Security Down to the Hardware Level

8

Anastacia Alvarez and Massimo Alioto

This chapter introduces the concept of Physically Unclonable Functions (PUFs), their prospects for hardware security in IoT devices, and their interaction with traditional cryptography. Section 8.1 summarizes the background on PUFs, whereas Sect. 8.2 covers the metrics that are commonly used to evaluate PUF performance. Such metrics are used to comparatively review the state of the art on PUFs in Sect. 8.3. Section 8.4 covers vulnerabilities to malicious attacks attempting to clone or mimic a PUF. In the last section, we introduce the novel concept of PUF-enhanced cryptography as a promising direction aiming to merge PUFs and cryptography in a cohesive framework for IoT hardware-level security.

8.1 Physically Unclonable Functions for IoT

The spatial pervasiveness and the prospectively very large number of deployed nodes monitoring the environment, people, shared resources and goods, makes security a fundamental challenge in IoT. Serious security issues are indeed arising in terms of data authenticity, integrity and confidentiality. Indeed, it is typically necessary to assure that the data and the sender are legitimate, the data has been sent uncorrupted, and oftentimes data needs to be unreadable from an unintended receiver. Accordingly, IoT requires security to be assured down to the hardware level, as the authenticity and the integrity need to be assured also in terms of the hardware implementation of each IoT node (i.e., each node needs to be confirmed to be authentic and intact, while signaling if it has been counterfeited or tampered with).

In the recent past, Physically Unclonable Functions (PUFs) have emerged as potentially highly secure and lightweight solution to ensure data and hardware security, assuring trustworthiness down to the chip level (Mathew et al. 2014; Maes et al. 2012; Rosenblatt et al. 2013; Su et al. 2007; Maes 2012). A PUF is a function that maps an input (digital) challenge to an output (digital) response in a repeatable but unpredictable manner, leveraging on chip-specific random process variations. PUFs are sometimes referred to as "silicon biometrics", i.e. something equivalent to a "chip fingerprint" that is unique for each die. As such, it eliminates the need to store any key, as the latter is naturally generated and embedded into the chip during its manufacturing. This avoids the need for key programming (e.g., via fuses or e-Flash), and makes IoT nodes less prone to the many existing attacks that uncover the content of memories (Nedospasov et al. 2013), as discussed below.

PUFs are used for chip identification and authentication (Rosenblatt et al. 2013; Su et al.

A. Alvarez (✉) • M. Alioto
National University of Singapore, Singapore, Singapore
e-mail: anastacia@u.nus.edu

M. Alioto (ed.), *Enabling the Internet of Things*, DOI 10.1007/978-3-319-51482-6_8

2007; Maes 2012; Alvarez et al. 2015; Gassend et al. 2002), secure key storage and lightweight encryption (Mathew et al. 2014; Xu et al. 2014), hardware-entangled cryptography (Sadeghi and Naccache 2010) and identification of malicious hardware (Maes 2013). Chip identification and authentication are typically performed by preliminarily storing all challenge-response pairs (CRPs) of the chip PUF in a secure database, during a first enrollment phase. These (or a subset thereof) are used to verify the response of the chip to a given challenge during in-field operation, making sure not to reuse CRPs to reduce susceptibility to cloning, and counteract replay attacks. Figure 8.1 shows an illustration of the enrollment process and chip authentication.

To keep data secure during transmission, it is typically encrypted using a key that is stored externally, or in an on-chip non-volatile memory (NVM). Unfortunately, storing the key off chip or in an on-chip NVM facilitates the recovery of the key by other parties. Indeed, several studies have shown that NVM are prone to attacks and easy to read out (Samyde et al. 2002; Kömmerling et al. 1999). PUFs replace the conventional key storage, and hence offer superior robustness against invasive attacks, as they do not really store information since they recreate the keys only while the chip is being powered on.

8.2 PUF Properties and Metrics

Ideally, an array of PUF bitcells generate chip-specific keys that are:

- unpredictable, leveraging on on-chip random process variations

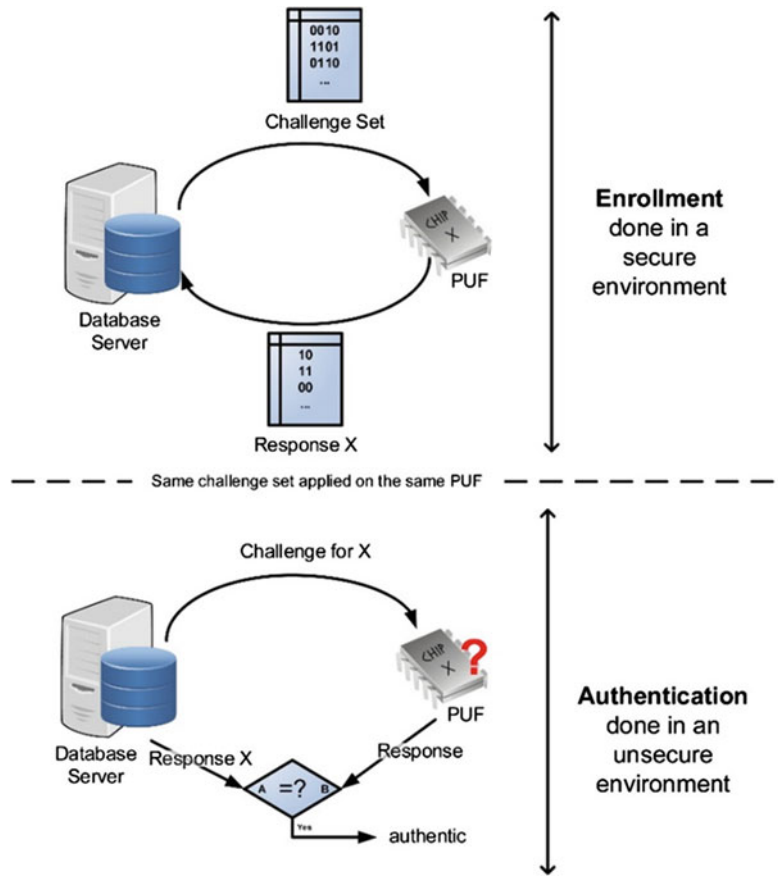

Fig. 8.1 Illustration of typical chip enrollment and subsequent in-field authentication using challenge-response pairs (CRPs) from PUFs

- repeatable, by amplifying random variations, while rejecting global variations and noise (Maes et al. 2012)
- not directly accessible or measurable externally, once the enrollment phase is completed.

There are two main types of PUFs: weak PUFs and strong PUFs. Weak PUFs have limited number of challenge-response pairs, making them equivalent to random key generators that are typically used for encryption and decryption. Weak PUFs essentially provide chip ID, whereas strong PUFs offer a very large number of challenge-response pairs (CRPs), each for one-time use. Given the long lifespan required by IoT applications, PUFs with very large number of CRPs (and therefore large area) are very expensive and typically infeasible. As a numerical example, Table 8.1 shows an example of the cost for a PUF with 256-bit key in 65 nm (Alvarez et al. 2015; Alvarez et al. 2016), whose cost invariably exceeds the overall IoT node cost.

Given the fundamental PUF properties, such as stability, repeatability, uniqueness and randomness (Maes 2013), and knowing the statistical nature of process variations, several metrics have been introduced to quantify the quality of PUF bitcells. In the following, such metrics are summarized in Table 8.2, where typical values based on current literature are also reported.

In detail, any PUF output should ideally remain the same under fluctuating environmental conditions (e.g., voltage, temperature), and at any process corner. Actual PUFs are not able to provide perfectly stable outputs, due to non-perfect rejection of noise, global and environmental variations. Stability is measured by counting all bits that become unstable across repeated PUF evaluations and environmental conditions, within the specified range of voltage and temperature of operation.

Repeatability (or reproducibility) and uniqueness are measured from the Hamming Distance (HD) across several measurements of PUF keys. Such measurements are compared to a reference "golden" key (Maes 2013) that is taken as the first measurement under nominal conditions. Repeatability is the average intra-PUF Hamming Distance (HD) between the golden key and several key evaluations with the same challenge in the same chip, under different environmental conditions. By definition, highly reproducible PUFs should have low intra-PUF Hamming distance (ideally zero). Uniqueness, on the other hand, is taken as the average inter-PUF HD between the golden key and key evaluations from different chips under the same PUF input (Mathew et al. 2014). The inter-PUF HD should be close to the ideal value equal to half the length of the PUF key (e.g., the ideal inter-PUF HD of a 256-bit key is 128). Alternatively, the fractional Hamming Distance (FHD) can be used to

Table 8.1 Example of SRAM PUF silicon cost (assumed to be 5 cents/mm^2, with area/bit representative of very dense SRAM PUFs)

(Encrypted) data transmitted every	PUF capacity (MB)	PUF area (mm^2)	Silicon cost (US$)
1 h	5	24	1.2
10 min	32	147	7.4
1 min	320	1478	74

Table 8.2 PUF metrics and typical values

Metric	Measured by	Typical value	Ideal value
Stability	Unstable bits	1–60%	0
Repeatability	Intra-PUF FHD	0.8–15%	0
Uniqueness	Inter-PUF FHD	30–60%	50%
Identifiability	Inter/intra HD	5–80	∞
Randomness	0/1 bias	40–60%	50%

quantify reproducibility and uniqueness (Maes et al. 2012), where the Hamming distance is simply expressed as a percentage of the key length, or the number of bits N in a PUF key (ideal inter-PUF FHD is 50%). Identifiability quantifies the distinguishability of a PUF instance to other instances, and is loosely taken as the ratio of the inter-PUF and intra-PUF HD (on the assumption that it is both repeatable and unique), where a larger value is desired (Maes 2013; Mathew et al. 2014; Yang et al. 2015).

Figure 8.2 shows an example of probability distribution function of reproducibility (intra-PUF FHD) and uniqueness (inter-PUF FHD). A perfectly identifiable PUF ideally has no intersection between the inter-PUF and intra-PUF curves, which means that a single PUF response is enough to determine whether the chip is authentic or not. In practical cases, the two curves in Fig. 8.2 have an intersection, and an optimal decision threshold needs to be chosen to determine whether a given PUF is identifiable. As shown in Fig. 8.2, such decision threshold is set by the point where Type I and Type II errors are minimized. Type I error is the false positive, where an invalid key is accepted as a valid one. Type II error, on the other hand, is the false negative, where a valid key is discarded as an invalid one.

Regarding chip authentication, false rejection rate (FRR) and false acceptance rate (FAR) can be used as relevant metrics to assess its quality and security level (Stanzione et al. 2009; Lim et al. 2005). Referring to Fig. 8.2, FRR corresponds to the probability of having an output with FHD under the false negative area, whereas FAR corresponds to the area under the false positive area. Accordingly, the PUF yield Y can be defined as the probability that no authentication error occurs during the lifetime of a given PUF chip, i.e. $Y = 1 - FAR - FRR$. The bit error rate (BER) or the percentage of unstable bits can also be used as a metric of the quality of chip authentication, when the whole array is considered, rather than dividing the array into keys of length N.

Another important property of PUFs is the randomness of its responses, as needed to ensure

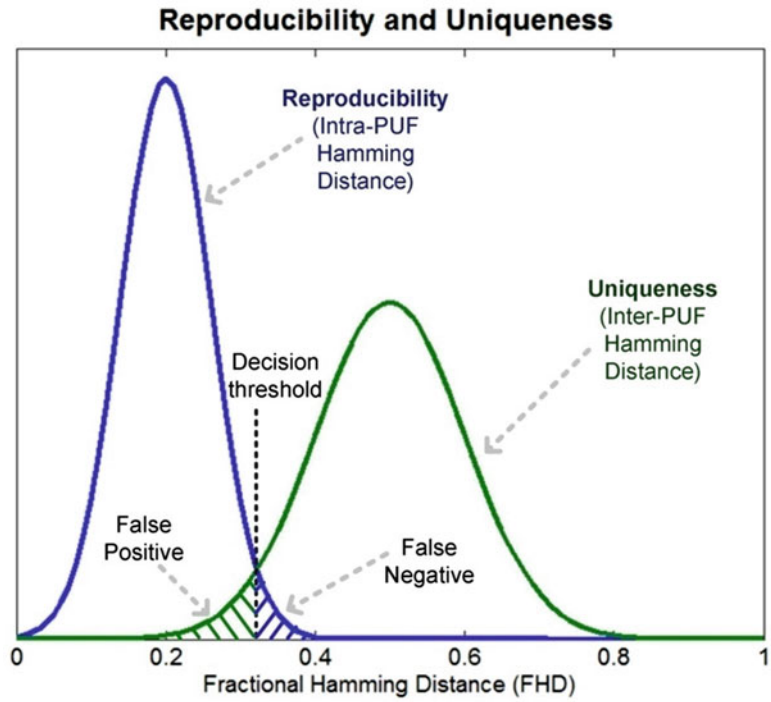

Fig. 8.2 Sample Inter- and Intra-PUF FHD showing decision threshold and Type I (false positive) and Type II (false negative) errors

their unpredictability. Randomness is routinely quantified through the statistical characterization in terms of 0/1 bias (defined as the probability of having a 1 in a PUF output bit (Yu et al. 2012)), the entropy (Mathew et al. 2014), and more thoroughly through the NIST randomness tests (Rukhin et al. 2010) (see below). To quantify the randomness of PUF responses across different positions of PUF bitcell within the die, the autocorrelation function (ACF) is routinely used to detect repeating or correlated patterns among different responses (Mathew et al. 2014; Yang et al. 2015). The correlation between PUF output bits is generally due to layout-dependent variations (Alvarez et al. 2015, 2016; Li et al. 2015). Visually, randomness can be represented in the form of the speckle diagram shown in Fig. 8.3, where each pixel represents a PUF bitcell and the PUF output 0's (1's) are represented with black (white) pixels. In Fig. 8.3, the distribution looks somewhat random (i.e., there are no clear patterns) and the 0/1 bias is also close to ideal value of 0.5.

The NIST statistical test suite (Rukhin et al. 2010) is a set of tests to quantify the randomness of a stream of bits. Version 2.1.2 contains 15 tests, each one exercising one property to test randomness. The simplest test is the frequency test, which computes the 0/1 ratio of the whole bitstream. For each of the tests, certain parameters need to be preliminarily set (e.g., length of bitstream n, block size M). Table 8.3 shows the complete list of the tests and parameters to be set.

IoT devices are tightly energy constrained, since they are either battery operated or energy harvested, hence the energy consumption of the PUF is another important metric. To abstract the energy from the PUF organization and size, the most commonly adopted metric is the energy per bit, obtained by dividing the average energy per access by the number of bits within the key. The energy per bit of existing PUFs typically ranges from tens of fJ/bit to tens of pJ/bit (Alvarez et al. 2015, 2016).

Due to the stringent cost requirement in IoT nodes (including silicon area), another important PUF metric is the effective area per bit, as obtained by considering the actual number of available PUF bits obtained after removing unstable bits, and including the area cost of the circuitry performing post-processing on the raw PUF output (see later). Robustness to ageing and chip lifetime are assessed through accelerated ageing tests (Puntin et al. 2008; Stanzione et al. 2009; Selimis et al. 2011). Modeling complexity, in terms of the number of brute force trials needed to model the PUF, can likewise be used to characterize PUFs (Stanzione et al. 2009).

8.3 PUF Topologies and State of the Art

The concept of PUFs have been introduced in the early 2000s, and they have been initially referred to as ICID (Lofstrom et al. 2000), Physical One-Way Functions (POWF) (Pappu et al.

Fig. 8.3 Sample speckle diagram from (Alvarez et al. 2015)

Table 8.3 NIST statistical test suite

NIST test	Description	Minimum stream length n	Other parameters
Frequency Test	Takes ratio of number of 1's and 0's	100	–
Frequency test within a block	Ratio of 1's and 0's with M-bit block	100	$M \geq 20$ $M > 0.01n$
Runs test	Relative oscillation of bit stream	100	–
Longest Run of Ones	Length of longest consecutive 1's with a block	128	M (set based on present n)
Binary Matrix Run	Rank of disjoint sub-matrix	$38 \cdot M \cdot Q$	M, Q
DFT	Detect periodic features	10^3	–
Non-overlapping Template	Detect occurrence of patterns in an m-bit window	10^6	$m = [2,10]$
Overlapping Template	Detects occurrence of patterns, with overlaps included	10^6	$m = [2,10]$
Universal Statistical Test	Number of bits between matching patterns	387,840	$L = [6,16]$ $Q = 10 * 2^L$
Linear Complexity Test	Length of equivalent LFSR	10^6	M
Serial Test	Detect frequency of overlapping patterns	–	$m < \log_2 n - 2$
Approximate Entropy	Detect frequency of overlapping patterns	–	$m < \log_2 n - 5$
Cumulative Sums	Random walk	100	–
Random Excursions Test	Random walk cycle	10^6	–
Random Excursions Variant Test	Deviations from a random walk	10^6	–

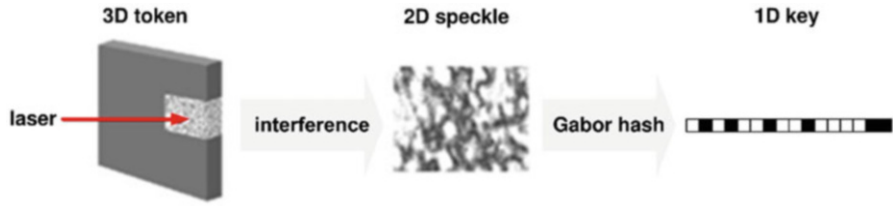

Fig. 8.4 Illustration of the Physical One-Way Function from a non-homogenous material (Pappu et al. 2002)

2002), or Physical Random Functions (PRF) (Gassend et al. 2002), among others. ICID uses an array of MOSFET to generate the random values from random process mismatch, via FET drain current. The physical one-way function was proposed as a solution to the need for a one-way function (easy to evaluate but difficult to invert) for cryptographic applications. The approach uses a laser to scatter light through an inhomogeneous structure (at some precise angle, which serves as the challenge), as shown in Fig. 8.4. The resulting optical speckle diagram is hashed to obtain the key. Most of the literature has then reverted to silicon-based solutions, leveraging the low-cost and high-volume capability of CMOS chips.

Most of the existing silicon PUFs can be classified as either delay-based or memory-based PUFs (Gassend et al. 2002; Guajardo et al. 2007; Suh et al. 2007; Kumar et al. 2008). In delay-based PUFs, bits are generated by comparing the delay of two nominally identical paths. The sign of the random delay difference between the two delays determines the output bit. One of the earliest implementation of such a concept is the ring oscillator (RO) PUF (Gassend et al. 2002; Suh et al. 2007), whose digital output is determined by the relative frequency of each

selected pair of nominally identical ring oscillators. Figure 8.5a shows the general diagram of a ring oscillator PUF, where the challenge selects two of the available ring oscillators, and the corresponding response depends on whether the frequency of the first selected oscillator is greater than the second or not. Knowing that these inverter chain ring oscillators tend to be very sensitive to environmental conditions, several techniques have been introduced to improve the high native instability rate, and poor statistical quality of this pair-wise comparison. Some of these techniques include the adoption of k-sum or 1-out-of-k masking techniques (Suh et al. 2007; Lee et al. 2004).

Another delay-based PUF is the arbiter PUF (Suh et al. 2007; Lim et al. 2005; Lee et al. 2004), as shown in Fig. 8.5b. It compares the delay of two delay lines, and suffers from the same limitations as the RO PUF (Gassend et al. 2002; Lee et al. 2004). A improved delay-based version was recently proposed (Yang et al. 2015), based on the oscillation collapse in an even-stage ring of delay-adjustable stages. The delay is set by an applied input (PUF challenge) via inverter replica multiplexing. The native instability of PUF outputs was substantially reduced at the cost of much higher energy and the need for CTAT biasing.

All above delay-based PUFs are also intrinsically vulnerable to PUF modeling attacks, which can capture and clone the content of the entire PUF with very low effort. Indeed, the PUF output is dictated by the overall PUF delay, which is in turn defined by the sum of the delays of cascaded stages. Since each stage delay is fixed (although unpredictable), identifying all stage delays from the analysis of the PUF outputs entails only a linear complexity, making the PUF easy to clone (Rührmair et al. 2010).

In memory-based PUFs, a bistable structure of two cross-coupled inverters is used to generate the output bits. They leverage on the natural tendency of cross-coupled inverters to resolve

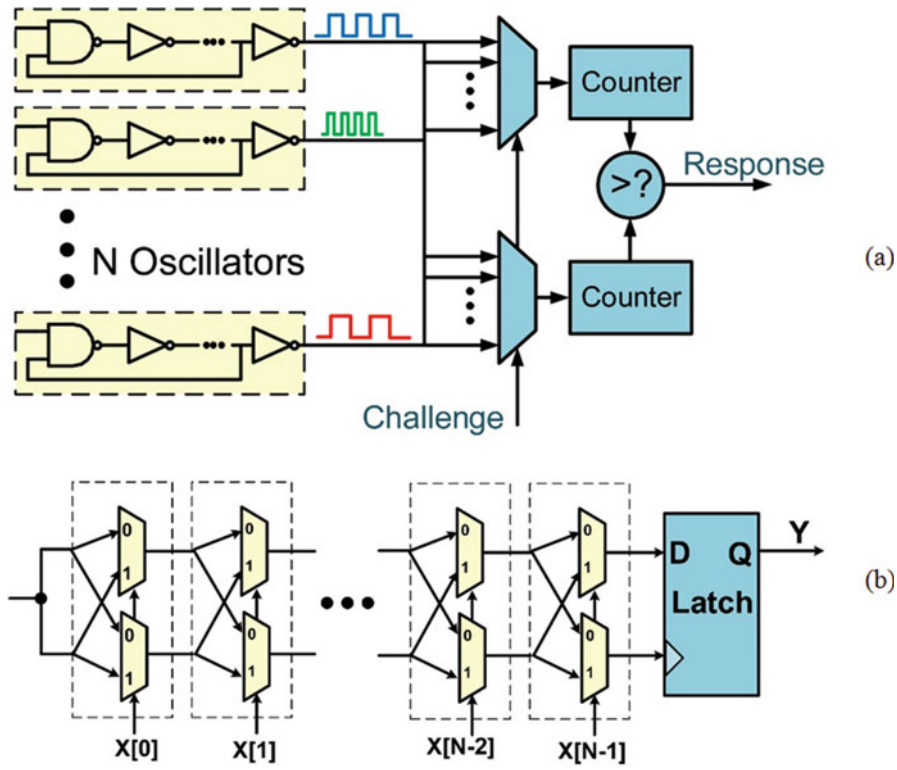

Fig. 8.5 Delay-based PUFs: (**a**) ring oscillator (RO) PUF (Suh et al. 2007), (**b**) arbiter PUF (Lee et al. 2004)

to a preferred state at the power-up, as determined by their asymmetry due to random variations (Suh et al. 2007). For example, SRAM PUFs leverage this property in SRAM bitcells (Guajardo et al. 2007; Holcomb et al. 2009). Other similar PUFs are the Latch PUF (Su et al. 2007), DFF PUF (Maes et al. 2008), butterfly PUF (Kumar et al. 2008), and the buskeeper PUF (Simons et al. 2012), which is similar to the SRAM PUF albeit without the write ability, as access transistors are removed since PUF bitcells are read-only. The butterfly PUF (Kumar et al. 2008) follows the same concept of leveraging on the unstable state of cross-coupled inverters. It was proposed for implementation in an FPGA and uses the available cross-coupled latches instead of inverters, as shown in Fig. 8.6. The operation starts by asserting the *excite* signal, thereby forcing the PUF to be in the unstable state. This signal is then released and after a few clock cycles, *out* signal settles to its natural stable state determined by the random variations in the related logic gates.

The recent literature on memory-based PUFs and their experimental characterization has shown that PUFs typically have poor stability (Schrijen et al. 2012), and are highly vulnerable to semi-invasive attacks such as electrical and optical probing (Nedospasov et al. 2013). The same vulnerability to semi-invasive attacks is found in other PUFs that rely on the same principle, such as senseamp (Bhargava et al. 2010; Bhargava and Mai 2014). For such PUFs, reasonable levels of stability are typically achieved through substantial temporal redundancy at the expense of energy consumption (Helinski et al. 2009). Other proposed PUFs are based on (1) the glitch generated in digital paths, although they suffer from high instability rates (Suzuki et al. 2010), (2) the difference in leakage current generated by nominally identical transistors, although at the cost of large energy due to the necessary circuitry for current/voltage references and opamp (Ganta et al. 2011), (3) DRAM errors under different wordline voltage, although such PUFs are highly vulnerable to non-invasive attacks (Rosenblatt et al. 2013), (4) open or short connection in vias (Choi et al. 2014), (5) oxide breakdown in OTP ROMs (Liu et al. 2010), (6) capacitance mismatch (Tuyls et al. 2006; Roy et al. 2009; Wan et al. 2015), (7) the variations in the supply network resistance, although this requires the generation of very large currents (Helinski et al. 2009).

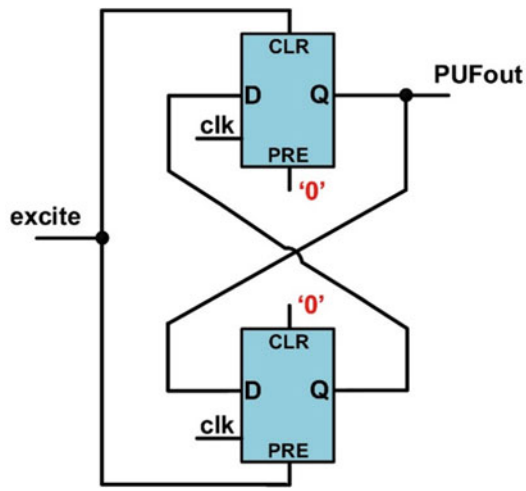

Fig. 8.6 Butterfly PUF (Kumar et al. 2008)

A hybrid PUF was proposed in (Satpathy et al. 2014; Mathew et al. 2014, 2016) combining delay and metastability as sources of randomness. The basic bitcell is shown in Fig. 8.7, where bistability is forced through the pre-charge transistors controlled by *clk0* and *clk1*. The randomness in delay is introduced through the clock skew between *clk0* and *clk1*. In order to reduce unstable bits, significant temporal majority voting is employed. Soft dark bit masking was also used in (Satpathy et al. 2014) by modulating the load in the *bit* and *bit'*, and masking bits that become unstable with the change in the load. Indeed, load modulation simply injects controlled perturbation in the stability of the PUF bitcell, which in turn permits to identify the truly stable bitcells that do not change output even in the presence of such perturbation.

In order to achieve adequate native stability in spite of voltage and temperature fluctuations, authors in (Li et al. 2015) proposed to use a proportional-to-absolute-temperature (PTAT) as a bitcell. Figure 8.8 shows the bitcell and the

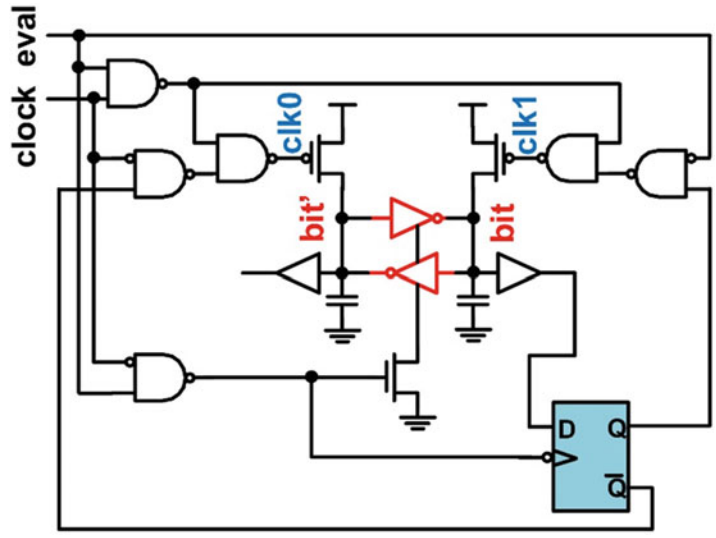

Fig. 8.7 Metastability-based PUF (Mathew et al. 2014)

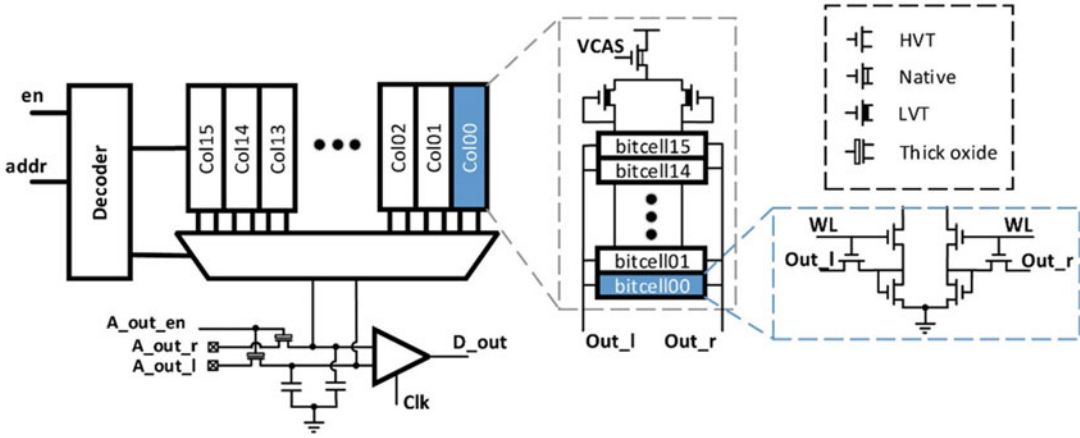

Fig. 8.8 PTAT-based PUF (Li et al. 2015)

architecture and principle of operation of the PTAT-based PUF. As seen in the figure, the PUF bitcell output is determined by the sign of the difference between *Out_l* and *Out_r*, both of which are independent of voltage and temperature. Aside from the high resiliency against voltage and temperature variation, another noteworthy feature of this PUF is its good area efficiency (only 727 F^2/bit, being F the minimum feature size of the adopted process), as enabled by the shared header per column.

A class of static mono-stable PUFs (Alvarez et al. 2015, 2016) for extremely low energy operation and low native instability rate was recently proposed, which relies on the amplification of random transistor mismatch through two complementary current mirrors. Figure 8.9 shows two implementations of the same general concept. Figure 8.9a shows the INV_PUF bitcell implementation of this concept, which comprises the cascode current mirrors M1–M4 and M5–M8. The two 1:1 current mirrors see the same current flowing through their respective input transistors (M3 and M5), and tend to mirror it to their output transistors (M4 and M6, respectively). Without mismatch, M4 and M6 would

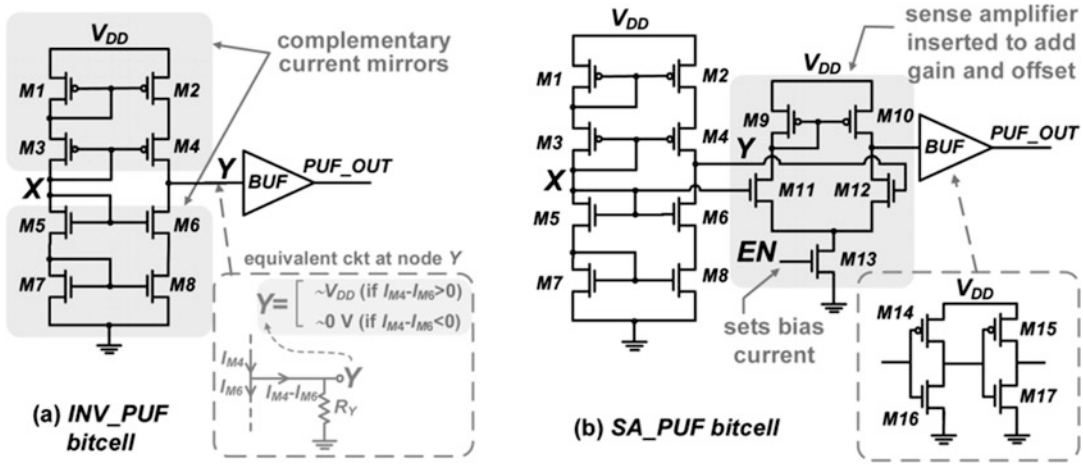

Fig. 8.9 Static mono-stable PUFs (Alvarez et al. 2016) (**a**) INV_PUF and (**b**) SA_PUF

conduct the same saturation current ($I_{M4,SAT}$ and $I_{M6,SAT}$), and node Y would assume the same voltage as node X (e.g., $V_{DD}/2$), due to the symmetry of the topology in Fig. 8.9a. However, random mismatch between M1–M2 and M7–M8 makes these currents unpredictably different. The large output small-signal impedance R_Y at node Y (Fig. 8.9a) translates the difference in such currents into a large voltage deviation. Accordingly, the voltage at node Y becomes essentially V_{DD} if $I_{M4,SAT}$- $I_{M6,SAT}>0$, or ground if $I_{M4,SAT}$- $I_{M6,SAT}<0$. Thus a digital output that is dominantly defined by the random mismatch between the two current mirrors is generated. In Fig. 8.9b, the alternative SA_PUF topology adds a sense amplifier (transistors M9–M13) after M1–M8 to further increase the voltage gain (and thus reduce the number of unstable bits) and introduce additional random mismatch through the sense amp offset.

Table 8.4 compares various PUFs in terms of the above mentioned metrics, with the best performing PUF for each metric being highlighted in bold. As can be seen from this table, most PUFs are typically affected by relatively poor randomness/statistical quality (see, e.g., bias) (Maes et al. 2012; Maes 2013; Yu et al. 2012) and up to 30% instability rate (Maes et al. 2012; Alvarez et al. 2015, 2016; Bhargava and Mai 2014).

A more complete list of fabricated PUFs can be found in the new public PUF database (Alioto and Alvarez 2016). Extracted trends in terms of native instability rate, area, and energy are shown in Fig. 8.10. From Fig. 8.10a, the metastability-based PUFs have the worst native instability rate, while the monostable PUFs exhibit the best native instability rate. The high native instability rate in metastability-based PUFs is reduced through post-processing and other stability enhancement techniques that increase testing time (i.e., cost). For the rest of the PUFs, the native instability rate has slightly increased over the years.

From Fig. 8.10b, the area per bit is highest for delay-based PUFs, due to the large number of stages required to (1) limit the oscillation frequency to acceptable values that can be distinguished by the subsequent circuitry, (2) to mitigate the instability rate of individual ring oscillators via k-sum or 1-out-of-k masking (Suh et al. 2007; Lee et al. 2004). In general, the area efficiency of PUF bitcells has improved over time, especially due to the adoption of more digital approaches that offer better density than analog ones. Analog PUF bitcells have an opposite trend, as their area tends to increase over time, when area is normalized to the square of the minimum feature size of the technology. This is mostly because of their analog nature, which

Table 8.4 Comparison of different PUFS

PUF	SRAM_PUF*	RO_PUF*	ICID (Lofstrom et al. 2000)	Arbiter (Lim et al. 2005)	Latch (Su et al. 2007)	Mathew et al. (2014)	PTAT (Li et al. 2015)	Yang et al. (2015)	INV_PUF (Alvarez et al. 2016)
Technology	65 nm	65 nm	0.35 µm	0.18 µm	0.13 µm	22 nm	65 nm	40 nm	65 nm
Stability (% native unstable bits at nominal condition)	16.66	18.16	1.3	9.8	3.04	30	7.1	12.5	2.34
V-T variation	0.6–1 V	0.4–0.5 V	1.1–5 V, −25 to 250 °C	1.8 V ±2%, 27–70° C	0.9–1.2 V	0.7–0.9 V		0.7–0.9 V	0.6–1 V, 25–85 °C
% error with VT variation	55.73	53.9	5	4.82	5.46875	30		12.5	5.72
Energy (pJ/bit)	1.1	0.4748	8333.3333	0.17125	0.93	0.19	1.1	17.75	0.015
Area (F^2/bit)	306	39,000	1708	708,403	4369	9628	727	2062	6000
Randomness (bias = probablity of 1)	0.6141	0.5023				0.4805	0.4928		0.5016
Uniqueness (mean inter-PUF FHD)	0.3321	0.4738	0.4911	0.3800	0.5055	0.5100	0.5001	0.5007	0.5014
Repeatability (mean intra-PUF FHD)	0.0602	0.0458	0.0134			0.0268	0.0057	0.0101	0.0034
Identifiability (inter-PUF/intra-PUF FHD)	6	10	37			19	88	50	149
NIST test						PASS	PASS	PASS	PASS
Entropy	0.9903	0.9947				0.9997	0.9998		0.9967
Autocorrelation function @ 95% confidence	0.0156	0.0884				0.01	0.0188	0.0283	0.0363

*Data taken from (Alvarez et al. 2016)

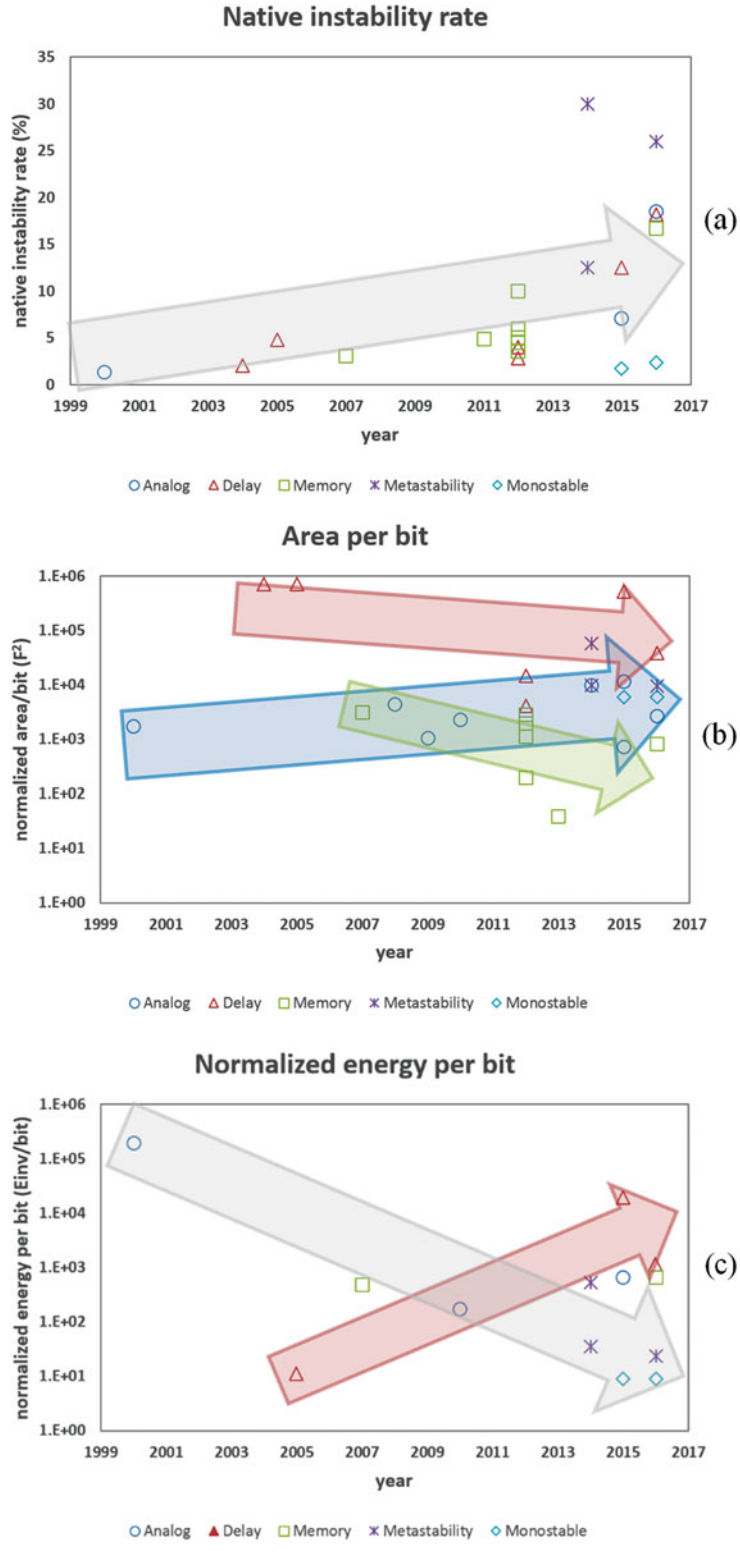

Fig. 8.10 Trend of (**a**) native instability rate, (**b**) area per bit (normalized to F^2, F being the minimum feature size of the CMOS process), (**c**) energy per bit for different PUFs implemented in custom PUF chips (Alioto and Alvarez 2016) (normalized to the energy E_{inv} consumed by a minimum-sized inverter in a single transition)

does not really enable shrinking with finer technologies.

From Fig. 8.10c, the energy per bit is improving, thanks to the adoption of more energy-aware PUFs. The circuit improvements in terms of energy definitely dominate the benefits of mere technology scaling. This is shown by Fig. 8.10c, which plots the energy normalized to the energy consumed by a minimum-sized inverter in the same technology, and hence represents a technology-independent metric. Interestingly, from Fig. 8.10c delay-based PUFs are an exception, as they tend to have larger energy per bit over the years. This is due to the need for a larger number of ring oscillators or oscillations to maintain acceptable stability, in spite of the progressively worse native stability in 8.10a.

Some prior work enables the capability to assure a well-defined stability safety margin at the output word level (Yu et al. 2012), as a form of robustness assurance against individual bit instability. Other prior work focuses on improving the stability of PUF bitcells without quantitative stability assurance at run-time. For example, introducing burn-in hardening in (Mathew et al. 2014) improves stability at the expense of significantly longer testing time, which conflicts with the very low cost requirement of IoT nodes (see Chap. 1). Another way to improve the statistical quality and suppress a limited number of unstable bits is through digital post-processing, at the expense of substantially larger silicon area and energy. The post-processing block can be a mixture of the following techniques:

- Error Correcting Code (ECC), which introduces a large area/energy overhead especially for high levels of targeted security, as its complexity grows exponentially in applications requiring wider PUF outputs; post-processing also leaks information and makes the PUF more vulnerable to physical attacks (Bhargava and Mai 2014). Various ECCs were used (Yu et al. 2012), such as 2D Hamming (Gassend et al. 2002), BCH (Suh et al. 2007; Mathew et al. 2016), two-stage ECC (Bösch et al. 2008), soft-decision ECC (Maes et al. 2009a, b), Index-Based Syndrome (Yu et al. 2011), Code-Offset Syndrome (Gassend et al. 2002; Yu and Devadas 2010; Suh et al. 2007; Dodis et al. 2008; Paral et al. 2011; Eiroa et al. 2012), pattern matching techniques (Paral et al. 2011), and fuzzy extractors (Selimis et al. 2011; Dodis et al. 2008)
- temporal majority voting across repeated PUF readings, which typically slow down and increase the energy per access by more than an order of magnitude (Mathew et al. 2014, 2016; Li et al. 2015)
- on-the-fly PUF bitcell masking (Satpathy et al. 2014), and PUF redundancy (Yang et al. 2015; Suh et al. 2007), which skips the bitcells that are found to be unstable at testing time by storing the bit error map in an additional volatile memory array (Mathew et al. 2014; Bhargava and Mai 2014; Karpinskyy et al. 2016); this approach may introduce significant area/energy overhead, and considerably widens the opportunities to perform successful invasive attacks (e.g., interfering with PUF operation by writing on the additional memory).

Figure 8.11 shows an example where ECC is used to improve the reliability of the PUF (Gassend et al. 2002). In this implementation, redundant information is generated for each challenge-response pair, to allow the correction of the PUF output. The ECC overhead is ~14 kgates, which is about an order of magnitude bigger than the PUF array itself. Similarly, in (Rahman et al. 2014), ECC encoder was shown to have an area of ~3–12 kgates, with the ECC decoder requiring an even larger area of ~20–-75 kgates. Detection of instability was proposed in (Karpinskyy et al. 2016) during the PUF response generation, and in (Satpathy et al. 2014) at boot time, as depicted in Fig. 8.12.

8.4 PUF Vulnerability Analysis

Existing PUF solutions suffer from various limitations that have limited their adoption in

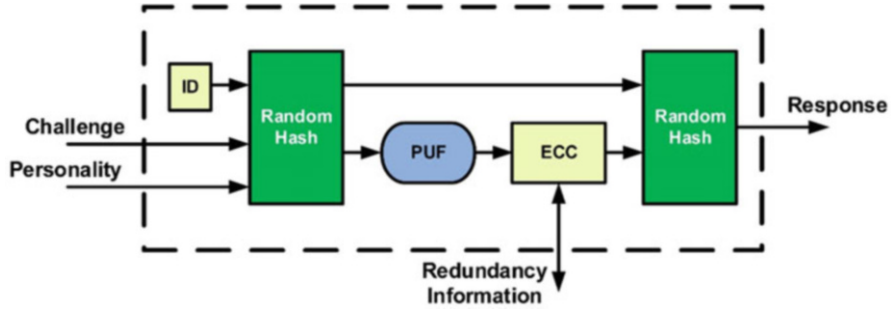

Fig. 8.11 Block diagram of an improved PUF that utilizes ECC to improve the PUF reliability (Gassend et al. 2002)

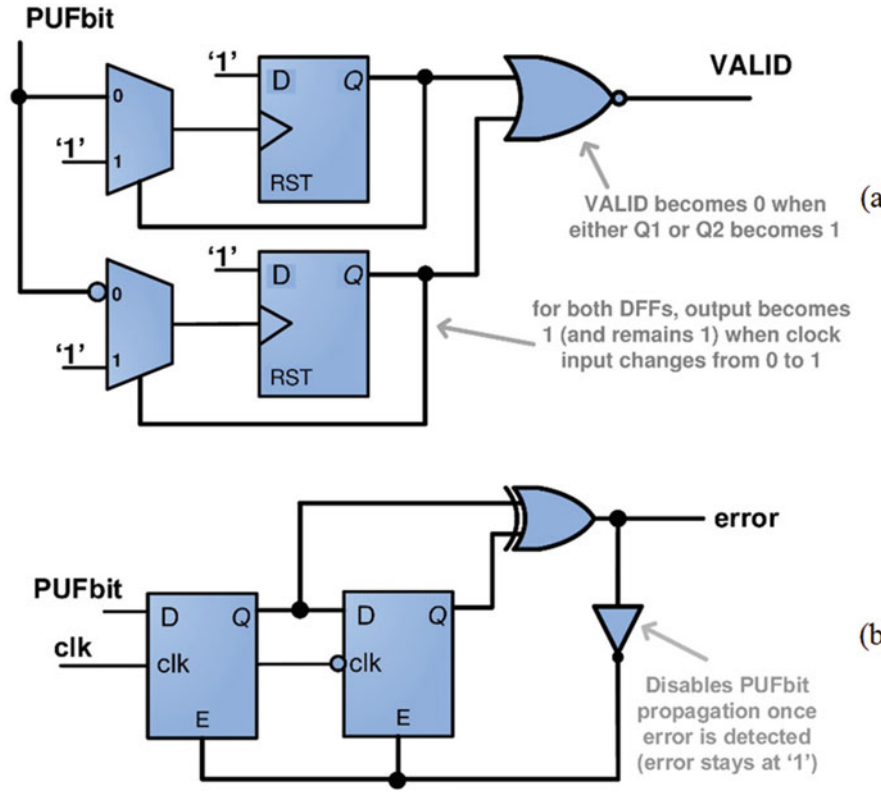

Fig. 8.12 Possible circuits for runtime error detection: (**a**) glitch detector from (Karpinskyy et al. 2016); (**b**) dark bit masking from (Satpathy et al. 2014)

real products. Indeed, to date there are only a few PUFs available in the market, such as (ICTK, Co. Ltd. 2014; Intrinsic-ID 2016; Invia PUF 2016; QuantumTrace 2013; Verayo Inc. 2013). For example, delay- and metastability-based PUFs are very sensitive to voltage/temperature variations, aging and noise (Maes et al. 2012), and are very hard to verify at design time in terms of output statistics and randomness. Hence, they typically require multiple silicon runs to reliably assess a given design. Glitch based PUFs are not yet mature, and are well known to be rather unstable and complex. Leakage-based PUFs are sensitive to environmental variations and need extra circuitry for voltage and current biasing, which are prohibitively costly in terms of area,

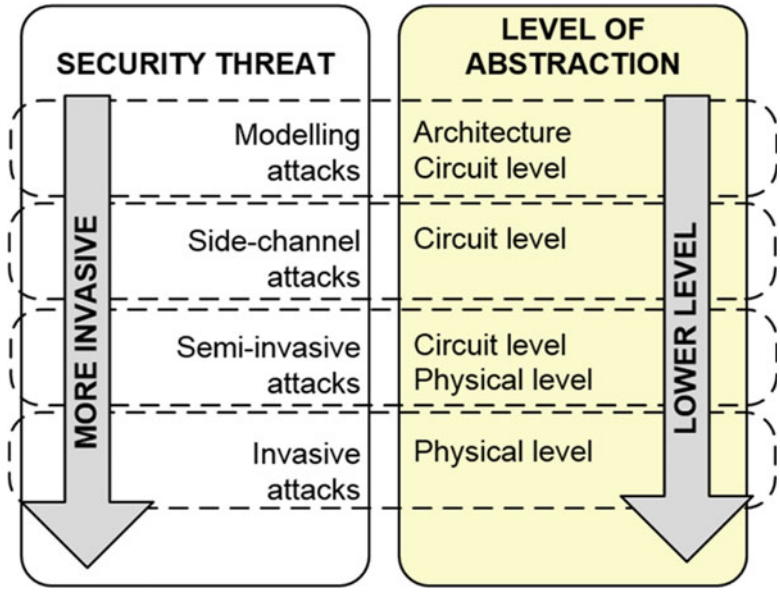

Fig. 8.13 Attacks to PUFs versus level of invasiveness and level of abstraction

energy and design effort. Memory-based PUFs are strongly technology dependent and hence not technology-portable, and suffer from bit flipping and data remanence (Eiroa et al. 2012). DRAM error maps are well known to have obvious correlation between different responses, thus drastically weakening the unpredictability of responses (Rosenblatt et al. 2013). Both delay- and supply network-based PUFs are very vulnerable to modeling attacks (Rührmair et al. 2013).

The trustworthiness of a PUF is defined by its resistance to attacks that aim to impersonate, replicate or recover portions of the PUF bits. These attacks can be active (injecting fault into the design) or passive (simply observing). They can also be classified as invasive (meaning requiring depackaging the chip to see the design or probe internal signals) or non-invasive. The most representative attacks to PUFs are summarized in Fig. 8.13, from the least to the most invasive. Modeling attacks are passive non-invasive, and involve only the observation of transmitted information and successive trials to impersonate the device by leveraging the small search PUF key space or poor randomness, or by recording and reusing previous CRPs (Sadeghi and Naccache 2010) (e.g., man-in-the-middle attacks). Delay-based PUFs are prone to these types of attacks. In (Rührmair et al. 2013), for example, they were able to show more than 95% prediction rate using machine learning to model the arbiter and ring oscillator PUFs.

For identification purposes, a "strong PUF" with large number of CRPs is clearly needed to limit the effectiveness of man-in-the-middle cryptanalytic attacks, but unfortunately all practically viable strong PUFs are very vulnerable to modeling attacks (Sadeghi and Naccache 2010; Helinski et al. 2009), and hence unsuitable for moderate-to-high levels of security.

Side-channel attacks aim to identify the PUF key employed by the chip through its correlation with the measured power consumption (e.g., DPA (Kocher et al. 1999), correlation attacks (Brier et al. 2004) and Leakage Power Analysis (Alioto et al. 2014, 2010a; Giorgetti et al. 2007) or electromagnetic emissions (Mangard et al. 2007)). These attacks are performed on the execution of the cryptographic algorithm that uses the key that the attacker is trying to retrieve. Differential Power Analysis (DPA) was proposed in 1998 (Kocher et al. 1999). Figure 8.14 shows a sample trace of the whole DES operation, where the 16 repetitive pulses correspond to

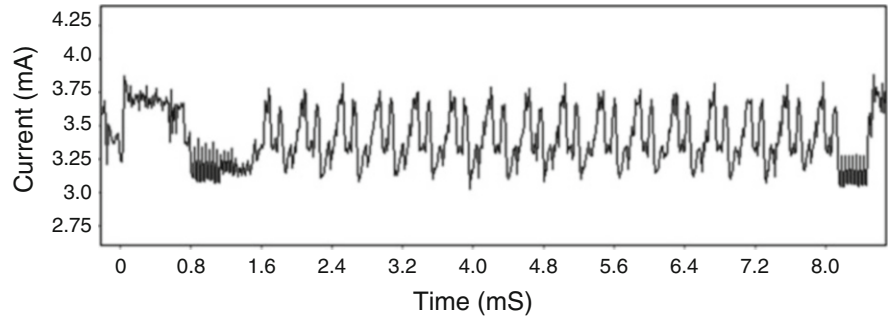

Fig. 8.14 SPA traces showing 16 rounds of DES

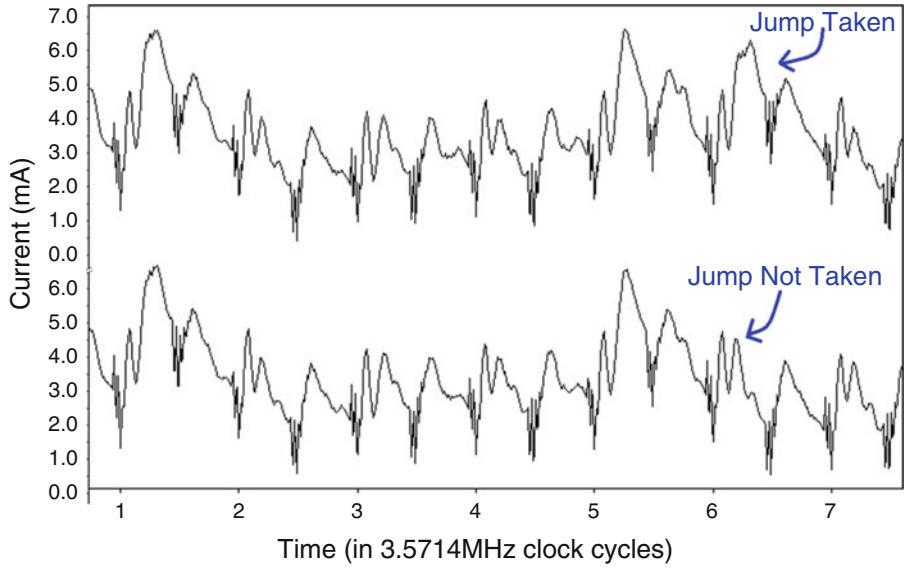

Fig. 8.15 SPA single round trace showing difference in power consumption

the 16 rounds of DES. Measurements are performed during encryption (or decryption) under different plaintexts, in order to see the difference in power consumptions in the different stages of the algorithm. A closer view of the same pulse in Fig. 8.14 could also reveal more about the data, as shown in Fig. 8.15. From the figure, the difference in power consumption in cycle 6 is due to a jump instruction (where jump is taken in the top plot and not taken in the bottom plot). The idea of DPA is to retrieve the key of the cipher using divide and conquer approach by guessing portions of the key, thereby breaking the exponential complexity of deciphering the key and reducing the number of required trials. For each trial, the cipher's power consumption estimated under two key bit guesses is compared, or correlated to the actual power. The difference of two traces (see, e.g., Fig. 8.15) has a spike in cycle 6 when such key bit is used in the algorithm execution, and is almost zero elsewhere. Similar procedure is applied to specific operations in the algorithm in order to help identify whether the initially assumed key is correct or not. A power model for DPA attacks on symmetric-key cryptographic algorithms implemented using static logic was proposed in (Alioto et al. 2010b). The model predicts the effectiveness of DPA attacks and the conditions for which the circuit becomes vulnerable to these attacks.

As DPA attacks target the cryptographic core rather than the PUF itself, one way to prevent them is to mask the power consumption of different operations. This can be done through a change in the algorithm, masking data (Canright et al. 2008), or resorting to codeword encoding (Merli et al. 2013), among others. The use of different logic styles, such as the sense amplifier based logic (SABL) (Tiri et al. 2002) and dual-rail pre-charge (DRP) circuits, such as wave differential dynamic logic (WDDL) (Tiri et al. 2004), masked dual-rail pre-charge logic (MDPL) (Popp et al. 2005) and dual-rail random switching logic (DRSL) (Chen et al. 2006), was shown to be effective in masking operations by maintaining almost the same power consumption regardless of the operation and processed data (Schaumont et al. 2007; Monteiro et al. 2011).

Semi-invasive attacks (e.g., fault attacks) aim to interfere with the circuit operation by introducing glitches and injecting faults that expose data that would otherwise be securely processed internally. Fault injection and timing attacks can be avoided by consistency checking, at the expense of additional runtime and/or area. Laser scanning was used in (Holcomb et al. 2009) and (Nedospasov et al. 2013) to read out the state values of memory elements.

Invasive attacks aim to physically observe (e.g., reverse engineering) or modify the chip physical implementation (e.g., probing, fibbing), and can be counteracted through secure coprocessors (Smith and Weingart 1999). This solution, however, is very expensive both in terms of area and energy, as the coprocessor has to be powered at all times. The work in (Wan et al. 2015) proposes to counteract invasive attack by laying out metal wires on top of transmission lines to switch capacitor circuitry. By doing this, the capacitance of the sampling capacitor changes during invasive attacks, making the PUF output invalid.

The targeted level of trustworthiness defines an adequate set of attacks that need to be counteracted in a PUF design, although to date only individual and fragmented techniques have been proposed. As research challenge in the area of PUFs, a comprehensive set of techniques would be needed to meet a given level of trustworthiness, with each being allocated to the appropriate level of abstraction to meet a security level target at low cost.

8.5 Novel Concept of PUF-Enhanced Cryptography, Trends and Perspectives

Despite the recent and broad interest in PUFs, they have made a very limited impact on real applications to date due to several challenges that seriously hinder PUF trustworthiness, as above mentioned and summarized below:

- PUF responses can be very unstable (i.e., not repeatable), thus requiring a large cost in terms of testing time (post-silicon masking, PUF hardening), area or energy overhead (to suppress unstable bits at design time, and to include expensive post-processing blocks). Additional post-processing circuits also make PUF more vulnerable to physical attacks (e.g., side-channel, probing), as they become part of the PUF itself, and hence introduce additional backdoors to the PUF
- there is no available methodology to systematically assure a level of security (e.g., bit randomness, stability) at design/verification time. This prohibits the provability of PUF trustworthiness at design time, requires repeated silicon runs to converge to a targeted and provable security level, and drastically prolongs the design cycle and time to market
- existing solutions only target a specific type of security threat and level of abstraction (mostly circuit level). Hence, they cannot address the security challenges posed by different types of attacks, and do not introduce across-level solutions (e.g., from PUF layout to architecture and software)
- the PUF design is essentially a manual design process, which prohibits design automation and technology portability, and entails very low design productivity
- specifically for IoT nodes, constant node-to-node communications and data transmission

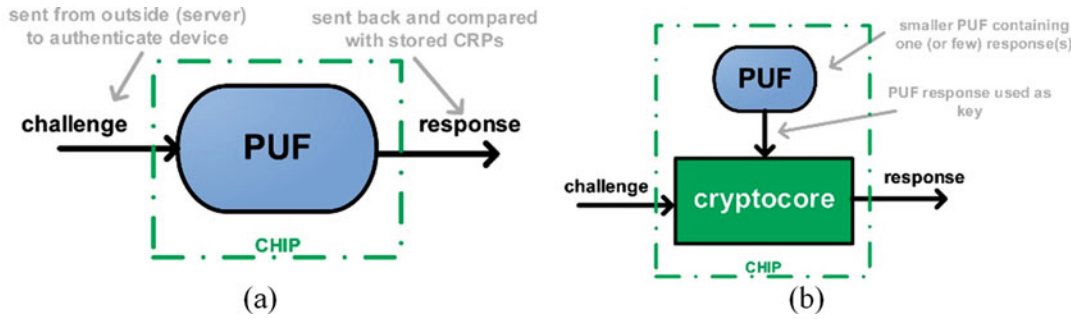

Fig. 8.16 (**a**) PUF as key generator, (**b**) crypto-core-based strong PUF

requires regular authentication. This translates into and unacceptably large number of CRPs and therefore PUF capacity (see Table 8.1). Public-key cryptography to establish trust cannot mitigate such problem, due to its prohibitive area and energy cost in IoT nodes.

So far, we have treated PUFs as secure random key generators for chip identification through conventional challenge-response pairs, as illustrated in Fig. 8.16a. A more promising approach implements a strong PUF through a crypto-core (e.g., AES) using the PUF key as encryption key, and then treating input/output values as CRPs (Bhargava and Mai 2014), as illustrated in Fig. 8.16b. In (Bhargava and Mai 2014), the introduction of AES increased the area by 4.6× compared to the PUF array, but increased the number of available CRPs exponentially. By using an AES design that is designed specifically for IoT applications (Zhao et al. 2015), the power consumption of AES is well below 1 μW and the area cost is reduced by 3×, thus becoming very affordable for the same exponential increase in CRPs.

For IoT node-to-node communications, the concept of combining a PUF with a crypto-core can also be used to reduce the circuit complexity and energy required for continuous authentication, thereby reducing the required PUF capacity at a given level of security. Conventional node-to-node communication is illustrated in Fig. 8.17, where CRPs are used to authenticate both nodes each time data is transferred between them.

Instead, a more efficient security scheme is introduced in Fig. 8.18. In this "PUF-enabled node-to-node communication" scheme, secure PUF key exchange is enabled at the authentication phase through cryptography. After one-time authentication, both nodes can communicate with each other securely through encryption and decryption using the exchanged keys, and without server assistance (therefore not needing a large CRP database). This makes communication over complex networks scalable, as the database is involved only at the first communication between nodes. As can be seen in the figure, node-to-node communication is simplified through the joint use of PUF and cryptography, which permit to securely exchange keys over an insecure channel, and avoiding the very energy- and area-hungry public-key cryptography. Such interesting and synergistic use of PUFs and cryptography is here introduced and named "PUF-enhanced cryptography".

Another interesting ramification of PUF-enhanced cryptography is the ability to substantially strengthen the security of a crypto-core against cryptanalytic attacks, by appropriately embedding a PUF into it. As illustrated in Fig. 8.19, PUF-enhanced cryptography goes beyond the traditional scheme of securely storing a single crypto-key, and permits to extend the crypto-key compared to the size imposed by the crypto-algorithm, thus making it stronger against cryptoanalytic attacks. Traditionally, key extension is not possible since its length is dictated by the encryption standard. However, in PUF-enhanced cryptography, a PUF with capacity

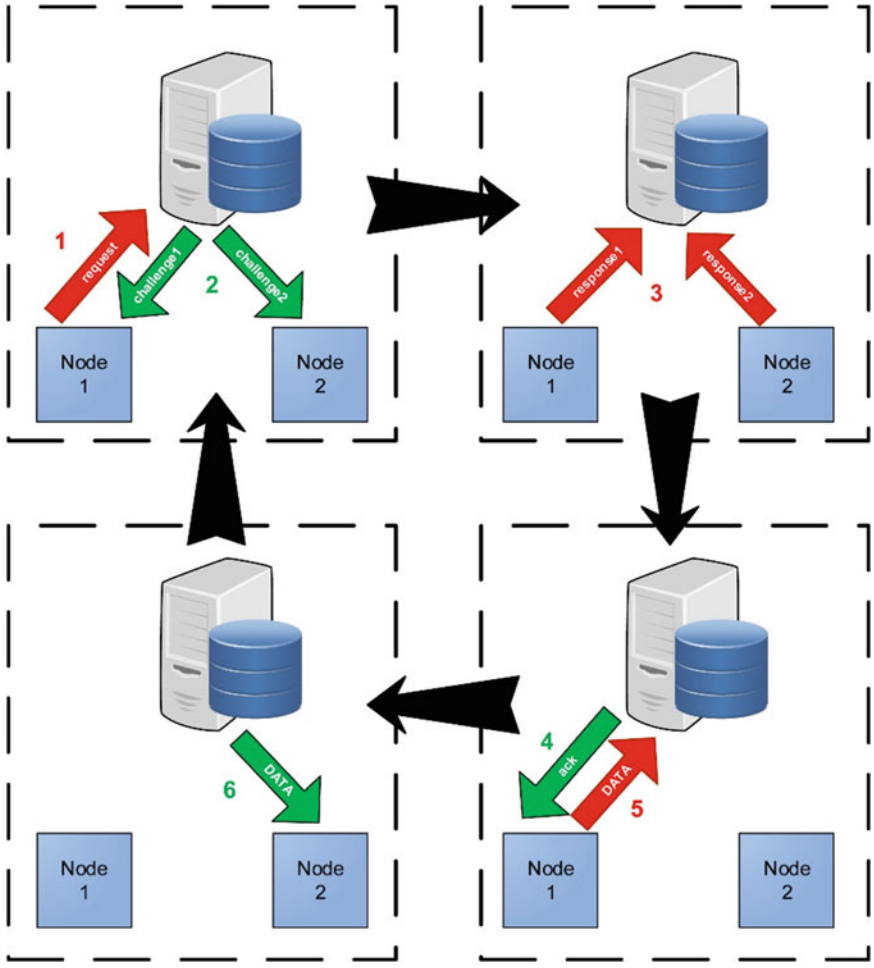

Fig. 8.17 Conventional node-to-node data transfer through server, which needs to constantly assist the two nodes during their communications

larger than the key is used to generate repeatable but unpredictable new keys that are combined with the conventional user key to generate the fixed length enhanced key used by the on-chip crypto-core. To this aim, the key enhancer in Fig. 8.19 is introduced to dynamically concatenate the user and PUF keys, and then compress them into the pre-defined length. Although the key enhancer in Fig. 8.19 is shown to be outside the crypto-core (i.e., without interfering with conventional operation), it can also extend to the inside of the latter, and operate across several blocks of plaintext. The encryption sequence is initialized by the user key, and then managed by a key enhancer. The key enhancer can likewise be a simple finite state machine, which generates time-varying challenges to a PUF, or a lightweight cipher itself (Shiozaki et al. 2015). As a result, as opposed to the traditional scheme that uses a single private key, the PUF-enhanced cryptography scheme in Fig. 8.19 actually uses a larger set of keys, whose number is basically limited by the desired PUF capacity.

From an attacker point of view, guessing the private crypto-key of a typical cryptography system requires an effort that is (exponentially) defined by the size of the single key size. Instead, in the PUF-enhanced cryptography scheme in Fig. 8.19, the search space for the crypto-key is enlarged by the capacity of the PUF, thus easily making the key search unfeasible even under

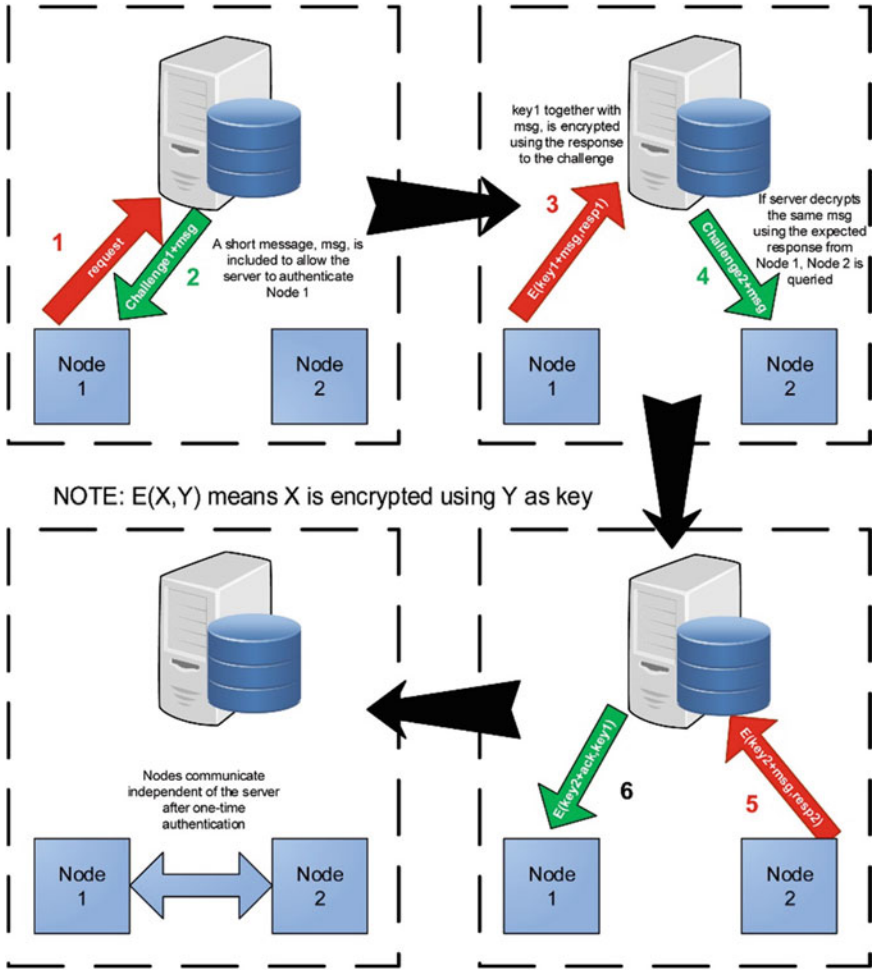

Fig. 8.18 PUF-enabled key exchange and node-to-node communication

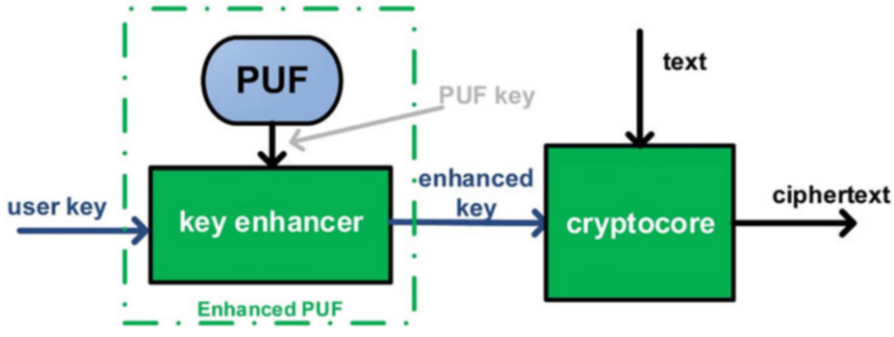

Fig. 8.19 The new concept of PUF-enhanced cryptography (key is continuously enriched with PUF key).

very powerful equipment and computing resources. In practical cases, the PUF-enhanced cryptography permits to drastically strengthen the security of an existing algorithm with (1) limited area cost, thanks to the exponential increase of the size of the key search space, under PUF

capacity extension, and (2) no throughput penalty, since the generation of the PUF output is generally much faster than encryption. When using PUFs like in (Alvarez et al. 2016), the latter property is enabled by the intrinsically high speed of the PUF architecture, since PUF bits are always available at the output and only need to be routed to the circuitry that consumes them.

The above mentioned dynamic change of the key over time is a tool to improve the strength of PUF-enhanced cryptography against crypto-analytic attacks. In the case of IoT devices relying on energy harvesting, changing keys becomes a necessity as dictated by the availability of supply. For example, in (Aysu and Schaumont 2016) key generation is divided into several phases and precomputation is done whenever supply available, and intermediate results are stored, for use in the next phase.

In summary, PUF-enhanced cryptography permits to drastically enhance the security of a crypto-core by leveraging its synergy with a PUF, to generate time-varying crypto-keys instead of having a fixed one. In addition, the adoption of such PUF to enhance the crypto-algorithm also permits to easily scale up the level of security on demand. Indeed, the level of security defines the number of PUF words that are needed, and hence it only affects the periodicity of the key enhancer for a given PUF capacity. Also, the PUF unambiguously authenticates the die that the crypto-core runs on. In addition, the addition of a PUF to a crypto-core generally entails a very small energy overhead, as the energy per bit of a PUF is typically two to three orders of magnitude smaller than a crypto-core. Very similar considerations hold for the area efficiency. These features are particularly interesting in the context of the Internet of Things, as they make crypto-algorithms and crypto-cores affordable in terms of area and energy, thus enabling continuous and ubiquitous security. When a much higher level of security is occasionally needed, the PUF enhancement permits to further scale it up at a very low area/energy cost.

Acknowledgement The authors acknowledge the kind support by the MOE2016-T2-1-150 grant from the Singaporean Ministry of Education.

References

M. Alioto, L. Giancane, G. Scotti, A. Trifiletti, Leakage power analysis attacks: a novel class of attacks to nanometer cryptographic circuits. IEEE Trans. Circuits Syst. I **57**(2), 355–367 (2010a)

M. Alioto, M. Poli, S. Rocchi, Differential power analysis attacks to precharged buses: a general analysis for symmetric-key cryptographic algorithms. IEEE Trans. Dependable Secur. Comput. **7**(3), 226–239 (2010b)

M. Alioto, S. Bongiovanni, M. Djukanovic, G. Scotti, A. Trifiletti, Effectiveness of leakage power analysis attacks on DPA-resistant logic styles under process variations. IEEE Trans. Circuits Syst. I Regul. Pap. **61**(2), 429–442 (2014)

A. Alvarez, W. Zhao, M. Alioto, 15 fJ/bit static physically unclonable functions for secure chip identification with <2% native bit instability and 140x inter/intra puf hamming distance separation in 65 nm. IEEE Int. Solid-State Circuits Conf. **5**, 256–258 (2015)

A.B. Alvarez, W. Zhao, M. Alioto, Static physically unclonable functions for secure chip identification with 1.9–5.8% native bit instability at 0.6–1 V and 15fJ/bit in 65 nm. IEEE J. Solid State Circuits **60**(5), 1–4 (2016)

A. Aysu, P. Schaumont, Precomputation methods for hash-based signatures on energy-harvesting platforms. IEEE Trans. Comput. **65**(9), 2925–2931 (2016)

M. Bhargava, K. Mai, An efficient reliable PUF-based cryptographic key generator in 65 nm CMOS. Design Autom. Test Europe Conf. Exhibition **1**, 1–6 (2014)

M. Bhargava, C. Cakir, K. Mai, Attack resistant sense amplifier based PUFs (SA-PUF) with deterministic and controllable reliability of PUF responses, in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (2010) pp. 106–111

C. Bösch, J. Guajardo, A.R. Sadeghi, J. Shokrollahi, P. Tuyls, Efficient helper data key extractor on FPGAs. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) **5154 LNCS**, 181–197 (2008)

E. Brier, C. Clavier, F. Olivier, Correlation power analysis with a leakage model, in *Cryptographic Hardware and Embedded Systems* (2004), pp. 16–29

D. Canright, L. Batina, A very compact 'perfectly masked' S-box for AES, in *Lecture Notes in Computer Science* (2008), pp. 446–459

Z. Chen, Y. Zhou, Dual-rail random switching logic: a countermeasure to reduce side channel leakage, in *Cryptographic Hardware and Embedded Systems (CHES)* (2006), pp. 242–254

B.D. Choi, T.W. Kim, D.K. Kim, Zero bit error rate ID generation circuit using via formation probability in 0.18 μm CMOS process. IET J. Mag. **50**(12), 876–877 (2014)

Y. Dodis, R. Ostrovsky, L. Reyzin, A. Smith, Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)

S. Eiroa, J. Castro, M. Martínez-Rodríguez, E. Tena, P. Brox, I. Baturone, Reducing bit flipping problems in SRAM physical unclonable functions for chip identification, in *IEEE International Conference on Electronics, Circuits, and Systems (ICECS)* (2012), pp. 392–395

D. Ganta, V. Vivekraja, K. Priya, L. Nazhandali, A highly stable leakage-based silicon physical unclonable functions, in *International Conference on VLSI Design* (2011), pp. 135–140

B. Gassend, D. Clarke, M. van Dijk, S. Devadas, Silicon physical random functions, in *ACM Conference on Computer and Communications Security (CCS)* (2002), p. 148

J. Giorgetti, G. Scotti, A. Simonetti, A. Trifiletti, Analysis of data dependence of leakage current in CMOS cryptographic hardware, in *Great Lakes Symposium on VLSI (GLSVLSI)* (2007), pp. 78–83

J. Guajardo, S.S. Kumar, G. Schrijen, P. Tuyls, FPGA intrinsic PUFs and their use for IP protection, in *Lecture Notes in Computer Science*, ed. by P. Paillier, I. Verbauwhede (Springer, Heidelberg, 2007), pp. 63–80

R. Helinski, D. Acharyya, J. Plusquellic, A physical unclonable function defined using power distribution system equivalent resistance variations, in *ACM/IEEE Design Automation Conference* (2009), pp. 676–681

D.E. Holcomb, W.P. Burleson, K. Fu, Power-up SRAM state as an identifying fingerprint and source of true random numbers. IEEE Trans. Comput. **58**(9), 1198–1210 (2009)

ICTK, Co. Ltd. (2014), http://www.ictk.com/servicenproduct/puf

Intrinsic-ID, SRAM PUF: the secure silicon fingerprint, in *White Paper* (2016)

Invia PUF IP (2016), http://invia.fr/infrastructure/physical-unclonable-function-PUF.aspx

B. Karpinskyy, Y. Lee, Y. Choi, Y. Kim, M. Noh, S. Lee, Physically unclonable function for secure key generation with a key error rate of 2E-38 in 45 nm smart-card chips, in *IEEE International Solid-State Circuits Conference (ISSC)* (2016), pp. 158–160

P. Kocher, J. Ja, B. Jun, Differential power analysis. Lect. Notes Comput. Sci. **1666**, 388–397 (1999)

O. Kömmerling, M.G. Kuhn, Design principles for tamper-resistant smartcard processors, in *USENIX Workshop on Smartcard Technology* (1999), pp. 9–20

S.S. Kumar, J. Guajardo, R. Maes, G. Schrijen, P. Tuyls, The butterfly PUF protecting IP on every FPGA, in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)* (2008), no. 71369, pp. 67–70

J.W. Lee, B. Gassend, G.E. Suh, M. van Dijk, S. Devadas, A technique to build a secret key in integrated circuits for identification and authentication applications, in *Symposium on VLSI Circuits* (2004), pp. 176–179

J. Li, M. Seok, A 3.07 μm^2/bitcell physically unclonable function with 3.5% and 1% bit-instability across 0 to 80 °C and 0.6 to 1.2 V in a 65 nm CMOS, in *IEEE Symposium on VLSI Circuits, Digest of Technical Papers* (2015), pp. 250–251

D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. Van Dijk, S. Devadas, Extracting secret keys from integrated circuits. IEEE Trans. Very Large Scale Integr. Syst. **13**(10), 1200–1205 (2005)

N. Liu, S. Hanson, D. Sylvester, D. Blaauw, OxID: on-chip one-time random ID generation using oxide breakdown, in *Symposium on VLSI Circuits* (2010), pp. 231–232

K. Lofstrom, W.R. Daasch, D. Taylor, IC identification circuit using device mismatch. IEEE Int. Solid-State Circuits Conf. **46**(8), 1999–2000 (2000)

M. Alioto, A. Alvarez, Physically Unclonable Function database (2016), http://www.green-ic.org/pufdb

R. Maes, *Physically Unclonable Functions: Construction, Properties and Applications* (Springer, London, 2013)

R. Maes, Physically unclonable functions : constructions, properties and applications. Katholieke Universiteit Leuven (2012)

R. Maes, P. Tuyls, I. Verbauwhede, Intrinsic PUFs from flip-flops on reconfigurable devices, in *Workshop on Information and System Security* (2008), no. 71369, pp. 1–17

R. Maes, P. Tuyls, I. Verbauwhede, A soft decision helper data algorithm for SRAM PUFs, in *IEEE International Symposium on Information Theory* (2009), pp. 2101–2105

R. Maes, P. Tuyls, I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs, in *Cryptographic Hardware and Embedded Systems (CHES)* (2009), pp. 1–15

R. Maes, V. Rozic, I. Verbauwhede, P. Koeberl, E. van der Sluis V. can der Leest, Experimental evaluation of physically unclonable functions in 65 nm CMOS, in *European Solid State Circuit Conference (ESSCIRC)* (2012), pp. 486–489

S. Mangard, E. Oswald, T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards* (Springer, New York, 2007)

S.K. Mathew, S.K. Satpathy, M.A. Anders, H. Kaul, S.K. Hsu, A. Agarwal, G.K. Chen, R.J. Parker, R.K. Krishnamurthy, V. De, A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22 nm CMOS. Digest Tech. Pap. - IEEE Int. Solid-State Circuits Conf. **2**(c), 278–280 (2014)

S. Mathew, S. Satpathy, V. Suresh, M. Anders, H. Kaul, A. Agarwal, S. Hsu, G. Chen, R. Krishnamurthy, V. De, A 4fJ/bit delay-hardened Physically unclonable

function circuit with selective bit destabilization in 14 nm trti-gate CMOS, in *Symposium on VLSI Circuits* (2016), pp. 248–249
D. Merli, F. Stumpf, G. Sigl, Protecting PUF error correction by codeword masking (2013), pp. 1–16
C. Monteiro, Y. Takahashi, T. Sekine, "Resistance against power analysis attacks on adiabatic dynamic and adiabatic differential logics for smart cards, in *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (2011), pp. 1–5
D. Nedospasov, J.P. Seifert, C. Helfmeier, C. Boit, Invasive PUF analysis, in *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)* (2013), pp. 30–38
R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, Physical one-way functions. Science **297**, 2026–2030 (2002)
Z.S. Paral, S. Devadas, Reliable and efficient PUF-based key generation using pattern matching, in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (2011), no. 978, pp. 128–133
T. Popp, S. Mangard, Masked dual-rail pre-charge logic: DPA-resistance without routing constraints, in *Cryptographic Hardware and Embedded Systems (CHES)* (2005), pp. 172–186
D. Puntin, S. Stanzione, G. Iannaccone, CMOS unclonable system for secure authentication based on device variability, in *European Solid State Circuit Conference (ESSCIRC)* (2008), pp. 130–133
QuantumTrace, LLC PUF IP Product (2013), http://www.quantumtrace.com/Products/IP/PUF%20IP/
M.T. Rahman, D. Forte, J. Fahrny, M. Tehranipoor, ARO-PUF: an aging-resistant ring oscillator PUF design, in *Design, Automation & Test in Europe Conference & Exhibition (DATE)* (2014), pp. 1–6
S. Rosenblatt, D. Fainstein, A. Cestero, J. Safran, N. Robson, T. Kirihata, S.S. Iyer, Field tolerant dynamic intrinsic chip ID using 32 nm high-K/metal gate SOI embedded DRAM. IEEE J. Solid State Circuits **48**(4), 940–947 (2013)
D. Roy, J.H. Klootwijk, N.A.M. Verhaegh, H.H.A.J. Roosen, R.A.M. Wolters, Comb capacitor structures for on-chip physical uncloneable function. IEEE Trans. Semicond. Manuf. **22**(1), 96–102 (2009)
U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. ürgen Schmidhuber, Modeling attacks on physical unclonable functions, in *Proceedings of ACM Conference on Computer and Communications Security* (2010), pp. 237–249
U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, S. Devadas, PUF modeling attacks on simulated and silicon data. IEEE Trans. Inf. Forensics Secur. **8**(11), 1876–1891 (2013)
A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, A statistical test suite for random and pseudorandom number generators for cryptographic applications. Natl. Inst. Stand. Technol. **800–22**(Rev 1a), 131 (2010)
A.-R. Sadeghi, D. Naccache (eds.), *Towards Hardware-Intrinsic Security: Foundations and Practice* (Springer, Berlin, 2010)
D. Samyde, S. Skorobogatov, R. Anderson, J.-J. Quisquater, On a new way to read data from memory, in *International IEEE Security in Storage Workshop* (2002), pp. 65–69
S. Satpathy, S. Mathew, J. Li, P. Koeberl, M. Anders, H. Kaul, G. Chen, A. Agarwal, S. Hsu, R. Krishnamurthy, 13fJ/bit probing-resilient 250 K PUF array with soft dark-bit masking for 1.94% bit-error in 22 nm tri-gate CMOS," in *European Solid State Circuit Conference (ESSCIRC)* (2014), pp. 239–242
P. Schaumont, K. Tiri, Masking and dual-rail logic don't add up, in *Cryptographic Hardware and Embedded Systems (CHES)* (2007), pp. 95–106
G.-J. Schrijen, V. Van Der Leest, Comparative analysis of SRAM memories used as PUF primitives, in *Design, Automation & Test in Europe Conference & Exhibition (DATE)* (2012), pp. 1319–1324
G. Selimis, M. Konijnenburg, M. Ashouei, J. Huisken, H. De Groot, V. Van Der Leest, G.J. Schrijen, M. Van Hulst, P. Tuyls, "Evaluation of 90nm 6T-SRAM as physical unclonable function for secure key generation in wireless sensor nodes, *Proceedings of IEEE International Symposium on Circuits Systems* (2011), pp. 567–570
M. Shiozaki, T. Kubota, T. Nakai, A. Takeuchi, T. Nishimura, T. Fujino, Tamper-resistant authentication system with side-channel attack resistant AES and PUF using MDR-ROM. in *IEEE International Symposium on Circuits and Systems (ISCAS)* (2015), pp. 1462–1465
P. Simons, E. Van Der Sluis, V. Van Der Leest, Buskeeper PUFs, a promising alternative to D Flip-Flop PUFs, in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (2012), pp. 7–12
S.W. Smith, S. Weingart, Building a high-performance, programmable secure coprocessor. Comput. Networks **31**(8), 831–860 (1999)
S. Stanzione, G. Iannaccone, Silicon physical unclonable function resistant to a 10^25-trial brute force attack in 90 nm CMOS, in *Symposium on VLSI Circuits* (2009), pp. 116–117
Y. Su, J. Holleman B. Otis, A 1.6pJ/bit 96% stable chip-ID generating circuit using process variations, in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference (ISSCC)* (2007), pp. 406–408
G.E. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in *ACM/IEEE Design Automation Conference* (2007), pp. 9–14

G.E. Suh, C.W. O'Donnell, S. Devadas, Aegis: a single-chip secure processor. IEEE Des. Test Comput. **24**(6), 570–580 (2007b)

D. Suzuki, K. Shimizu, The glitch PUF: a new delay-PUF architecture exploiting glitch shapes, in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)* (2010), pp. 366–382

K. Tiri, M. Akmal, I. Verbauwhede, A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards, in *European Solid-State Circuits Conference (ESSCIRC)* (2002), pp. 403–406

K. Tiri, I. Verbauwhede, A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation, in *Design, Automation & Test in Europe Conference & Exhibition (DATE)* (2004), pp. 246–251

P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, R. Wolters, Read-proof hardware from protective coatings, in *Cryptographic Hardware and Embedded Systems (CHES)* (2006), pp. 369–383

Verayo Inc. (2013), http://www.verayo.com/tech.php

M. Wan, Z. He, S. Han, K. Dai, X. Zou, An invasive-attack-resistant PUF based on switched-capacitor circuit. IEEE Trans. Circuits Syst. I **62**(8), 2024–2034 (2015)

T. Xu, J.B. Wendt, M. Potkonjak, Matched digital PUFs for low power security in implantable medical devices, *2014 I.E. International Conference on Healthcare Informatics* (2014), pp. 33–38

K. Yang, Q. Dong, D. Blaauw, D. Sylvester, A physically unclonable function with BER < 10^-8 for robust chip authentication using oscillator collapse in 40 nm CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (2015), pp. 254–256

M.M. Yu, S. Devadas, Secure and robust error correction for physical unclonable functions. IEEE Des. Test Comput. **27**(1), 48–65 (2010)

M.M. Yu, D.M. Raihi, R. Sowell, S. Devadas, Lightweight and secure PUF key storage using limits of machine learning, in *Workshop on Cryptographic Hardware and Embedded Systems* (2011), pp. 358–373

M.M. Yu, R. Sowell, A. Singh, D.M. Raihi, S. Devadas, Performance metrics and empirical results of a PUF cryptographic key generation ASIC, in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (2012), pp. 108–115

W. Zhao, Y. Ha, M. Alioto, Novel self-body-biasing and statistical design for near-threshold circuits with ultra energy-efficient AES as case study. IEEE Trans. VLSI Systems **23**(8), 1390–1401 (2015)