrusted memory uses an encryption key which depends uniquely on the PUF and its program hash. This requires a CPUF in order to accomplish the unique dependency. This idea is implemented in the AEGIS processor [112, 113].

Physically obfuscated keys generated from Weak PUFs seem to increase the difficulty of an invasive attack, but they still have a single digital point of failure. When the device is in use, the single physically obfuscated master key is present on it in digital form. If an adversary can get that key he has totally broken the device's security. CPUFs exploit the parameterizability of the complex physical system like Strong PUFs do. For each input to the physical system, a different key is produced. Thus the complexity of the physical system is exploited to the utmost.

As noted previously one difficulty with Weak PUFs is that their output is noisy. For use in cryptography, we need error-correction which does not compromise the security is required. For Weak PUFs only one response has to be made noise-free, for CPUFs many responses have to potentially be corrected. We need to store an error correcting syndrome with each challenge-response pair. Secure and robust error correction has been considered for Weak PUFs (see [7]) but these schemes need to be efficiently generalized to CPUFs.

## 6 Emerging PUF Concepts

There are a number of new concepts that have emerged in the area of PUFs, and the pace of innovation is rapid. We mention interesting new concepts proposed in the past couple of years in this section, and address ongoing research challenges in Section 7.

### 6.1 PUFs with Secret Models

In classical identification schemes based on Strong PUFs, the verifier must possess a large list of CRPs that have been pre-measured in a secure bootstrapping phase [68, 1]. The challenges sent to the prover are chosen randomly from this list, and the responses obtained from the prover are verified for correctness against this list. Since the list

must suffice for the lifetime of the device, it must be large, which imposes uncomfortable storage requirements on the verifier.

It has been independently observed by [114, 115, 77] that such storage requirements may be lifted if the verifier instead stores a secret model for the PUF, by which he can simulate and predict arbitrary responses of the PUF. Such secret models can furthermore allow the offline verification of a PUF's identity, i.e., they can enable identification protocols that are run without an online connection to a trusted authority holding a CRP-list. The underlying PUF primitive could be called *Secret Model PUF* or *SM PUF*, for short.

Secret Model PUFs are a very useful concept that leads to improved practicality features and new protocols. They do not lift two important constraints of Strong PUFs, though: First, the model itself must be kept secret, similar to a secret key. They therefore require the authenticating entity to store a symmetric key to decrypt the secret model stored in encrypted form on the PUF device. Second, SM PUFs still contain some secret information, namely the information that was used to set up the secret model (for example the internal runtime delays). These two requirements are only overcome by the concepts proposed in the next subsections 6.2 and 6.3.

### 6.2 Timed Authentication

For certain implementations of Strong PUFs, the real-time interval in which the PUF generates its responses may be noticeably shorter than the time that any numerical model or purported clone would require to the same end.

In a PUF-related context, this observation has first been stated in [77]. They noted that for certain FPGA-based PUFs, only the authentic hardware would be able to generate the response in a minimum number of cycles, and that a model built based on the device characteristics would likely be slower in finding the response to a given challenge (compared to the original device). They proposed an authentication protocol that exploits this unique property of the original FPGA device: A time-bound set by the protocol for obtaining the correct response after applying a random challenge ensured that only the authentic device could respond. This scheme has been referred to as Timed Authentication (TA) in [77].

[77] suggests an "asymmetry" in the timed computational capabilities of the authentic device compared to other entities. This asymmetry was elaborated on for thwarting the modeling attacks. However, the proposed protocol is a symmetric key like scheme, since it requires a secret list of CRPs. It was noted that asymmetry can lift the feature that the internal configuration of the PUF-hardware must remain secret. Think of the optical PUF introduced in Section 4.1 as an example: Even if the position of all internal scattering elements/bubbles was known to an adversary, he would still find it hard to simulate the complex input-output behavior of the scattering medium in real-time. The same holds for the FPGA-based implementation of TA discussed in [77].

### 6.3 PUFs with Public Models

Section 6.1 told us that a secret model for a Strong PUF can replace the CRP list. Section 6.2 described that certain Strong PUFs operate faster than any adversarial model and emulation. Both concepts can be combined to enable PUFs with simulation models that can be made public (and hence can be simulated by everyone), but which still operate faster than any clone or model (including the public model, of course). The manufacturer or some other entity can tie the model to the respective PUF, by, for example, signing the model, or keeping it in a trusted public register. This allows everyone to simulate the responses of the PUF with some time overhead. Only the party holding the PUF can determine the PUF's responses fast, i.e., within a certain time bound, by a physical measurement on the PUF. This allows public key like functionalities and protocols. Hardware systems based on such a concept have the further intriguing advantage that they can eradicate the presence of any form of secret information in the cryptographic hardware, while still being usable in typical digital network applications such as remote identification and message authentication.

**History.** The concept of PUFs with public models has been introduced and developed independently in several lines of research. Under the name of a Public PUF (PPUF), this primitive has been introduced in [116–118], building on a hardware concept that had

been published earlier [119, 120, 81, 77]. Protocols and applications of PPUFs have since been developed [117, 118, 121]. Under the name of a SIMPL system, the same concept was put forward completely independently in [122, 123]. Implementations, protocols and applications of SIMPLs have been elaborated on in [124–127, 90, 128]. In another line of research, the concept of PUFs with public models has been made explicit with implementation results on FPGAs under the name Time-Bounded Authentication (TBA) in [92, 9]; this builds on the concept of TA treated in the last section [77].

### 6.4 Quantum Readout PUFs

[129] proposed modifying the challenge-response mechanism of a PUF with quantum states, called a *Quantum Readout PUF* [130]. The properties of the quantum states prevent an adversary from intercepting the challenges and responses without modifying them. Thus, there is no need for a trusted location for bootstrapping. However, no proof-of-concept implementation or practical architecture for this structure has been proposed to date. Finally, interfacing the quantum readout device to the regular PUF is likely a challenge.

### 6.5 SHIC PUFs

A final recent concept are PUFs with Super-High Information Content, abbreviated *SHIC PUFs* [1] [10, 97, 98]. SHIC PUFs are Strong PUFs whose large number of CRPs are pairwise independent in an information-theoretic sense. Unlike other Strong PUFs, this allows them to become independent of computational assumptions in their security. The price they pay is a relatively large area consumption and slow read-out speed on the order of $10^2$ to $10^4$ bits per second. SHIC PUFs are unlikely to be used in low-cost commercial applications in the near future, since there are other, more favorable solutions to this end. But they represent an intriguing theoretical tool, since they are a variant of Strong PUFs with information-theoretic security. Furthermore, investigating their optimal implementation is rewarding from a technological perspective, since it relates to fundamental technological questions such as "How much random information can we store and reliably extract from a solid-state system?",

[1] SHIC PUFs are to be pronounced as *"chique PUFs"* according to [10].

and "How can we make the speed in which information is released from a solid-state system inherently *slow*?".

# 7 Future Research Topics

## 7.1 Open Public PUF Questions

The main open questions related to PUFs with Public Models concern their hardware realization:

- How can it be guaranteed that the model requires more time to simulate than the PUF device requires to return a response?
- How can it be guaranteed that a well-equipped adversary for sure takes longer than the PUF device, while any poorly equipped honest party can simulate the response in feasible time in the course of a communication protocol?
- Can the model be close enough to the PUF so that an adversary finds it difficult to physically clone the PUF, but loose enough to allow for variation due to environmental conditions of PUF responses?

While there have been many recent proposals for timed authentication, we are not aware of any implementation that definitively settles the above questions. This leaves strong potential for future research. If a workable, small and inexpensive implementation of PPUFs, SIMPL systems or TBA systems is found eventually, or if one of the existing implementations is shown to possess all necessary properties, this would have a massive impact on the way we perform cryptography and construct security hardware.

## 7.2 Efficient Hardware Implementations: Overhead versus Security

Recent work has discussed how it could be possible to safeguard PUFs against reverse-engineering and modeling attacks [77, 54, 88, 10, 97, 98]. However, most methods that aim at protecting against such attacks add strongly to the power, size, delay, instability, or cost overhead of the system. Also techniques for ensuring the tamper-proof properties, such as inaccessibility of the Weak PUF, would require addition of tamper-proof circuitry and material to the devices.

One major future research topic is how the security of Strong PUFs and/or the tamper sensitivity of Strong PUFs and Weak PUFs can be realized with a minimal hardware overhead. These future research questions naturally relate to circuit design and, concerning tamper sensitivity, also to the material sciences.

### 7.3 Error Correction and Practical Operability

A suite of security applications of PUFs, such as secret key generation by Weak PUFs, require full and error-free reconstruction of the keys. However, environmental conditions and aging may affect the measured responses in strong ways. Methods for compensation of such effects, such as circuit reliability enhancement techniques, error correction and secure sketches, are being developed [2, 131, 8, 4, 5, 7]. Further development of methods that ensure robustness of PUFs with a limited amount of leaked information is of great interest. One key challenge is that the maximum number of unpredictable bits should be known at the design time. If the unpredictability exceeds the bound set at the time of design, the error correction method would not be able to compensate for the errors. Therefore, careful experimental studies for each new PUF structure are needed for characterizing the performance under different temperature, voltage, and/or other environmental and operational conditions, constituting a future area of active and fruitful interplay between hardware analysis and error correction techniques.

### 7.4 IC Metering and Counterfeit Detection

A counterfeit product is an illegal forgery or imitation of an original design. Because of the dominance of the contract foundry model, IP sharing/reuse, and outsourcing, the electronic products are increasingly vulnerable to piracy attack and counterfeiting. *IC metering* is a set of security protocols that enable the design house (authentic IP owner) to achieve post-fabrication control over their ICs [119, 132, 66, 133]. In *passive IC metering*, the IP rights owner is able to identify and monitor the devices [119, 132]. Passive metering can be directly enabled by certain types of PUFs. In *active IP metering*, in addition to identification and monitoring, the IP rights holder can actively

control, enable/disable, and authenticate a device [66]. We refer the interested readers to Chapter 8 of this book for a comprehensive survey of this topic. Addressing piracy attacks is notoriously hard since the adversaries are often financially strong, technologically advanced and informed of the design details. A set of open research questions have to do with developing security methods, PUF architectures, and controlled PUF protocols that can directly address the piracy attack models and counterfeiting.

### 7.5 Attacks and Vulnerability Analysis

To date, a number of attacks and countermeasures for PUFs are reported, see for example the detailed discussions in Section 4.2. However, PUFs have yet to undergo more refined cryptanalysis and evaluation of physical and side-channel attacks by a large community of researchers, similar to the way many traditional cryptographic primitives and protocols have been analyzed and attacked. For PUFs to be widely accepted, this seems to be a central future task that needs to be performed.

### 7.6 Formalization and Security Proofs

One relatively untouched area within physical cryptography and PUFs are the foundations of these fields. Formal definitions and security proofs for PUF-based protocols are just about to develop. For example, [71, 72] provide a thorough discussion of existing PUF definitions. [72] give new formal definitions for Strong PUFs that lead to a first reductionist security proof for a Strong PUF-based identification scheme. This type of work will likely prove essential for sound future development of the field, and will represent one of the major upcoming research topics within the area.

### 7.7 New Protocols and Applications

Up to now, PUFs and UNOs have mainly been used for authentication and identification purposes, and have mainly been seen as a security tool. But recently, a fundamental result indicated that PUFs possess a strong cryptographic potential: Oblivious transfer (and all protocols that can be derived from it) can be realized by

Strong PUFs [134]. Protocol design and optimization will thus be active future research topics.

## 8 Conclusion

Security and protection based on random physical media and objects is a fast-growing field that has recently enjoyed considerable research interest. Ensuring authenticity, security, protection, and integrity of data, hardware and software intellectual property, computers, networks, identities, and cyber-physical systems is a standing challenge. Traditional digital methods for these tasks often rely on digital labels or digitally stored secret keys that are vulnerable to forging, cloning, and other attacks. As discussed extensively in the previous sections, the unique and unclonable character of disordered physical structures can be exploited to address many of the vulnerabilities of these traditional concepts.

This chapter presented a new classification for the area of physical disorder based cryptography and security. We dealt with disorder-based identification, authentication, and other security methods. We then focused on four new classes of security devices based on physical disorder: Unique Objects, Weak Physical Unclonable Functions (Weak PUFs), Strong PUFs, and Controlled PUFs. Alongside with defining each class and discussing the history and relevant work, we described existing hardware implementations of these novel security primitives. We discussed emerging concepts in the area, including Timed Authentication and Public PUFs and SIMPL systems. We complemented the chapter by a treatment of future research challenges, which could prove helpful as a guideline to graduate students or anyone who wants to conduct research in the area.

## 9 Acknowledgement

The authors would like to thank Prof. Wayne P. Burleson for his valuable comments and suggestions. The authors would also like to thank Azalia Mirhoseini for her help with some of the figures.

# References


1. B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in Computer and Communication Security Conference, 2002.
2. B. Gassend, "Physical Random Functions," Master's thesis, Massachusetts Institute of Technology, Jan. 2003.
3. G. Suh, C. O'Donnell, and S. Devadas, "AEGIS: a Single-Chip secure processor," IEEE Design & Test of Computers, vol. 24, no. 6, pp. 570–580, 2007.
4. C. Yin and G. Qu, "LISA: maximizing RO PUF's secret extraction," in Hardware-Oriented Security and Trust (HOST), 2010, pp. 100–105.
5. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in Hardware-Oriented Security and Trust (HOST), 2008, pp. 67–70.
6. R. Maes, P. Tuyls, and I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs," in Cryptographic Hardware and Embedded Systems (CHES), 2009, pp. 332–347.
7. M.-D. M. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," IEEE Design and Test of Computers, vol. 27, pp. 48–65, 2010.
8. M. Majzoobi, F. Koushanfar, and S. Devadas, "FPGA PUF using programmable delay lines," in IEEE Workshop on Information Forensics and Security, 2010, p. in press.
9. M. Majzoobi and F. Koushanfar, "Time-Bounded Authentication of FPGAs," in Under Revision for IEEE Trans. on Information Forensics and Security (TIFS), 2011.
10. U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, and M. Stutzmann, "Security applications of diodes with unique current-voltage characteristics," Financial Cryptography and Data Security (FC), pp. 328–335, 2010.
11. G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in Design Automation Conference (DAC), 2007, pp. 9–14.
12. A. Sadeghi and D. Naccache, Eds., Towards Hardware-Intrinsic Security: Foundations and Practice. Springer, 2010.
13. D. Kirovski, "Anti-Counterfeiting: Mixing the Physical and the Digital World," in Towards Hardware-Intrinsic Security, A.-R. Sadeghi and D. Naccache, Eds. Springer, 2010, pp. 223–233.
14. S. Li and A. Jain, Eds., Encyclopedia of Biometrics. Springer, 2009.
15. D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition. Springer, 2009.
16. D. Kirovski, personal communication, Dagstuhl, Germany, 2008.
17. S. Graybeal and P. McFate, "Getting out of the STARTing block," Scientific American (USA), vol. 261, no. 6, 1989.
18. D. Bauder, "An anti-counterfeiting concept for currency systems," Research report PTK-11990. Sandia National Labs. Albuquerque, NM, 1983.
19. J. Brosow and E. Furugard, "Method and a system for verifying authenticity safe against forgery," US Patent 4,218,674, 1980.
20. G. Simmons, "A system for verifying user identity and authorization at the point-of sale or access," Cryptologia, vol. 8, no. 1, pp. 1–21, 1984.
21. ——, "Identification of data, devices, documents and individuals," in IEEE International Carnahan Conference on Security Technology, 1991, pp. 197–218.

22. J. Buchanan, R. Cowburn, A. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. Allwood, and M. Bryan, "Forgery:fingerprintingdocuments and packaging," Nature, vol. 436, no. 7050, p. 475, 2005.
23. J. Smith and A. Sutherland, "Microstructure based indicia," Proceedings of the Automatic Identification Advanced Technologies AutoID, vol. 99, pp. 79–83, 1999.
24. E. Métois, P. Yarin, N. Salzman, and J. Smith, "FiberFingerprint identification," in Workshop on Automatic Identification, 2002, pp. 147–154.
25. P. Seem, J. Buchanan, and R. Cowburn, "Impact of surface roughness on laser surface authentication signatures under linear and rotational displacements," Optics letters, vol. 34, no. 20, pp. 3175–3177, 2009.
26. A. Sharma, L. Subramanian, and E. Brewer, "Secure rural supply chain management using low cost paper watermarking," in ACM SIGCOMM workshop on Networked systems for developing regions, 2008, pp. 19–24.
27. F. Beekhof, S. Voloshynovskiy, O. Koval, R. Villan, and T. Pun, "Secure surface identification codes," in Proceedings of SPIE, vol. 6819, 2008, p. 68190D.
28. W. Clarkson, T. Weyrich, A. Finkelstein, N. Heninger, J. Halderman, and E. Felten, "Fingerprinting blank paper using commodity scanners," in IEEE Symposium on Security and Privacy, 2009, pp. 301–314.
29. The ProteXXion System, Bayer AG, http://www.research.bayer.com/edition-19/protexxion.aspx and http://www.research.bayer.com/edition-19/19_Protexxion_en.pdfx.
30. Ingeniatechnology, http://www.ingeniatechnology.com/.
31. G. DeJean and D. Kirovski, "RF-DNA: Radio-frequency certificates of authenticity," Cryptographic Hardware and Embedded Systems (CHES), pp. 346–363, 2007.
32. D. Kirovski, "Toward an automated verification of certificates of authenticity," in ACM Electronic Commerce (EC), 2004, pp. 160–169.
33. Y. Chen, M. Mihçak, and D. Kirovski, "Certifying authenticity via fiber-infused paper," ACM SIGecom Exchanges, vol. 5, no. 3, pp. 29–37, 2005.
34. P. Bulens, F. Standaert, and J. Quisquater, "How to strongly link data and its medium: the paper case," IET Information Security, vol. 4, no. 3, pp. 125–136, 2010.
35. Y. Kariakin, "Authentication of articles," Patent writing, WO/1997/024699, available from http://www.wipo.int/pctdb/en/wo.jsp?wo=1997024699, 1995.
36. G. Hammouri, A. Dana, and B. Sunar, "CDs have fingerprints too," Cryptographic Hardware and Embedded Systems (CHES), pp. 348–362, 2009.
37. D. Vijaywargi, D. Lewis, and D. Kirovski, "Optical DNA," Financial Cryptography and Data Security (FC), pp. 222–229, 2009.
38. B. Zhu, J. Wu, and M. Kankanhalli, "Print signatures for document authentication," in Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS). ACM, 2003, pp. 145–154.
39. J. Collins, "RFID Fibers for Secure Applications," RFID Journal, vol. 26, 2004.
40. RF SAW Inc., http://www.rfsaw.com/tech.html.
41. Creo Inc., http://www.creo.com.
42. Inkode Inc., http://www.inkode.com.
43. Microtag Temed Ltd, http://www.microtag-temed.com/.
44. CrossID Inc., Firewall Protection for Paper Documents, http://www.rfidjournal.com/article/articleview/790/1/44.
45. C. Loibl, "Entwurf und Untersuchung berührungslos abfragbarer einzigartiger Objekte," Master's thesis, Fachgebiet Höchstfrequenztechnik, Technische Universität München, 2009.

46. MagnePrint, http://www.magneprint.com/.
47. U. Rührmair, M. Stutzmann, P. Lugli, C. Jirauschek, K. Müller, H. Langhuth, G. Csaba, E. Biebl, and J. Finley, "Method and system for security purposes," European Patent Application Nr. EP 09 157 041.6, March 2009.
48. C. Clelland, V. Risca, and C. Bancroft, "Hiding messages in DNA microdots," Nature, vol. 399, no. 6736, pp. 533–534, 1999.
49. November AG, http://www.november.de/archiv/pressemitteilungen/pressemitteilung/article/sichere-medikamente-dank-dna-codes-der-identif-gmbh.html.
50. D. Kirovski, "A point-set compression heuristic for fiber-based certificates of authenticity," in Data Compression Conference (DCC), 2005, pp. 103–112.
51. ——, "Point compression for certificates of authenticity," in Data Compression Conference (DCC), 2004, p. 545.
52. Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in Advances in cryptology-Eurocrypt 2004. Springer, 2004, pp. 523–540.
53. Alliance for Gray Market and Counterfeit Abatement (AGMA), http://www.agmaglobal.org/.
54. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in ACM Conference on Computer and Communications Security (CCS), 2010, pp. 237–249.
55. C. Bennett, G. Brassard, S. Breidbart, and S. Wiesner, "Quantum cryptography, or unforgeable subway tokens," in Advances in Cryptology–Proceedings of Crypto, vol. 82, 1983, pp. 267–275.
56. C. Bennett, G. Brassard, et al., "Quantum cryptography: Public key distribution and coin tossing," in International Conference on Computers, Systems and Signal Processing, vol. 175. Bangalore, India, 1984.
57. J. Guajardo, S. Kumar, G. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in Cryptographic Hardware and Embedded Systems (CHES), 2007, pp. 63–80.
58. K. Lofstrom, W. R. Daasch, and D. Taylor, "Ic identification circuit using device mismatch," in ISSCC, 2000, pp. 372–373.
59. P. Layman, S. Chaudhry, J. Norman, and J. Thomson, "Electronic fingerprinting of semiconductor integrated circuits," US Patent 6,738,294, September 2002.
60. Y. Su, J. Holleman, and B. Otis, "A 1.6pJ/bit 96 (percent) stable chip ID generating circuit using process variations," in IEEE International Solid-State Circuits Conference (ISSCC), 2007, pp. 200–201.
61. D. Holcomb, W. Burleson, and K. Fu, "Initial SRAM state as a fingerprint and source of true random numbers for RFID tags," in Proceedings of the Conference on RFID Security, 2007.
62. P. Tuyls, G.-J. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings," in Cryptographic Hardware and Embedded Systems (CHES), 2006, pp. 369–383.
63. R. Helinski, D. Acharyya, and J. Plusquellic, "A physical unclonable function defined using power distribution system equivalent resistance variations," in Design Automation Conference (DAC), 2009, pp. 676–681.
64. ——, "Quality metric evaluation of a physical unclonable function derived from an IC's power distribution system," in Design Automation Conference, ser. DAC, 2010, pp. 240–243.

65. G. E. Suh, "AEGIS: A Single-Chip Secure Processor," Ph.D. dissertation, Massachusetts Institute of Technology, Aug 2005.
66. Y. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in USENIX Security Symposium, 2007, pp. 291–306.
67. D. Holcomb, W. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," IEEE Transactions on Computers, vol. 58, no. 9, pp. 1198–1210, September 2009.
68. R. Pappu, "Physical one-way functions," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.
69. R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," Science, vol. 297, pp. 2026–2030, 2002.
70. B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, " Controlled Physical Random Functions ," in Annual Computer Security Applications Conference, 2002.
71. U. Rührmair, F. Sehnke, and J. Sölter, "On the Foundations of Physical Unclonable Functions," Cryptology ePrint Archive, International Association for Cryptologic Research, Tech. Rep., 2009.
72. U. Rührmair, H. Busch, and S. Katzenbeisser, "Strong PUFs: Models, Constructions, and Security Proofs," in Towards Hardware-Intrinsic Security, A.-R. Sadeghi and D. Naccache, Eds. Springer, 2010, pp. 79–96.
73. B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Delay-Based Circuit Authentication and Applications," in Symposium on Applied Computing (SAC), 2003.
74. J.-W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits with identification and authentication applications," in IEEE VLSI Circuits Symposium, New-York, June 2004.
75. D. Lim, "Extracting Secret Keys from Integrated Circuits," Master's thesis, Massachusetts Institute of Technology, may 2004.
76. B. Gassend, D. Lim, D. Clarke, M. van Dijk, and S. Devadas, "Identification and authentication of integrated circuits," Concurrency and Computation: Practice and Experience, vol. 16, no. 11, pp. 1077–1098, 2004.
77. M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Techniques for design and implementation of secure reconfigurable pufs," ACM Transactions on Reconfigurable Technology and Systems (TRETS), vol. 2, no. 1, 2009.
78. S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications," in Proceedings of 2008 IEEE International Conference on RFID (RFID 2008), May 2008, pp. 58–64.
79. D. Suzuki and K. Shimizu, "The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes," Cryptographic Hardware and Embedded Systems (CHES), 2010, pp. 366–382.
80. S. Devadas and B. Gassend, "Authentication of integrated circuits," US Patent 7,840,803, 2010, application in 2002.
81. Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak, "Trusted integrated circuits: A nondestructive hidden characteristics extraction approach," in Information Hiding (IH), 2008, pp. 102–117.
82. M. Potkonjak and F. Koushanfar, "Identification of integrated circuits," US Patent Application 12/463,984; Publication Number: US 2010/0287604 A1, May 2009.

83. F. Koushanfar, P. Boufounos, and D. Shamsi, "Post-silicon timing characterization by compressed sensing," in International Conference on Computer-Aided Design (ICCAD), 2008, pp. 185–189.
84. D. Shamsi, P. Boufounos, and F. Koushanfar, "Noninvasive leakage power tomography of integrated circuits by compressive sensing," in International Symposium on Low Power Electronic Designs (ISLPED), 2008, pp. 341–346.
85. M. Nelson, A. Nahapetian, F. Koushanfar, and M. Potkonjak, "Svd-based ghost circuitry detection," in Information Hiding (IH), 2009, pp. 221–234.
86. S. Wei, S. Meguerdichian, and M. Potkonjak, "Gate-level characterization: Foundations and hardware security applications," in Design Automation Conference (DAC), 2010.
87. F. Koushanfar and A. Mirhoseini, "A unified framework for multimodal submodular integrated circuits trojan detection," IEEE Trans. on Information Forensic and Security (TIFS), 2011.
88. G. Csaba, X. Ju, Z. Ma, Q. Chen, W. Porod, J. Schmidhuber, U. Schlichtmann, P. Lugli, and U. Ruhrmair, "Application of mismatched cellular nonlinear networks for physical cryptography," in International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA). IEEE, 2010, pp. 1–6.
89. P. Tuyls and B. Škorić, "Strong Authentication with Physical Unclonable Functions," Security, Privacy, and Trust in Modern Data Management, pp. 133–148, 2007.
90. U. Rührmair, "SIMPL Systems, Or: Can we construct cryptographic hardware without secret key information?" in International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), ser. Lecture Notes in Computer Science, vol. 6543. Springer, 2011.
91. U. Rührmair, C. Jaeger, and M. Algasinger, "An Attack on PUF-based Session Key Exchange and a Hardware-based Countermeasure," Financial Cryptography and Data Security (FC), 2011, to appear.
92. M. Majzoobi, A. E. Nably, and F. Koushanfar, "FPGA Time-Bounded Authentication," in Information Hiding Conference (IH), 2010, pp. 1–15.
93. J. Bekenstein, "How does the entropy/information bound work?" Foundations of Physics, vol. 35, no. 11, pp. 1805–1823, 2005.
94. E. Oztürk, G. Hammouri, and B. Sunar, "Towards robust low cost authentication for pervasive devices," in Pervasive Computing and Communications (PerCom), 2008, pp. 170–178.
95. M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Testing techniques for hardware security," in International Test Conference (ITC), 2008, pp. 1–10.
96. ——, "Lightweight secure PUF," in International Conference on Computer Aided Design (ICCAD), 2008, pp. 670–673.
97. U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, and G. Csaba, "Applications of high-capacity crossbar memories in cryptography," IEEE Transactions on Nanotechnology, no. 99, p. 1.
98. C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, and M. Stutzmann, "Random pn-junctions for physical cryptography," Applied Physics Letters, vol. 96, p. 172103, 2010.
99. P. Tuyls, B. Skoric, S. Stallinga, A. H. M. Akkermans, and W. Ophey, "Information-theoretic security analysis of physical uncloneable functions," in Financial Cryptography and Data Security (FC), 2005, pp. 141–155.
100. B. Škorić, "On the entropy of keys derived from laser speckle; statistical properties of Gabor-transformed speckle," Journal of Optics A: Pure and Applied Optics, vol. 10, p. 055304, 2008.

101. B. Skoric, S. Maubach, T. Kevenaar, and P. Tuyls, "Information-theoretic analysis of capacitive physical unclonable functions," Journal of Applied Physics, vol. 100, no. 2, p. 024902, 2009.
102. I. Kim, A. Maiti, L. Nazhandali, P. Schaumont, V. Vivekraja, and H. Zhang, "From Statistics to Circuits: Foundations for Future Physical Unclonable Functions," Towards Hardware-Intrinsic Security, pp. 55–78, 2010.
103. F. Sehnke, J. Schmidhuber, and U. Rührmair, "Security Benchmarks for Strong Physical Unclonable Functions," 2010, in submission.
104. B. Gassend, M. van Dijk, D. Clarke, E. Torlak, S. Devadas, and P. Tuyls, "Controlled physical random functions and applications," ACM Transactions on Information and System Security (TISSEC), vol. 10, no. 4, pp. 1–22, 2008.
105. B. S. Yee, "Using secure coprocessors," Ph.D. dissertation, Carnegie Mellon University, 1994.
106. A. Carroll, M. Juarez, J. Polk, and T. Leininger, "Microsoft "palladium": A business overview," in Microsoft Content Security Business Unit, August 2002. [Online]. Available: http://www.microsoft.com/presspass/features/2002/jul02/0724palladiumwp.asp
107. T. Alves and D. Felton, "Trustzone: Integrated hardware and software security," ARM white paper, jul 2004.
108. Microsoft, "Next-Generation Secure Computing Base," http://www.microsoft.com/ `resources/ngscb/defaul.mspx`.
109. T. C. Group, "Tcg specification architecture overview revision 1.2," http://www.trustedcomputinggroup.com/home, 2004.
110. D. Lie, C. Thekkath, M. Mitchell, P. Lincoln, D. Boneh, J. Mitchell, and M. Horowitz, "Architectural support for copy and tamper resistant software," in Int'l Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), 2000, pp. 168–177.
111. D. Lie, "Architectural support for copy and tamper-resistant software," Ph.D. dissertation, Stanford University, Dec 2003.
112. G. E. Suh, D. Clarke, B. Gassend, M. van Dijk, and S. Devadas, "AEGIS: Architecture for tamper-evident and tamper-resistant processing," in Int'l Conference on Supercomputing (MIT-CSAIL-CSG-Memo-474 is an updated version), 2003.
113. G. E. Suh, C. W. O'Donnell, I. Sachdev, and S. Devadas, "Design and implementation of the AEGIS single-chip secure processor using physical random functions," in International Symposium on Computer Architecture (ISCA), 2005.
114. S. Devadas, "Non-networked rfid puf authentication," US Patent Application 12/623,045, 2008.
115. E. Oztiirk, G. Hammouri, and B. Sunar, "Towards robust low cost authentication for pervasive devices," in International Conference on Pervasive Computing and Communications (PerCom), 2008, pp. 170 –178.
116. N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in Information Hiding. Springer, 2009, pp. 206–220.
117. M. Potkonjak, "Secure authentication," US Patent Application 12/464,387; Publication Number: US 2010/0293612 A1, May 2009.
118. ——, "Digital signatures," US Patent Application 12/464,384; Publication Number: US 2010/0293384 A1, May 2009.
119. F. Koushanfar, G. Qu, and M. Potkonjak, "Intellectual property metering," in International Workshop on Information Hiding (IHW), 2001, pp. 81–95.

120. F. Koushanfar and M. Potkonjak, "Cad-based security, cryptography, and digital rights management," in Design Automation Conference (DAC), 2007, pp. 268–269.
121. M. Potkonjak, S. Meguerdichian, and J. Wong, "Trusted sensors and remote sensing," in IEEE Sensors, 2010, pp. 1–4.
122. U. Rührmair, M. Stutzmann, G. Csaba, U. Schlichtmann, and P. Lugli, "Method for security purposes," European Patent Filings EP 09003764.9, EP 09003763.1, EP 09157043.2, March 2009.
123. U. Rührmair, "SIMPL Systems: On a Public Key Variant of Physical Unclonable Functions," Cryptology ePrint Archive, International Association for Cryptologic Research, Tech. Rep., 2009.
124. U. Rührmair, Q. Chen, M. Stutzmann, P. Lugli, U. Schlichtmann, and G. Csaba, "Towards Electrical, Integrated Implementations of SIMPL Systems," Cryptology ePrint Archive, International Association for Cryptologic Research, Tech. Rep., 2009.
125. Q. Chen, G. Csaba, X. Ju, S. Natarajan, P. Lugli, M. Stutzmann, U. Schlichtmann, and U. Ruhrmair, "Analog circuits for physical cryptography," in 12th International Symposium on Integrated Circuits (ISIC'09), Singapore, December 14 – 16, 2009. IEEE, 2009/2010, pp. 121–124.
126. U. Rührmair, Q. Chen, M. Stutzmann, P. Lugli, U. Schlichtmann, and G. Csaba, "Towards electrical, integrated implementations of simpl systems," in Workshop in Information Security Theory and Practice (WISTP), 2010, pp. 277–292.
127. Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, M. Stutzmann, and U. Rührmair, "Circuit-based approaches to SIMPL systems," Journal of Circuits, Systems, and Computers, vol. 20, pp. 107–123, 2011.
128. U. Rührmair, "SIMPL Systems as a Cryptographic and Security Primitive," in To be submitted to IEEE Trans. on Information Forensics and Security (TIFS), 2011.
129. B. Škorić, "Quantum Readout of Physical Unclonable Functions," Progress in Cryptology–AFRICACRYPT 2010, pp. 369–386, 2010.
130. B. koric, "Quantum readout of physical unclonable functions," in Progress in Cryptology (AFRICACRYPT), ser. Lecture Notes in Computer Science, D. Bernstein and T. Lange, Eds. Springer Berlin / Heidelberg, 2010, vol. 6055, pp. 369–386.
131. C. Bösch, J. Guajardo, A. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," in Cryptographic Hardware and Embedded Systems (CHES), 2008, pp. 181–197.
132. F. Koushanfar and G. Qu, "Hardware metering," in Design Automation Conference (DAC), ser. DAC, 2001, pp. 490–493.
133. Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in ICCAD, 2007.
134. U. Rührmair, "Oblivious transfer based on physical unclonable functions (extended abstract)," in TRUST, ser. Lecture Notes in Computer Science, A. Acquisti, S. W. Smith, and A.-R. Sadeghi, Eds., vol. 6101. Springer, 2010, pp. 430–440.