Massimo Alioto

Trends in Hardware Security

From basics to ASICs

This article presents an excerpt of the tutorial on hardware security delivered at the 2019 IEEE International Solid-State Circuits Conference [4] and an introduction to a performance scaling trend perspective for security primitives. The latter provides insight into the recent and foreseeable evolution of the state of the art, in terms of both technical advances and new directions to pursue. Technology trends are derived from new public databases summarizing the state of the art, such as the Physically Unclonable Function database (PUFdb) and the hardware security database (HWsecdb). Also presented is a reasoned review of prior art in circuits and subsystems for hardware security, from the very basics to best-in-class silicon demonstrations. As the emphasis here is on ubiquitous (i.e., down to low-end devices) and always-on security, this article focuses on solutions suitable for tightly energy- and area-constrained systems.

Digital Object Identifier 10.1109/MSSC.2019.2923503
Date of publication: 27 August 2019

1943-0582/19©2019IEEE

Hardware Security

Security has become a crucial dimension in the design of systems on chip (SoC) to prevent attacks that can interfere with our everyday actions that rely on the exchange of confidential data, including those related to finance, purchases, banking, health care, government, and blockchain transactions (e.g., cryptocurrency and much more in the years to come). The general objective of data security assurance is summarized in Figure 1. The sender (the traditional character Alice) intends to share confidential information with the receiver (Bob) over an insecure channel, where malicious users can perform actions that violate some basic requirements of the data transaction. For example, eavesdropping (see Eve in Figure 1) aims to extract information from the observation of the channel, password breaking (Craig) tries to find the secret password to allow full access to the transmitted data and enable the attacker to impersonate the receiver, and intrusion (Trudy) pursues the alteration of the message being exchanged. Such considerations for in-transit data are immediately extended to data at rest (e.g., stored on a device). The following security services are commonly required to preserve data:

- *confidentiality*: preventing unauthorized disclosure of the exchanged data (e.g., eavesdropping)
- *integrity*: guaranteeing that the message was not altered
- *authentication*: confirming the identity of the sender
- *nonrepudiation*: guaranteeing that no other sender could have sent the message (e.g., needed in payment transactions to avoid repudiation of actually intended transactions)
- *digital signature*: publicly verifying the integrity of the message and performing authentication and nonrepudiation.

As it is unfeasible to assure data security with absolute certainty, such properties are guaranteed only in a statistical sense (i.e., under a targeted level of confidence). These security services are guaranteed by first establishing a shared secret (e.g., a cryptographic key) between the sender and receiver and then using cryptographic algorithms to enforce the property of interest via message encryption/decryption with the key. Because trust between the sender and receiver is founded on the shared secret and the ownership of the cryptokey, the latter is often called the *root of trust*. In contrast, the crypto-algorithm is invariably assumed to be publicly known (Kerckhoff's principle). Indeed, relying on the secrecy of the algorithm would not assure security in the long term: it is only a matter of time before the adopted encryption algorithm is uncovered.

Once the root of trust is established, its security properties have to be propagated through the entire chain of trust, encompassing all levels of abstraction (including the hardware, software, application, and service level). This article focuses on hardware security and thus from the transistor to the architectural level. The shift from software security to hardware security is increasingly required in a wide range of applications, from systems demanding a stringent level of security to tightly constrained systems where the large power penalty of traditional software security methods cannot be afforded.

Hardware security assurance has become more challenging and of wide interest due to two considerations. The first lies in the exponential increase of connected devices, which introduces an unprecedented number of backdoors in existing and prospective networks [the ultimate 1-trillion-device scale of the Internet of Things (IoT) is a clear example of this]. The second set of additional challenges comes from the fragmentation of the semiconductor supply chain: an untrusted player might have the ability to insert malicious hardware (e.g., a hardware trojan) at any stage of the chip lifecycle, including design, manufacturing, packaging, and testing.

As a representative and motivational example of a silicon system requiring state-of-the-art hardware security techniques, the architecture of e-wallets for cryptocurrencies is shown in Figure 2. Such wallets store the private keys necessary to manage a cryptocurrency account as proof of public key ownership, giving unrestricted power to the key owner (e.g., for payments and transfers). The economic value unlocked by hardware wallets can indeed be very high, thus requiring the highest level of security. Similar considerations hold for many other prospective applications of the blockchain. E-wallets are the most secure type of cryptocurrency wallets [1], as their silicon implementation makes them invulnerable with respect to software viruses and vulnerabilities (as no operating system runs on them). On-chip storage encryption prevents access to the keys in case of e-wallet theft. As shown in Figure 2, SoCs for e-wallets include various security-specific subsystems, including random key generation (often called *entropy generation*), cryptographic primitives for encryption and hashing, and protections against physical attacks. The next sections focus on the first two categories of subsystems; readers are referred to the extensive literature on physical attack counteraction for the third.

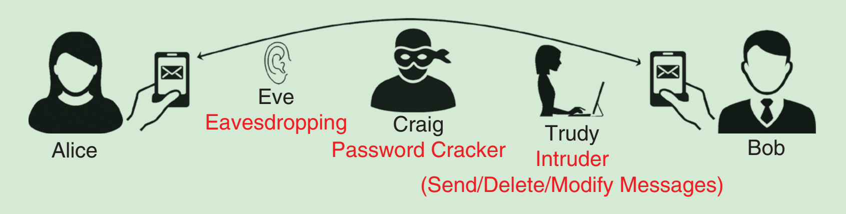


FIGURE 1: The information exchange between sender (Alice) and receiver (Bob) over an insecure channel where malicious users can monitor or alter transmitted data.

Entropy Generation

The generation of keys is an essential task in secure systems, whether the key is used for cryptographic operations or to identify (e.g., with chip ID) and authenticate a device or for cryptographic operations. Key generation can be either static (see the section "Static Entropy Generation and PUFs") or dynamic (see the section "Dynamic Entropy Generation and Random Number Generators"), depending on whether the secret key needs to be generated once and for all or afresh. When there is no obvious or exploitable vulnerability in a secure system (i.e., from the cryptoalgorithm or its specific implementation), the adversary is forced to break into it by retrieving the key via exhaustive search. The effort entailed by such a brute-force attack is proportional to the number of possible key guesses in

$$\textit{key space size} = 2^{\text{keylength}\cdot\text{entropy}}, \quad (1)$$

where "keylength" is the bit width of the key (e.g., 128 b), and "entropy" is the well-known amount of information carried by each bit, which is expressed in bits and ranges from 0 (no information and a perfectly predictable value) to 1 (each key bit carries information of a full bit). Under a perfectly random key, entropy =1 and the security strength of the system is $2^{\text{keylength}}$. Under practical and imperfectly random key generation methods with entropy <1, from (1), the effective key length (i.e., keylength · entropy) is reduced, degrading the security strength.

Using private-key cryptography as a highly representative example [e.g., the Advanced Encryption Standard (AES) cryptostandard], practical key lengths assuring an adequate security strength range from 128 to 256 b. In particular, 128 b is generally considered sufficient for most applications in the upcoming decade, as breaking the key requires the adversary to have resources in the low millions of dollars range and the attack to be performed by a large organization (e.g., see [9]). Keys of 192 b increase the number of resources the adversary requires by an order of magnitude. Thanks to the extremely high attack cost, 256 b is considered very secure. The foreseeable introduction of quantum computers will require that such key lengths be doubled to withstand attacks based on the most efficient quantum algorithms for function inversion (i.e., Grover's algorithm [26]).

The entropy in (1) can be defined in various ways, depending on the application. The popular Shannon entropy definition is used for moderately stringent security requirements (e.g., chip ID). From Figure 3, entropy is degraded compared to its ideal value (equal to 1 in practical cases), where the probabilities Pr[0] and Pr[1] of having 0 and 1 are not perfectly equal. As common rule of thumb, entropy is considered to be adequate when the effective key length [keylength · entropy in (1)] is degraded by less than 1 b compared to its ideal value equal to keylength. For a 256-b key, this translates into an entropy requirement of 0.998 or equivalently maintaining Pr[0] (or Pr[1]) close to the ideal value of 0.5 (more precisely, between 0.42 and 0.58), as seen in Figure 3 [8], [20], [41]. Instead, the min-entropy definition [75] is generally adopted for security quantification of cryptokeys and passwords.

Min-entropy is a conservative measure of the key unpredictability: it is defined as the probability of a successful guess of the most likely key value (for example, if a given key has a 50% chance to be generated, the min-entropy is 1 b, as the uncertainty is equivalent to guessing a single truly random bit). In the example of a 256-b key, the min-entropy target of 0.998 translates into the much more stringent requirement of maintaining Pr[0] between 0.497 and 0.503, as shown in Figure 3 (corresponding to an equivalent Shannon entropy of 0.99997). These considerations dictate the maximum 0/1 bias requirement of circuit implementations for static and dynamic entropy

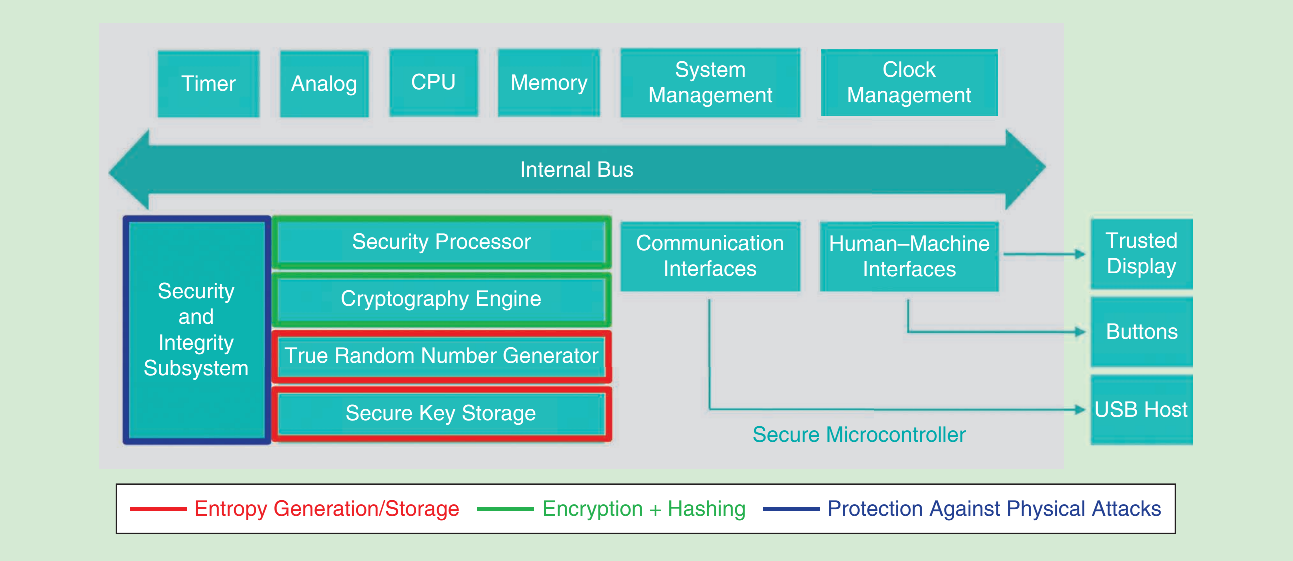


FIGURE 2: The typical architecture of e-wallets for cryptocurrencies and related security subsystems.

generation, as discussed in the following sections.

Static Entropy Generation and PUFs

The on-chip availability of fixed keys is essential in security, as exemplified by the retention of chip ID to unambiguously identify a device, the protection of software intellectual property (IP) running on a chip, the remote attestation of hardware and software integrity, and the creation of a chip-specific root of trust that depends on the underlying chip. Traditionally, static entropy is simply generated off chip before deployment (e.g., at testing time) and stored in a nonvolatile manner (e.g., fuses and flash memory). However, such storage methods are well known to be vulnerable to a wide range of well-documented attacks at the software level as well as physical attacks ranging from noninvasive to invasive (e.g., see [3], [22], [39], [64], [76], [77], and [79]). To overcome the limitations of traditional key storage, it is necessary to ensure that the following properties are met:

- Physical inspection of the die (e.g., imaging and reverse engineering) should not expose the secret, as opposed to nonvolatile memories and fuses.
- The secret keys should be physically available only when the chip is powered on, so that the adversary cannot retrieve the key when powered off because, in this case, the device would be vulnerable to physical attacks due to the related protection techniques being disabled.

To overcome these challenges, PUFs (i.e., physically unclonable functions) have been extensively explored and more recently introduced in commercial chips and as a commercially available design IP. From a behavioral viewpoint (see Figure 4), PUFs are ideally digital blocks that respond to inputs (challenges) with perfectly repeatable outputs (responses), where the input–output mapping is unpredictable and unknown to an external observer. In PUFs, the responses are not stored but recreated on the fly, as defined by chip-specific random (within-die) variations, thus requiring the chip to be powered on to deliver the responses and satisfy both previously described properties. The challenge-response pairs (CRPs) represent the root of trust as stored in the form of golden responses or golden keys on a secure server at a preliminary enrollment phase before chip deployment. After enrollment, CRPs are no longer accessible from off chip, disabling the test port that previously scanned out the CRPs.

PUFs can be categorized into two classes. Weak PUFs exhibit a number of CRPs that are approximately linear with the PUF silicon area or the number of available PUF bit cells. The limited availability of CRPs in weak PUFs mandates that CRPs are not disclosed in plain over the insecure channel, and they should instead be encrypted or used as cryptokeys (i.e., the PUF is necessarily coupled with a crypto-algorithm). Accordingly, all generated response bits of weak PUFs must be perfectly stable when the same challenge is repeatedly applied because even a single bit change would completely disrupt the output of the coupled crypto-algorithm. Weak PUFs have been used for several security purposes, such as chip ID and authentication [10], [27], [71], [80], lightweight encryption [56], [93], secure exchange of private keys with no involvement of public-key cryptography [2], hardware-entangled cryptography [103], identification of malicious hardware [50], and strong PUF creation via synergy with crypto-engines [18].

On the other hand, strong PUFs exhibit a number of CRPs that increase exponentially with the silicon area. The abundant availability of CRPs in strong PUFs allows in-plain

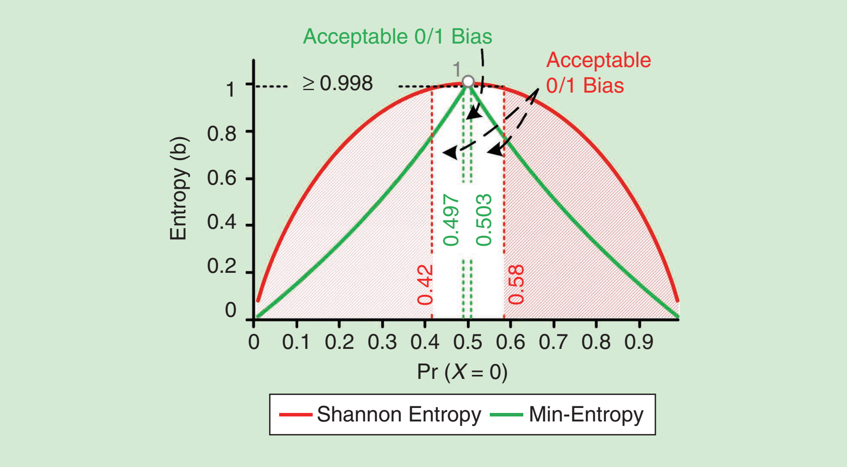


FIGURE 3: The Shannon entropy and min-entropy versus Pr[0] as well as numerical examples of targets to keep the effective key-length degradation lower than 1 b in a 256-bit key.

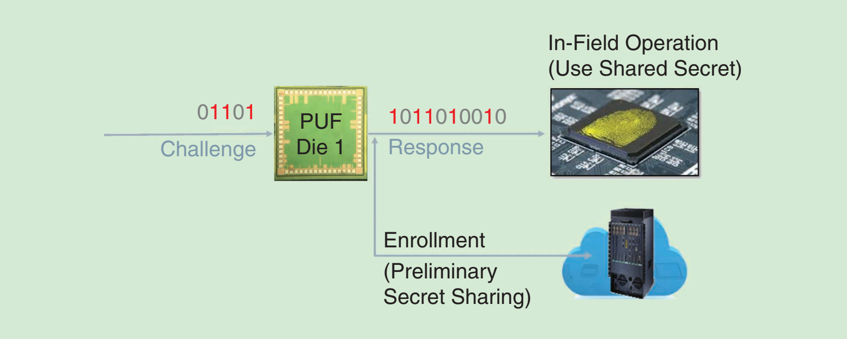


FIGURE 4: The PUF operation at enrollment and in field.

transmission of CRPs because replay attacks are inherently prevented by the very low probability of reusing a CRP. This enables the adoption of the popular and particularly simple security protocols based on CRP exchange, where the response is compared to the golden response in the server. As a major difference with respect to weak PUFs, such protocols for strong PUFs can forgive the instability of a limited number of bits in the response (e.g., by associating the correctness of a response to the adequate closeness to the golden value rather than to bit-accurate matching). In other words, bit stability in strong PUFs is relaxed compared to that in weak PUFs.

PUF Quality and Fundamental Metrics

PUFs are essentially made up of circuits that magnify within-die variations while rejecting the effect of all other variation contributions, such as

- die to die—to maintain uniform statistical properties of the response across dice
- voltage and temperature—so that the responses are nearly independent of their inevitable fluctuations
- aging—to maintain consistent response throughout the life of the device.

The quality of a PUF is assessed through a number of well-established metrics that quantify the PUF response stability, degree of randomness, sensitivity to variations, area and energy efficiency, and so on [7]. As the first metric for PUF stability, the unstable bit count is the cumulative count of occasionally flipping bits over the whole population of PUF bit cells for a given number of repeated iterations under the same challenge. The unstable bit count provides information on the worst-case incorrect bit count under the pessimistic assumption that all bits can be simultaneously flipped (e.g., they exhibit correlation, whereas their actual flipping might also be affected by random phenomena such as thermal noise) and no stability-enhancement method is introduced. As a more relevant stability metric, the bit error rate (BER) counts the average of the simultaneous instability for the PUF word output, which, in turn, sets the key error rate (KER) (i.e., the probability of having at least one flipped bit in a response). Because an incorrect key leads to a failing transaction (e.g., in the communication between two devices), the KER needs to be kept low enough so that the medium time before fault (MTBF) is comparable to, or at least a significant fraction of, the life of the device.

By definition, the MTBF is the ratio of the average PUF interaccess time $t_{\text{inter-access}}$ (i.e., the time between two successive PUF accesses) and the probability of the KER having a PUF failure due to instability. In a typical case of a duty-cycled sensor node where a measurement is taken every time the node is woken up, $t_{\text{inter-access}}$ is simply the wake-up period [2] (i.e., the period following which the node moves to the active mode after entering sleep mode). As per the plot in Figure 5, a reasonable MTBF on the order of years requires the KER to be on the order of a typical target of 10^{-6} in a PUF with $t_{\text{inter-access}} = 1\,\text{min}$. Such KER targets can be relaxed by two orders of magnitude or more when $t_{\text{inter-access}}$ increases to hours or longer.

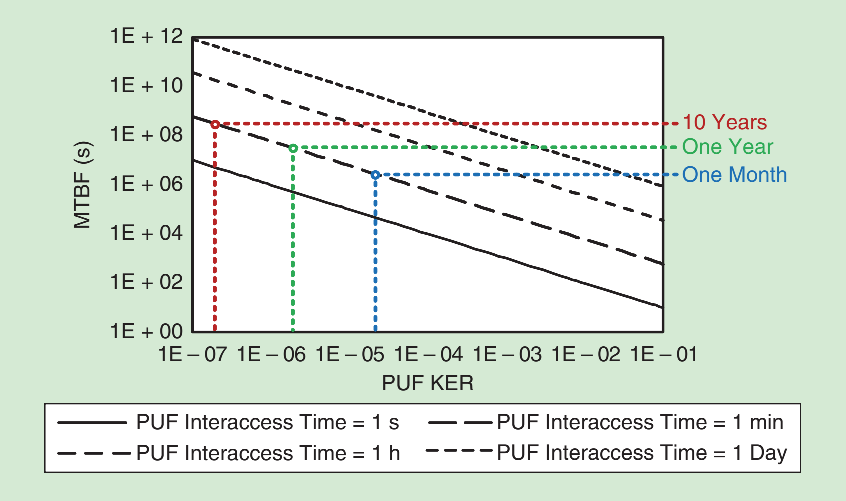


FIGURE 5: The PUF MTBF versus KER for various PUF interaccess times.

The repeatability of the responses is quantified by the average intra-PUF Hamming distance (HD) between the response and the golden key (i.e., the number of bits by which they differ) [50]. Better repeatability makes the intra-PUF HD closer to the ideal value of 0. The PUF uniqueness is quantified by the inter-PUF HD as defined by the average HD between the responses to the same challenge coming from different dice [50]. If perfectly random, the response of two dice to the same challenge would differ by 50% of their bits on average. In actual PUF implementations, the inter-PUF HD is statistically distributed, and the deviation of its average from the ideal value measures how unique (i.e., chip specific) the PUF responses are. The identifiability is related to both the intra- and inter-PUF HD, as a PUF in a die is easier to identify from other dice if the inter-PUF HD is large (i.e., close to 50%) and the intra-PUF HD is small (i.e., close to 0). Accordingly, the identifiability is defined as the ratio of the average intra- and inter-PUF HD [50].

The randomness of the responses is measured by various metrics. The most immediate is the 0/1 bias (i.e., the probability of having a 0 or a 1), the difference of which compared to the ideal 50% value is a proxy for the level of randomness degradation. Being interdependent (see the “Entropy Generation” section), the Shannon entropy of the responses is routinely used as a metric. The more stringent suite of National Institute of Standards and Technology (NIST) statistical pass/fail tests [72] is routinely used to assess whether an adequate level of randomness is achieved. The 0/1 bias requirement

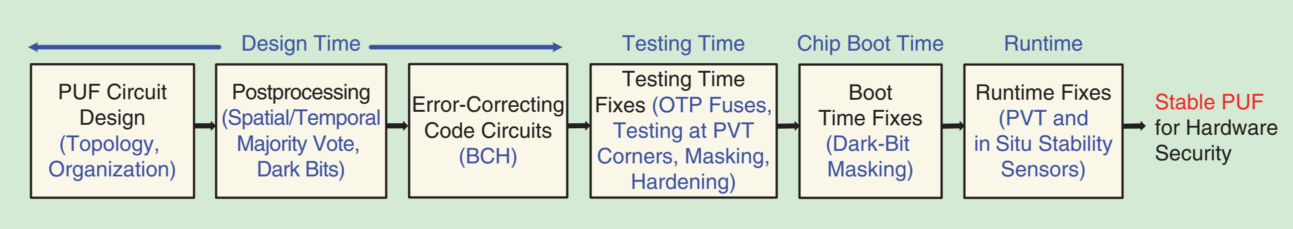

Design Time
Testing Time
Chip Boot Time
Runtime
PUF Circuit Design (Topology, Organization)
Postprocessing (Spatial/Temporal Majority Vote, Dark Bits)
Error-Correcting Code Circuits (BCH)
Testing Time Fixes (OTP Fuses, Testing at PVT Corners, Masking, Hardening)
Boot Time Fixes (Dark-Bit Masking)
Runtime Fixes (PVT and in Situ Stability Sensors)
Stable PUF for Hardware Security

FIGURE 6: **Examples of methods to mitigate PUF instability in different phases of the chip lifecycle. OTP: one-time programmable; PVT: process/voltage/temperature.**

from NIST tests is equivalent to the min-entropy in Figure 3 [43].

Qualitative randomness analysis is often performed visually through speckle diagrams, which visualize the output bits versus the challenge in a black-and-white, matrix-like diagram [2]. Speckle diagrams reveal obvious spatial patterns and correlations between response bits and thus allow observation of the effect of layout-dependent variations. Quantitatively, such variations are measured by the autocorrelation function of the concatenated response bits and, in particular, by the function's value at a 95% confidence level [2]. Other important and architecture-independent PUF metrics are area per bit and energy per bit, as well as the voltage (temperature) sensitivity that quantifies the percentage increase of unstable bits and BER normalized to the supply voltage (temperature) change that caused it.

PUF Stability Improvement

The stability of raw PUFs is generally inadequate to assure the targeted KER (see the section "PUF Quality and Fundamental Metrics"), which mandates the adoption of additional techniques to meet the KER requirement. As summarized in Figure 6, at design time, the PUF stability can be improved through the circuit and physical design of the bit cell as well as the PUF organization (e.g., redundancy). As the inadequately stable bits can be effectively treated as errors, error-correcting codes (ECCs), such as Bose–Chaudhuri–Hocquenghem (BCH) codes, are almost invariably introduced to suppress their effect on the response [50]. However, the ECC area and energy cost are typically two orders of magnitude larger than the PUF itself [4], and they linearly increase with the number of correcting bits [85]. In particular, the typical gate count of BCH implementations is at least several tens of thousand gates [85], which is very expensive compared to the similar (or even lower) complexity of typical on-chip microcontrollers utilizing the PUF. Accordingly, postprocessing techniques are introduced at the output of the raw PUF to reduce its BER and, therefore, the required number of ECC correcting bits.

Among the many postprocessing techniques at design time (see Figure 6), spatial majority vote (SMV) combines multiple PUF bits to generate a response bit that is more resilient against the instability of individual bits. SMV is effective in mitigating voltage- and temperature-induced instability because the bit cells with low stability margin and, thus, higher sensitivity to voltage and temperature are statistically infrequent. Similarly, the same PUF bit can be generated multiple times and postprocessed with a temporal majority vote (TMV) to mitigate the effect of on-chip noise on instability. At testing time, masking is widely used to eliminate the PUF bits that are so unstable they cannot be corrected by postprocessing and ECC. Similarly, hardening the chip via operation at voltage and temperature above the ratings at testing time improves stability in marginally stable bit cells [60]. Hardening is economically feasible only if the considered application requires a chip burn-in on the whole production sample (as opposed to a random sample), as in the case of automotive, biomedical, aerospace, and industrial applications. For example, hardening is generally infeasible for IoT sensor nodes in view of their tight cost constraint. As another technique, dark bit masking [81] extends the masking concepts to the chip boot time, during which transiently flipping bit cells are masked to eliminate the ones that are marginally stable (this does not capture the effect of voltage and temperature in bit cells that are flipped but stable at the considered environmental conditions).

Recently, PUF instability mitigation has begun moving to runtime, thanks to the introduction of on-chip sensors that detect the actual instability margin instead of pessimistically margining it for the worst-case process/voltage/temperature (PVT) corner and noise [85]. As shown in Figure 7, runtime BER monitoring comprises PVT sensors as well as inexpensive in situ (i.e., bit-cell level) instability sensors, whose output is merged into a simple BER machine learning model that estimates the number of necessary correction bits in the ECC instead of using its worst-case value. The reconfigurable ECC in Figure 7 thus operates at the strictly necessary number of correction bits and energy instead of at the worst-case number and highest energy [85]. As a side benefit, the estimated correction bits allow the detection of malfunctions and fault injection attacks, where the adversary alters the operating conditions to induce unintended behavior and expose vulnerabilities.

Weak PUF State of the Art and Trends

PUFs are invariably based on comparing nominally equal voltages/currents, or the current contention of nominally equal devices, where the random output is determined by random mismatch. The first silicon PUF proposed belongs to the category of delay-based PUFs and uses N pairs of nominally equal ring oscillators [27]. Their response bit is 1 (0) if the first oscillator is faster (slower) than the second, depending on the mismatch. The resulting circuit is a weak PUF, being the maximum number of independent comparisons $\log_2 N! \sim N^{1.3}$ [50]. Ring oscillator PUFs have relatively poor stability [10] because most ring oscillators have a frequency lying around the mean value of their distribution, making the frequencies of their pairs generally close to each other and also making the frequency comparison sensitive to noise and the PVT corner. This is a general challenge in existing PUFs based on comparing nominally equal structures, as discussed previously.

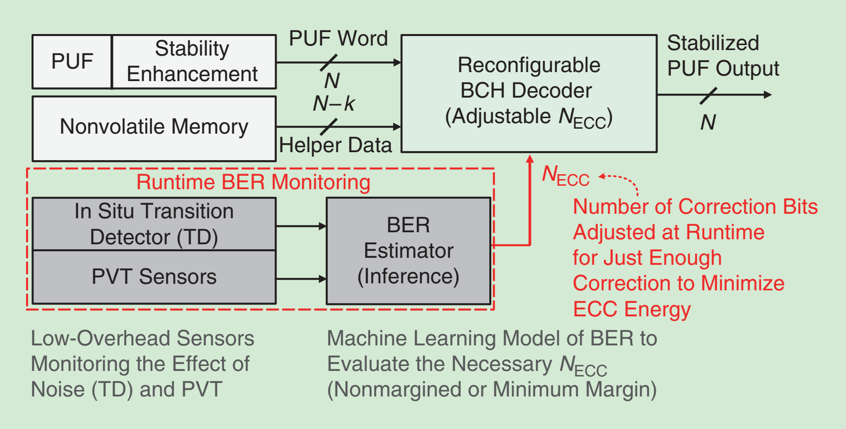

PUF
Stability Enhancement
PUF Word
N
N−k
Nonvolatile Memory
Helper Data
Reconfigurable BCH Decoder (Adjustable N_{ECC})
Stabilized PUF Output
N
N_{ECC}
Runtime BER Monitoring
In Situ Transition Detector (TD)
PVT Sensors
BER Estimator (Inference)
Number of Correction Bits Adjusted at Runtime for Just Enough Correction to Minimize ECC Energy
Low-Overhead Sensors Monitoring the Effect of Noise (TD) and PVT
Machine Learning Model of BER to Evaluate the Necessary N_{ECC} (Nonmargined or Minimum Margin)

FIGURE 7: Techniques to mitigate PUF instability are now moving to in situ sensing and runtime compensation [85].

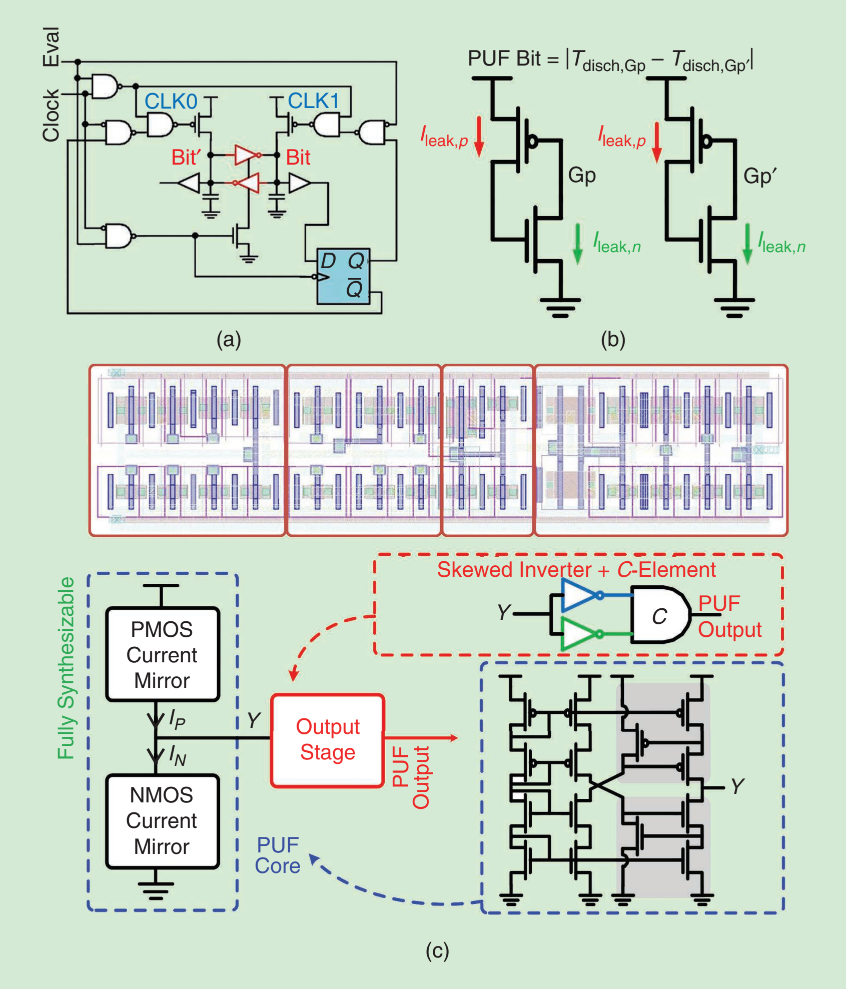


FIGURE 8: Examples of recently proposed weak PUFs: (a) metastability based [82], (b) thyristor based [47], and (c) static monostable for standard cell-based design through digital automated flows. [86]. CLK: clock.

Memory-based PUFs are a very popular and commercially available class. The most widely adopted is based on the preferred power-up state of individual static random-access memory (SRAM) bit cells, which is determined by the mismatch-induced asymmetry in the strength of the two sides of differential bit cells [28], [73], [90], [104]. SRAM PUFs are area efficient thanks to their inherently high bit-cell density, and the ubiquitous presence of SRAMs in SoCs makes this raw PUF widely available. However, the relatively poor stability of SRAM PUFs requires a rather large area and energy cost due to the necessary postprocessing and ECC. The same principle of a preferred power-up state has been exploited in other memory-based PUFs, such as the butterfly [37] (i.e., cross-coupled flip-flops), latch [80], and D flip-flop PUF [61].

Analog PUFs are based on the generation of nominally equal voltages or currents and comparison with a reference or a pair-wise comparison to leverage the insensitivity of differential structures to PVT variations. For example, the compact two-transistor voltage reference in [49] has been adopted as an elementary cell of an array, whose pairs of temperature-compensated voltages are compared to generate random bits based on the relative mismatch. Metastability and positive feedback in memory elements [e.g., the cross-coupled inverters in red in Figure 8(a)] have also been used as a source of static entropy, leveraging the natural tendency to settle to a preferred state determined by mismatch after being reset or precharged [56], [82]. Positive feedback is also the core mechanism used in thyristor-based PUFs (i.e., PMOS and NMOS transistors whose drain drives the gate of the other), as seen in Figure 8(b). In this PUF, the output nodes Gp and Gp′ in Figure 8(b) are first precharged to the supply voltage and then let free to discharge toward the ground [47]. The output bit is determined by which side is faster and, thus, by mismatch.

The class of static monostable PUFs introduced in [105] is based on PUF bit cells that generate a static output and have only one stable state. The static behavior assures that the output is independent of coupling noise and insensitive to routing (as opposed to, for example, delay-based PUFs). Monostability ensures that the correct output bit is delivered, even when occasionally intense transient noise flips the bit cell (as opposed to memory-based PUFs). This is achieved by connecting back-to-back current mirrors, as in Figure 8(c), where the output is high (low) if the PMOS (NMOS) current mirror is stronger than the other due to mismatch. The output is essentially full swing because of the inherently high small-signal output resistance of current mirrors, which assures a large output voltage change under a small PMOS/NMOS current mismatch [105].

This PUF has been shown to be suitable for the standard cell layout style and also to be placement-independent because of negligible layout-dependent variations. This is achieved by crafting the bit-cell layout to homogenize the layout environment around the mismatch-critical transistors (i.e., current mirrors) across all bit cells, keeping them at the center of the bit cell, and consistently adding decaps next to the bit cells. The resulting bit cells can be freely placed and routed to design PUFs in a matter of hours rather than several weeks using a fully automated digital design flow [86]. This is different from other PUFs that require significant design effort in view of the analog-style bit-cell layout and design (e.g., a ring oscillator) or array organization (e.g., metastability based). In [86], hysteresis and temperature compensation have been added to improve PUF stability. For another example of a PUF in the same class, two-transistor voltage generation and mismatch amplification via two-transistor gain stages have been demonstrated in [96].

A recently proposed highly stable and ECC-less class is based on the induction of the random presence or absence of a resistive path in the bit cell. This is achieved at manufacturing time by the via PUF [106] purposely violating the via spacing design rules to randomly create shorts or open circuits in pairs of neighboring vias, depending on random manufacturing imperfections. A similar outcome is achieved at testing time through intentional oxide rupture [91], [92] by applying a stress voltage in transistor pairs, the response bit being determined by which transistor breaks first. In this PUF, a resistive path is created between the gate and drain of the transistor experiencing breakdown, while the other does not have such a direct gate-drain path. Although highly stable and area efficient, these circuits are actually not PUFs, strictly speaking. Indeed, their physical inspection (e.g., delayering and imaging) can potentially reveal the secret bits. In addition, PUF responses are available even when the device is powered off, as the information is stored in a nonvolatile manner, thus violating the previously described fundamental PUF properties.

A thorough review of PUF state of the art and related technological trends is available in the PUFdb in [69]. Based on the PUFdb data, the trend of the area per bit in Figure 9(a) shows that PUFs are generally becoming more area efficient over time, with the metastability and static monostable classes exhibiting the largest improvement of 1.4×/year, as expectable from their digital nature. The area of analog PUFs is instead stagnant due to the expectedly slow shrinkage of analog circuits across CMOS generations. As shown in Figure 9(b), the digital nature of metastability-based and static monostable weak PUFs has enabled a rapid improvement in the energy

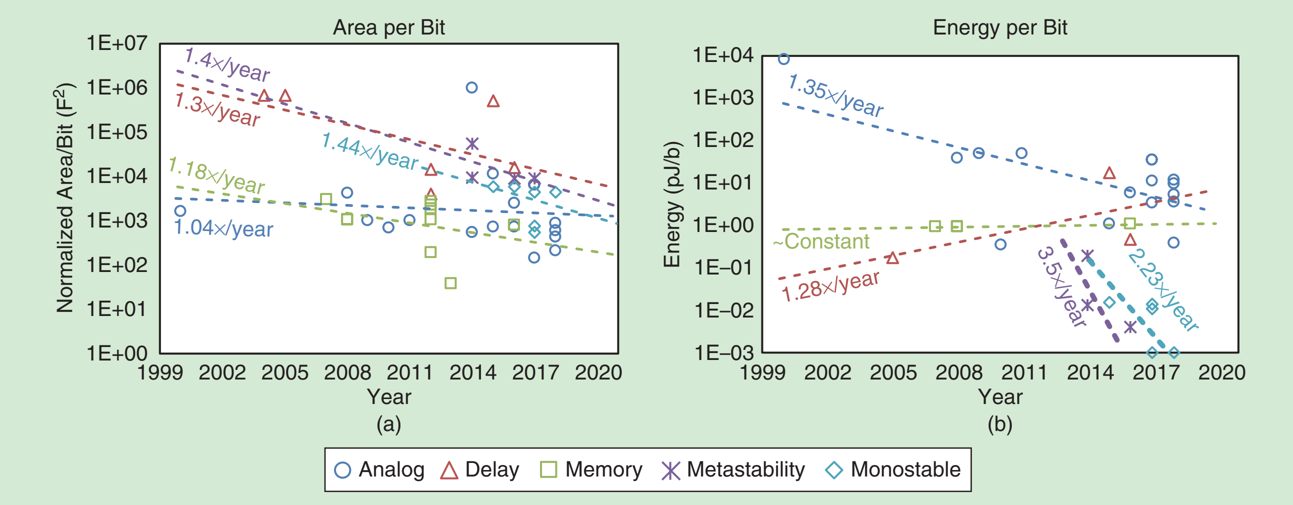


FIGURE 9: State-of-the-art PUF trends: (a) area (normalized to F^2, where F = minimum feature size of the process) and (b) energy consumption [69].

	JSSC 2018 [107]	JSSC 2018 [86]	ISSC 2018 [47]	ISSC 2017 [96]	JSSC 2016 [105]	JSSC 2017 [82]	ISSC 2016 [108]	ISSC 2014 [78]	
Entropy Source	PTAT Voltage Reference (SRAM)	Static Monostable	4-T Thyristor Leakage	2-T Amplifier	Static Monostable	Metastability	Biased NAND	Metastability	
Technology	65 nm	40 nm	180 nm	180 nm	65 nm	14 nm	45 nm	22 nm	
Voltage (V)	1	0.8–1	1.2–1.8	0.8–1.8	0.6–1	0.55–0.75	–	0.7–0.9	
Temperature (°C)	15–85	–40–125	0–80	–40–120	25–85	25–110	–25–85	25–50	
Bit-Cell Area/bit (F^2)	600	3,650	445	553	6,000	11,000	3,100	11,000	①
Native Worst BER (%)	2.16	3.2	1.2	3.13	–	5.76	2.9 (No VDD Var.)	8.5 (No T Var.)	②
Voltage Sensitivity (%/V)	–	2.9	–	–	13	–	–	4.9	
Temperature Sensitivity (%/°C)	0.11	0.015	–	0.02	0.047	–	0.025	–	③
Native Unstable bits (%)	5.39	3.48	6.65	1.73	2	~26.8	–	30	
Energy (fJ/b)	380	1.02	9,800	13.5	15	4	–	13	④
Inter-PUF FHD	0.502	0.4907	0.492	0.499	0.501	0.486	0.498	0.49	
Intra-PUF FHD	0.0015	0.0049	0.0072	0.0008	0.0035	0.034	–	0.026	
Identifiability	332	102	68	623	140	14	–	19	⑤
Entropy	–	0.9972	–	–	0.9966	0.99993	0.999998	0.9997	
Automated Design	No	Yes	No	No	No	No	No	No	⑥
Stabilization Methods	TMV + Comparator Swap	Temp. Feedback Compens.	TMV + Remapping	TMV11	No	Bit Destabilization, TMV, Soft-Bit Masking, Hardening	SMV, Masking, ECC	Soft Dark-Bit Masking	

① Up to Tens of SRAM Bit Cells

② Instability in the Range of a Few Percentage Points

③ 2–8× Less Temperature Sensitive With Active Compensation

④ Energy Down to femtojoule per bit

⑤ Inter-/Intra-PUF Hamming Distance Up to Hundreds

⑥ Stdcell-Based Design Is Now Possible

(a)

	TCAS-I 2018 [109]	VLSI 2017 [98]	VLSI 2017 [33]	ISSCC 2015 [95]	
Entropy Source	Inverter Logic Threshold	Sub-*V*th Current Array	2 × 6T SRAM Bit-Cell	Even-Stage Ring Oscillator	
Nonlinearity Source (Machine Learning Attack Resilient)	16 Out of 256 Selection	Series/Parallel Connection	Wordline Permutation	–	
Technology (nm)	40	130	28	40	
Voltage (V)	0.9–1.3	1.08–1.32	0.5–0.9	0.7–1.2	
Temperature (°C)	–40–90	–20–80	0–80	–25–125	
PUF Core Area ($10^6 \cdot F^2$)	2.94	2.64	1.45	0.53	
Native Worst BER (%)	6.1	9	12	9	①
Native Unstable Bits (%)	4.92	–	–	–	
Native PUF Energy (pJ/b)	7.7	11	0.097	17.75	②
Throughput (Mb/s)	0.5	0.006	1,100	1.6	③
Number of CRP Supported	1.8×10^{19}	3.7×10^{19}	1.17×10^{11}	5.5×10^{28}	
Resilient to Modeling Attacks	Yes	Yes	Yes	No	

① Higher BER Than Weak PUFs (Acceptable)

② Energy Much Higher Than Weak PUFs (but Comparable to ECC)

③ Throughput Typically Much Lower Than Weak PUFs

(b)

FIGURE 10: A summary of the recent state of the art in (a) weak PUFs and (b) strong PUFs. PTAT: proportional to absolute temperature; VLSI: very-large-scale integration.

per bit of better than 2×/year, largely exceeding the expectable reduction from technology scaling, thanks to continuous circuit innovation. The energy of memory-based PUFs is not benefitting from technology scaling, as expected from the limited reduction in the wordline and bitline capacitance across CMOS generations.

A summary of the performance achieved by recent weak PUF state of the art is reported in Figure 10(a), which shows that the area per bit can be up to several tens of times larger than an SRAM bit cell. The worst-case native BER is now on the order of a few percentage points. The adoption of active temperature compensation has brought temperature sensitivity to approximate a percentage point for a typical 100 °C temperature fluctuation [86]. Relentless improvements in energy efficiency have brought the energy per bit down to the femtojoule level [86]. The identifiability has now reached values of up to several hundred [96], [107].

Strong PUF State of the Art and Trends

The abundant availability of PUF responses of strong PUFs translates into a high level of security only if the computational effort to model them is adequately large. The first strong PUF proposed is the arbiter PUF [46], but it is not useful in real applications because it is highly vulnerable to modeling and machine learning attacks [70] (i.e., any response can be modeled as a simple function of a few random components and thus, easily predicted for any challenge). Indeed, the arbiter PUF simply compares two logic path delays, each of which is the sum of gate delays, which are, in turn, selected by the challenge bits through multiplexing. Observation of the responses and related challenges allows identification of the basic gate delays with only a linear computational effort with respect to the number of observations (similar to the solution of a set of linear equations). Accordingly, the actual challenge in strong PUFs is to combine a very large number of CRPs with a mechanism that combines multiple sources of randomness in a strongly nonlinear manner (so that the computational effort of their modeling becomes very high).

Representative examples of state-of-the-art strong PUFs are reported in Figure 11. The PUF in Figure 11(a) is based on the concept of oscillation collapse in a ring with an even number of inverter gates, which is inherently bystable [95]. The transitions throughout two opposite sides of the ring propagate and race against each other until one of the two stable states is reached to deliver the PUF response. The race-through process is governed by the inverter delays, each of which is determined by the corresponding challenge bit via a binary choice of inverter replicas, and the response is determined by the number of cycles necessary to observe the oscillation collapse. Because no explicit nonlinearity is introduced, the PUF is potentially vulnerable to machine learning attacks.

An explicit nonlinearity is introduced in the sequence-dependent SRAM PUF in [33], where pairs of opposite-valued bit cells sharing the same bitline are simultaneously activated to purposely trigger their current contention. The latter generates a random PUF bit, and the process is then iterated multiple times across different pairs to ultimately determine the corresponding response bit. The necessary PUF nonlinearity arises from the dependence of the output on the sequence of pairs experiencing contention. The strong PUF-based sub-*V*th current array in [98] performs the comparison of the current delivered by two transistor arrays in deep subthreshold, whose outcome defines the corresponding response bit. The nonlinearity stems from the nonlinear dependence of the subthreshold current on the number of stacked transistors and parallel transistor stacks, which is, in turn, determined by the challenge.

As extracted from the PUFdb [69], Figure 10(b) summarizes some highly representative strong PUFs proposed in the last few years. The BER of strong PUFs is allowed to be larger than that of weak PUFs because of the one-time

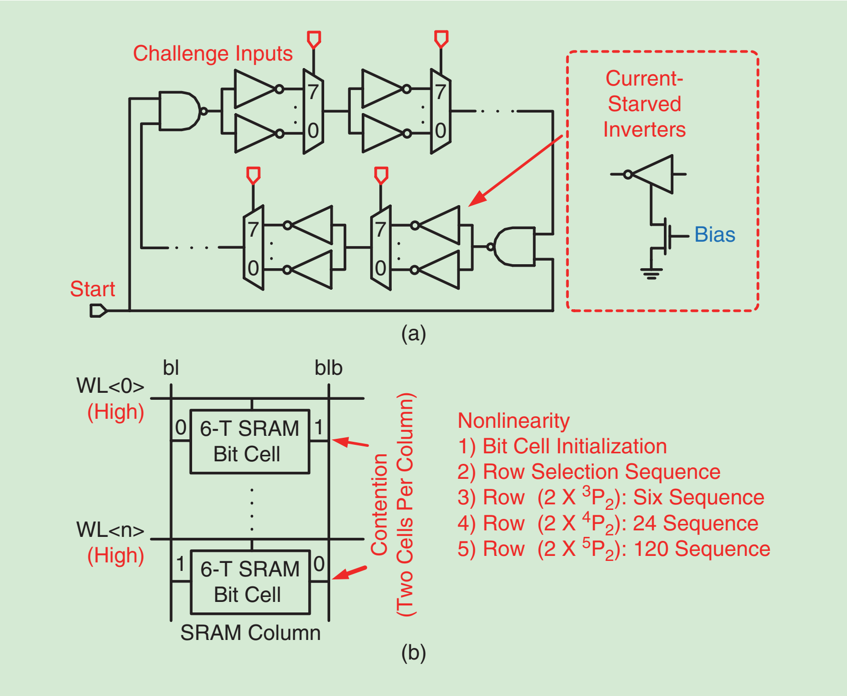


FIGURE 11: Examples of recently proposed strong PUFs based on (a) oscillation collapse [95] and (b) nonlinear sequence-dependent SRAM [33]. CTAT: conversely proportional to absolute temperature.

only usage of each CRP (see the discussion in the section "Static Entropy Generation and PUFs"), and it is typically in the 10% range. The energy is two to three orders of magnitude higher than in weak PUFs and typically in the picojoule/bit range. Such energy is comparable to weak PUFs, including the energy contribution of postprocessing and ECC (see the section "PUF Stability Improvement"). As seen in Figure 10(b), the throughput of strong PUFs is severely degraded compared to weak PUFs.

Dynamic Entropy Generation and Random Number Generators

Random number generators (RNGs) are another fundamental primitive of hardware-secure systems in view of the necessity to generate on-the-fly random bits for the creation of fresh session keys, nonces and initialization vectors for encryption/decryption, and bit padding when the word being encrypted does not fit the encryption block size [74], [78]. RNGs can be classified as true RNGs (TRNGs) and pseudo-RNGs (PRNGs). TRNGs are based on a random physical process and are thus inherently much more secure because future or past values cannot be inferred from present values. PRNGs are deterministic systems generating bit sequences that appear to be random, but their entire trajectory can be predicted once the adversary knows the seed or an intermediate state. The much higher level of security in TRNGs is obtained at the cost of worse throughput, energy, and statistical quality (e.g., entropy), as discussed in the "PRNGs" section.

TRNGs

TRNGs are based on a few random physical principles, as summarized in Figure 12 and extracted from the HWsecdb [31]. Noise-amplification-based TRNGs [Figure 12(a)] compare the instantaneous amplitude of the noise generated by an on-chip source with its dc value to generate random output bits. This principle has been demonstrated through the amplification of thermal noise generated by resistors [14], [32], silicon-nitride transistors [62], transistors in post-soft breakdown [99], and random telegraph noise [13], [19], [25], [57]. Among the main challenges related to noise-amplification-based TRNGs, the inevitable presence of offset in the amplification and comparison stage determines a shift in the 0/1 bias. Also, several nonwhite noise contributions are inevitably superimposed to thermal noise (e.g., flicker and coupling/substrate/supply noise), leading to temporal correlation between neighboring output bits. Furthermore, the bandwidth (i.e., power budget) of the amplification stage limits the TRNG throughput due to the rapidly increasing correlation between adjacent outputs at frequencies close to the bandwidth. In other words, TRNG performance is dictated mainly by the analog performance of its building blocks.

Metastability-based TRNGs [Figure 12(b)] leverage the high sensitivity to noise of memory elements around the metastable point, generating an output that is noise dependent [35], [36], [53], [55], [87]. This behavior is the opposite of the metastability-based PUFs described in the section "Weak PUF State of the Art and Trends," whose random mismatch is instead leveraged to have consistent metastability resolution to the same (although unpredictable) level. Thus, mismatch in metastability-based TRNGs must be suppressed through foreground [35] or background circuit calibration [36], [53], [55], [87] or output whitening postprocessing, as in Figure 13(a) [66].

Jitter in oscillators has been exploited as another source of dynamic entropy. A popular approach to translate jitter into random bits is based on jittered oscillator sampling, seen in Figure 12(c) [12], [17], [43]. A jittery slow oscillator samples a fast oscillator around its expected transition, generating a high (low) output if the fast oscillator transitioned earlier (later), as in Figure 12(c). Adaptive tuning of the sampling time [12] and output stream compression [17], [43] improve the output quality to the desired entropy level. A variant of that principle is represented by the exploitation of frequency/oscillation collapse [94], [100].

In frequency collapse, transitions are injected at different points of a conventional ring oscillator. The race-through in the propagation of the multiple transitions in the ring is dictated solely by the accumulated random jitter because the same gates are crossed in the ring. The TRNG output is derived from the cycle count at the frequency collapse [100]. The oscillation collapse TRNG [94] uses

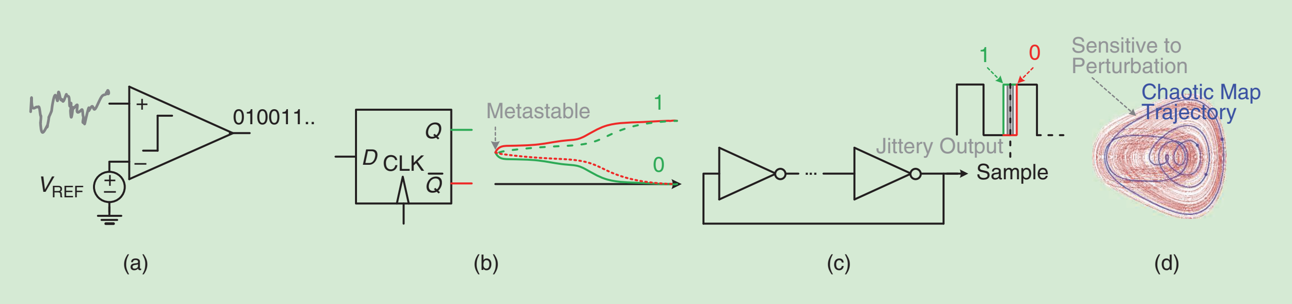


FIGURE 12: The principles used for random number generation: (a) noise amplification, (b) metastability, (c) jitter accumulation, and (d) chaotic maps.

a mechanism similar to the PUF in Figure 11(a) [95], introducing a tuning loop to nearly cancel the effect of variations and so have an output determined mostly by noise (i.e., again, accumulated jitter). One of the available inverter replicas at each ring stage is chosen via multiplexing to equalize the racing paths. Similarly, the phase inversion approach in [40] generates an output according to the time required by two nominally equal oscillators to invert their relative phase due to accumulated jitter. Variations are suppressed through the insertion of proper feedback resistors. Randomness has also been extracted from the random time a transistor takes to experience dielectric (soft) breakdown [48].

As a fourth fundamental class of TRNGs, the strong sensitivity of the trajectory of chaotic maps to perturbation has been widely exploited [see Figure 12(d)]. In analog implementations of chaotic maps, the strong sensitivity of piecewise-linear maps to voltage gains has been mitigated by continuously tuning such gains through an entropy-control feedback loop [5]. Chaotic maps have also been created with data converters in the form of pipelined [67], [68] and subranging successive-approximation-register analog-to-digital converters [38]. Also, the chaotic behavior of resistively/capacitively coupled ring oscillators [23], [24] has been explored in view of their simplicity and mostly digital nature [see Figure 13(b)].

The combination of multiple TRNG classes has been explored in hybrid TRNGs to take advantage of their different sensitivity to environmental conditions and so improve the overall robustness of the generated sequences. For example, noise amplification and jittered oscillator sampling are simultaneously leveraged in [16], whereas thermal noise amplification and chaotic map are combined in [42]. Unified PUFs/TRNGs have also been explored to merge both PUFs and TRNGs in the same circuitry [83] using the most (least) stable PUF bit cells for static (dynamic) randomness.

Finally, post-CMOS technologies have also been explored to generate dynamic randomness. Among the others, the randomness of the write time in spin-transfer torque magnetic RAM devices [97] and the stochastic nature of the state being written at moderate write current level [21] have been explored. Similarly, the variability of the resistance of resistive RAM bit cells has been shown to be suited for dynamic entropy generation [11].

Figure 14(a) summarizes the area and energy trend in TRNGs [31]. The area of TRNGs greatly depends on the considered class, ranging from the area of tens of gates for the smallest metastability- and jitter-based TRNGs to more than 100-k gates for the largest metastability-based and hybrid TRNGs, whose area is dominated by postprocessing circuitry. The area efficiency is improving by 1.4× per year and has become very competitive in TRNGs based on chaotic maps because of the introduction of simpler and mostly digital architectures, such as coupled ring oscillators. In metastability-based TRNGs, area is shrinking at the same pace as the technology allows due to their digital nature. Jitter-based TRNGs are shrinking more slowly than technology scaling due to the increasingly higher cost of calibration and postprocessing because of larger variations in technologies with finer minimum feature size. As shown in Figure 14(b), very

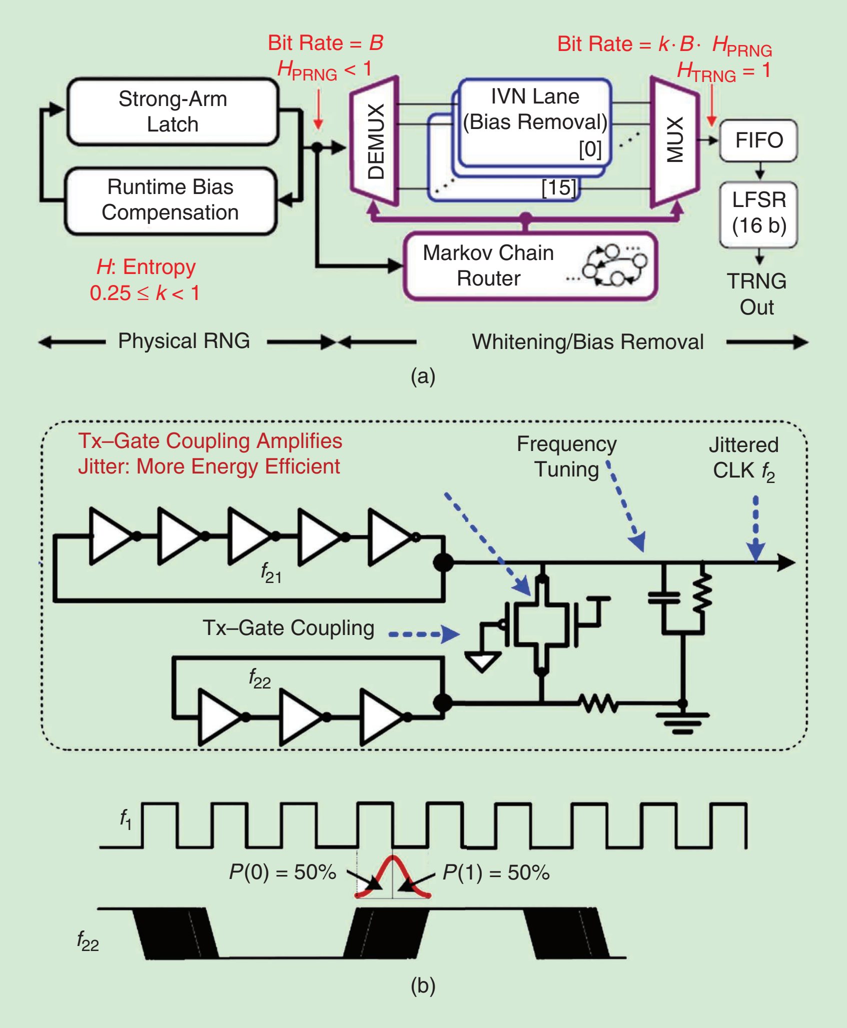


FIGURE 13: Examples of recently proposed TRNGs: (a) metastability based with Markov chain whitening [66] and (b) coupled ring oscillators [23]. LFSR: linear-feedback shift register; Tx: transmitter; FIFO: first in, first out; IVN: iterated Von Neumann; MUX: multiplexer; DEMX: demultiplexer.

rapid (2.36×/year) improvements in energy efficiency are being achieved in chaotic map-based TRNGs because of the adoption of simple, mostly digital architectures. The energy of jitter-based TRNGs is also decreasing faster than technology scaling for the same reason. Representative examples from the recent state of the art are summarized in Figure 15(a). The typical throughput ranges from megabits to gigabits per second, and the energy ranges from picojoules to tens of picojoules (i.e., comparable to strong PUFs and weak PUFs with ECC).

PRNGs

The state-of-the-art PRNGs in Figure 15(b) are significantly more area efficient (by 100–1,000×) and energy efficient (10–100×) than TRNGs, and they have higher throughput (up to 1,000,000×) than TRNGs. A main disadvantage of PRNGs is that their deterministic behavior makes them prone to extrapolation of past values and prediction of future values if the adversary gains knowledge of the state at a given point of time. Also, PRNGs are essentially finite-state machines with a maximum number of states equal to 2^N (assuming the state is represented with N bits), which makes their output intrinsically periodic in the long run. A maximal period of ~2^N cycles is routinely required to guarantee that the adversary is not able to replay output sequences suffering from shorter periods.

Several classes of PRNGs with maximal periodicity have been proposed in the past, among which the simplest is the linear-feedback shift register (LFSR) [74], although it cannot be used for security purposes because its state can be identified at low (linear) complexity from the observation of the output bits. Digital implementations of chaotic maps, such as the Renyi map, have also been demonstrated to have maximal periodicity, a statistical quality passing all randomness tests, and nonlinear behavior preventing cryptoanalysis attacks [6]. Cryptographically secure PRNGs have been widely explored to assure that the effort to retrieve the next state from the output is more difficult than a polynomial, such as Blum Blum Shub, or private-key block ciphers with output being fed back to the input [74].

Postprocessing for RNG Entropy Enhancement

The raw TRNG output invariably needs to be postprocessed by an entropy extractor to mitigate 0/1 bias and correlation and, thus, meet the stringent entropy requirement in cryptokeys (see the section "Entropy Generation"). The entropy extractor relaxes the tight entropy requirement in the raw TRNG as needed due to the fundamental limitations imposed by circuit nonidealities. This comes at the cost of area and energy penalty compared to the raw TRNG. The simplest entropy extractors (e.g., tens of gates) are able to correct the 0/1 bias, but they cannot perform output whitening (i.e., suppressing the correlation between adjacent bits), as summarized in Figure 15(c). Among them, exclusive-OR (XOR) entropy extractors leverage the piling-up lemma [52], guaranteeing power-law improvement in the 0/1 bias when XORing multiple independent streams (e.g., multiple TRNGs) compared to a single one. Von Neumann (VN) entropy extractors decimate the output, generating a 0 (1) when the last two bits from the raw TRNG are 10 (01). VN extractors inhibit their output otherwise to exploit the inherently equal number of rising and falling transitions in a digital signal [19], [67], [88].

As seen in Figure 15(c), entropy extractors performing output whitening and successive bias correction require higher complexity, ranging from hundreds to tens of thousands of gates. Among them, block ciphers (e.g., AES) excited by a raw TRNG are a popular class of entropy extractors that leverage the intrinsic confusion/diffusion properties of ciphers,

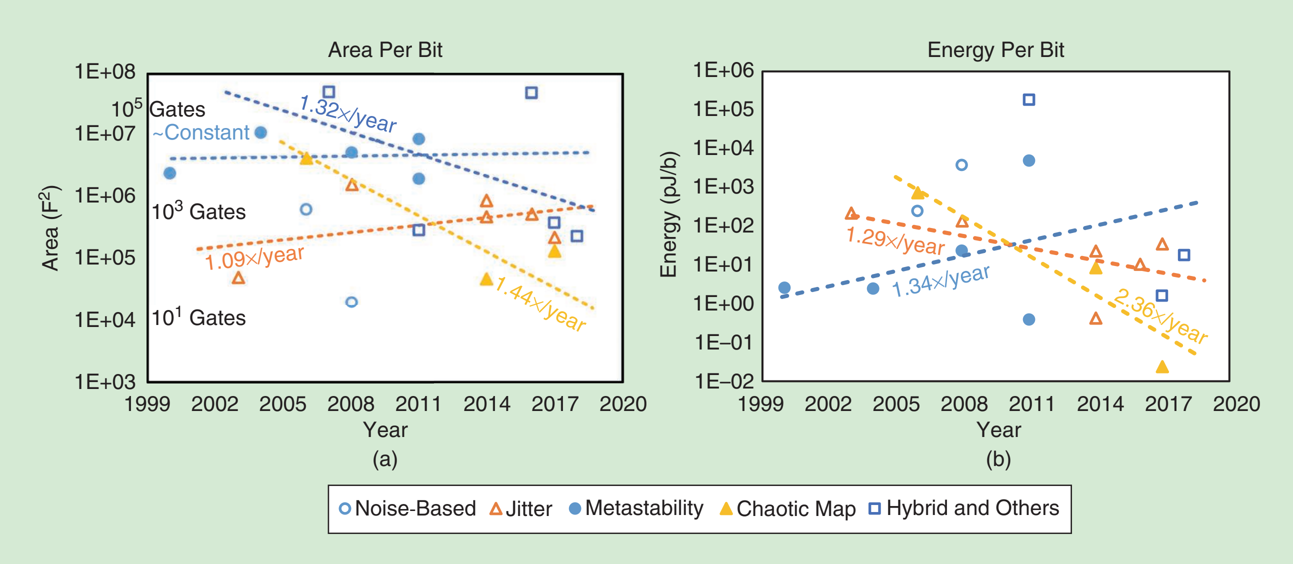


FIGURE 14: TRNG trends in terms of (a) area and (b) energy consumption [31].

which remove bias and correlation from the output bitstream [30], [32], [53]. Considering the performance and overhead of typical ciphers (see the section “Cryptographic Modules”), cipher-based entropy extractors typically degrade throughput by an order of magnitude, the gate count by one to three orders of magnitude, and the energy by more than an order of magnitude.

Barak–Impagliazzo–Wigderson entropy extractors [B06] leverage the confusion/diffusion enabled by basic operations of ciphers (e.g., finite-field addition and multiplications) at a moderate sub-k gate complexity. The Markov chain whitening approach in [66] reduces the correlation across subsequent bits, similar to output subsampling, while reusing the previous bits to have a more favorable throughput-correlation tradeoff. As another important class of entropy extractors, LFSRs and nonlinear-feedback shift registers (NLSRs) preserve throughput and perform whitening [65], [66], [67]. As Figure 15(c) shows, the energy penalty of entropy extractors correcting only the bias is on the order of picojoules to tens of picojoules per bit, and it increases to hundreds of picojoules in extractors performing

	VLSI 2018 [66]	VLSI 2018 [83]	ISSCC 2017 [40]	JSSC 2017 [16]	JSSC 2016 [53]	JSSC 2016 [94]	ISSCC 2014 [100]	
Entropy Source/Architecture	Metastability	Metastability	Jitter	Metastability + Jitter	Metastability	Jitter	Jitter	
Technology	65 nm	14 nm	65 nm	65 nm	14 nm	40 nm	28 nm	
Voltage (V)	0.53–1	0.55–0.75	1.08–1.2	1.2	0.4–0.75	0.6	0.9	①
Throughput (Mb/s)	3.2–86	1,480	8.2–9.9	3,000	8.6–162.5	2	23.16	②
Area (F^2)	$2.37 \cdot 10^6$	$10.8 \cdot 10^6$	$218 \cdot 10^3$	$380 \cdot 10^3$	$5.15 \cdot 10^6$	$519 \cdot 10^3$	$472 \cdot 10^3$	③
Energy (pJ/b)	2.58–6.08	2.5	35.5–42.2	1.6	23	11–23	23	④
NIST Pass	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
Calibration Needed	Yes	No	No	Yes	No	Yes	No	
Entropy Extractor	MC	VN	XOR	No	XOR + BIW	No	No	
Resilient Against V_{DD}	Yes	N/A	Yes	N/A	Yes	Yes	Yes	
Resilient Against Temp.	Yes	N/A	N/A	N/A	Yes	Yes	N/A	

① Digital Allows Low Voltage ② From Mb/s to Gb/s ③ Size: 20–100 μm ④ Energy From Picojoules to Tens of Picojoules

(a)

	Gate Count (Gates)	Energy (pJ/b)	Throughput (Gb/s)
Range	10^1–10^3	10^{-3}–10^{-1}	Up to 10^1
Improvement Over TRNG	10^2–10^3	10^1–10^2	Up to 10^6

(b)

	XOR	VN	BIW	MC	LFSR/NLSR	Block Ciphers
Complexity (Gates)	10^1	10^1	10^2	10^3	10^1–10^4	10^3–10^4
Throughput Reduction	10^0–10^1	Data Dependent	10^1	< 10^1	No	10^1
Energy/bit (pJ)	10^0	10^0–10^1	10^1–10^2		10^{-1}–10^2	10^1–10^2
Bias Correction	Yes	Yes	Yes	No	Yes	Yes
Whitening	No	No	Yes	Yes	Yes	Yes

🟥 Entropy-Extraction Low-Energy Methods Are Comparable to Raw TRNG

🟩 Simultaneous Bias Correction and Whitening Are Energy Hungry

(c)

FIGURE 15: A summary of the recent state of the art in (a) TRNGs, (b) PRNGs, and (c) entropy extraction circuits from TRNGs. MC: Markov chain; BIW: Barak–Impagliazzo–Wigderson.

whitening. Apart from VN and LFSR/NLSR entropy extractors, the aforementioned improvements come with a throughput degradation of up to tens of times compared to the raw TRNG.

Cryptographic Modules

Most fundamental cryptographic operations are based on a few primitives performing private- and public-key cryptography as well as hashing [63]. As common property, such one-way functions are easy to evaluate in one direction and computationally infeasible in the opposite direction. The following discussion summarizes state-of-the-art silicon demonstrations of such functions.

Efficient Primitives for Private-Key Cryptography

The most popular private-key cryptographic algorithm is undoubtedly AES [74], whose best-in-class implementations have a throughput target spanning a very wide range, from hundreds of kilobits per second (e.g., for IoT applications) to tens of gigabits per second (e.g., for server applications), as shown in Figure 16. Subpicojoule/bit operation is now a reality in accelerators for sensor nodes with minimum energy [101], leading to power consumption down to several tens of nanowatts. Affordable even by low-end devices, such as active radio-frequency identification devices with microwatt power budget, subpicojoule/bit operation enables ubiquitous and always-on data security in sensor nodes.

Very area-efficient accelerator architectures (with few k gates) expectedly have poorer energy efficiency, on the order of various picojoules/bits [29], [59], [102]. Compared to low-energy accelerators, the AES software implementation and execution on microcontroller units (MCUs) lead to much worse energy, i.e., hundreds of picojoules/bits and throughput of kilobits per second, leading to submicrowatt power consumption for 128-b AES. On the other end and at the gigabits-per-second throughput range, the energy of general-purpose microprocessors for mobile and desktop applications is in the nanojoule/bit range [89]. High-performance AES accelerators can also achieve such high throughput levels while improving the energy efficiency down to a few picojoules/bits. Compared to microprocessors, field-programmable gate array implementations reduce energy and throughput by an order of magnitude (mostly due to their lower clock frequency).

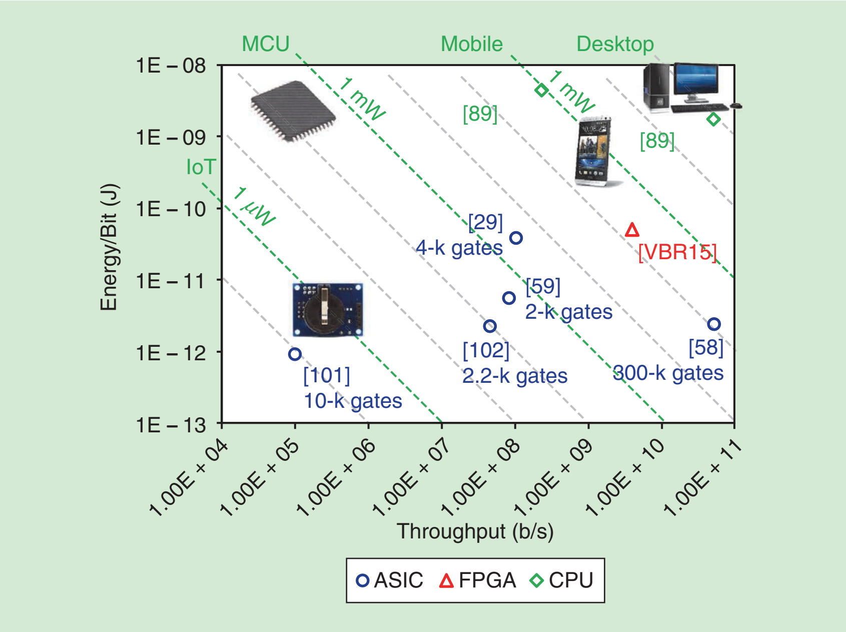


FIGURE 16: The state of the art in AES encryption and energy-throughput tradeoff. FPGA: field-programmable gate array.

The need for ubiquitous and always-on data security has led to a widespread effort to improve energy efficiency by exploring several private-key crypto-algorithms that are inherently less complex than AES, such as Camellia, Prince, Simon, and PRESENT. As shown in Figure 17(a), the complexity of such accelerators for lightweight cryptography ranges from k gates to 20-k gates. Expectedly, various accelerators with subpicojoule/bit energy have been demonstrated for such simpler algorithms [34], [54], [84]. A new breed of algorithms for lightweight cryptography is currently being explored in the NIST "Lightweight Cryptography" project, where secure and affordable solutions for IoT devices are being selected with the ultimate goal of creating a standard [45]. As an emerging trend in accelerators for lightweight cryptography, flexibility is being introduced to enable the different algorithms adopted in different geographical areas. This also enables future upgradeability for security patching and improvements along the chip lifecycle, as needed in IoT devices with a long lifetime.

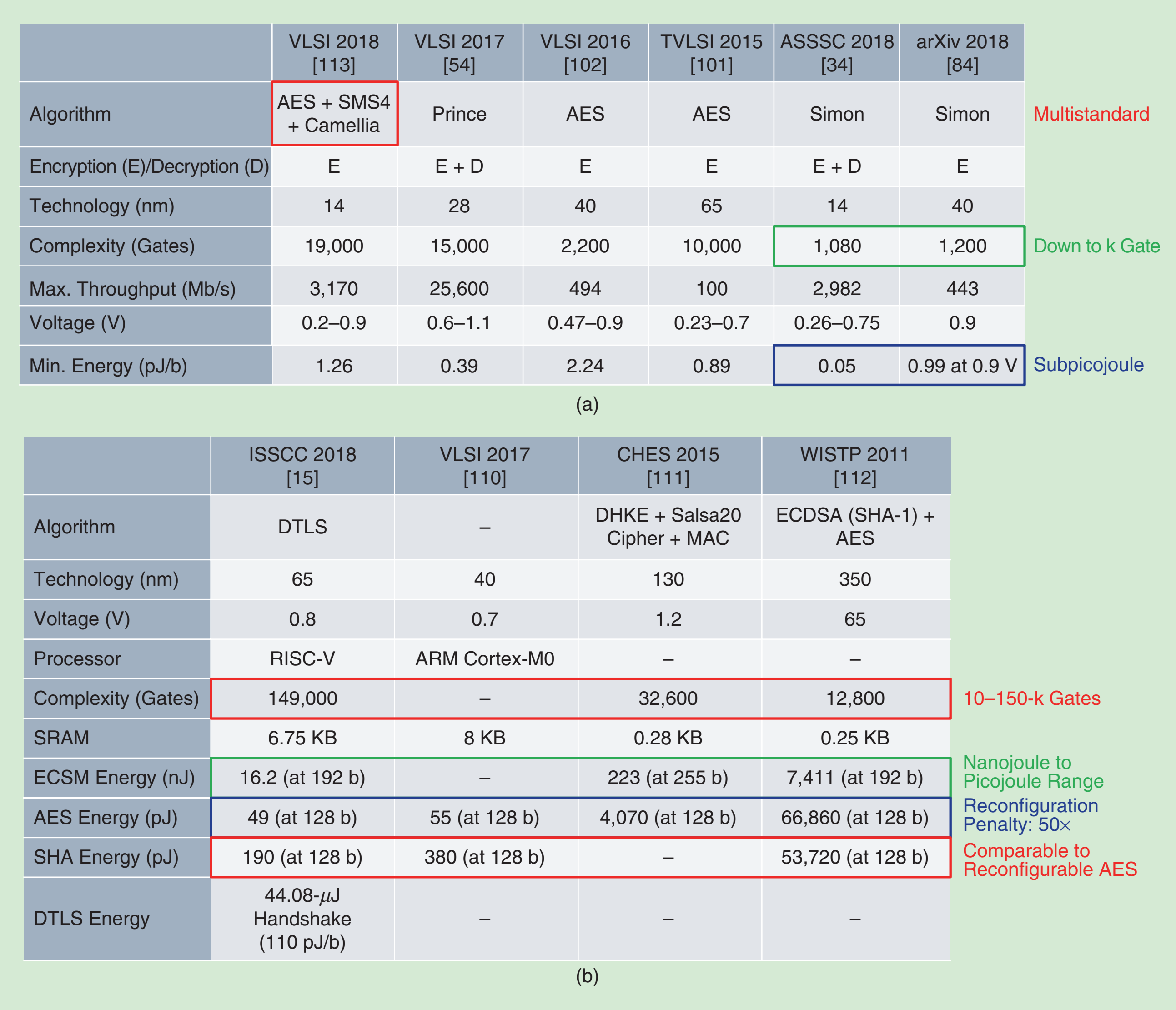
Efficient Primitives for Public-Key Cryptography and Hashing

Public-key cryptography is used in many security tasks, such as private key exchange and digital signature. Public-key cryptography is well known to be vastly more computationally expensive than private key, and this is reflected in the complexity and energy consumption of state-of-the-art designs [15], [110]–[112]. As summarized in Figure 17(b), the flexibility of post-silicon geographical differentiation and upgradeability is also being explored for its use in public-key cryptography. Flexibility is enabled by managing the control flow with a microcontroller or other programmable logic, while performing the elementary functions in energy-efficient accelerators.

As seen in Figure 17(b), the area of public-key cryptocores is higher than those of lightweight private keys by at least an order of magnitude and can be more than 100-k gates. Their flexibility (including private-key operation) comes at the cost of a higher energy by two to four orders of magnitude compared to low-energy private-key accelerators without algorithmic flexibility. Figure 17(b) also shows the substantial energy cost of basic operations used in public-key cryptography. For example, elliptic-curve scalar multiplication entails an energy/bit from tens of nanojoules to microjoules, which is higher than lightweight private-key accelerators by three to six orders of magnitude [15], [44].

Also as shown in Figure 17(b), state-of-the-art accelerators for the Datagram Transport Layer Security protocol used in Internet communications require an energy of tens of microjoules, which is more than 100,000× larger than the energy cost of AES encryption. Similar energy is consumed when executing the widespread Elliptic-Curve Diffie–Hellman Ephemeral private key exchange in MCUs, with an execution time on the order of seconds [44]. An order of magnitude lower energy and execution time is required for Elliptic Curve Digital Signature Algorithms [44]. In summary, public-key cryptography is substantially more expensive than private key and should be used very judiciously, adopting alternative security protocols that leverage less expensive primitives, such as PUFs (e.g., for private key exchange).

Finally, hashing is another fundamental cryptographic function that maps a message of arbitrary length into a digest with fixed length [74]. Energy-efficient hashing has recently gained popularity among the general public, as it is the building block of blockchain mining. In practical applications, mining permits the uncovering of new cryptocurrency coins, and energy-efficient hashing makes it financially more rewarding because of the reduction in electricity cost [1]. As shown in Figure 17(b),



	VLSI 2018 [113]	VLSI 2017 [54]	VLSI 2016 [102]	TVLSI 2015 [101]	ASSSC 2018 [34]	arXiv 2018 [84]	
Algorithm	AES + SMS4 + Camellia	Prince	AES	AES	Simon	Simon	Multistandard
Encryption (E)/Decryption (D)	E	E + D	E	E	E + D	E	
Technology (nm)	14	28	40	65	14	40	
Complexity (Gates)	19,000	15,000	2,200	10,000	1,080	1,200	Down to k Gate
Max. Throughput (Mb/s)	3,170	25,600	494	100	2,982	443	
Voltage (V)	0.2–0.9	0.6–1.1	0.47–0.9	0.23–0.7	0.26–0.75	0.9	
Min. Energy (pJ/b)	1.26	0.39	2.24	0.89	0.05	0.99 at 0.9 V	Subpicojoule

(a)

	ISSCC 2018 [15]	VLSI 2017 [110]	CHES 2015 [111]	WISTP 2011 [112]	
Algorithm	DTLS	–	DHKE + Salsa20 Cipher + MAC	ECDSA (SHA-1) + AES	
Technology (nm)	65	40	130	350	
Voltage (V)	0.8	0.7	1.2	65	
Processor	RISC-V	ARM Cortex-M0	–	–	
Complexity (Gates)	149,000	–	32,600	12,800	10–150-k Gates
SRAM	6.75 KB	8 KB	0.28 KB	0.25 KB	
ECSM Energy (nJ)	16.2 (at 192 b)	–	223 (at 255 b)	7,411 (at 192 b)	Nanojoule to Picojoule Range
AES Energy (pJ)	49 (at 128 b)	55 (at 128 b)	4,070 (at 128 b)	66,860 (at 128 b)	Reconfiguration Penalty: 50×
SHA Energy (pJ)	190 (at 128 b)	380 (at 128 b)	–	53,720 (at 128 b)	Comparable to Reconfigurable AES
DTLS Energy	44.08-μJ Handshake (110 pJ/b)	–	–	–	

(b)

FIGURE 17: A summary of the recent state of the art in lightweight accelerators for (a) private-key encryption and (b) public-key encryption and hashing. ECDSA: Elliptic Curve Digital Signature Algorithms; DTLS: Datagram Transport Layer Security; SHA: Secure Hash Algorithm; ECSM: elliptic curve scalar multiplication.

state-of-the-art accelerators executing the popular Secure Hash Algorithm 2 consume energy greater than but still comparable to AES.

Conclusions

In this article, the state of the art in primitives for hardware security was reviewed, beginning with the fundamentals and then moving on to silicon solutions for tightly energy- and area-constrained systems. A performance scaling trend perspective was provided by introducing the HWsecdb and the PUFdb maintained by the Green IC group of the Department of Electrical and Computer Engineering at the National University of Singapore. The analysis of PUFs and RNGs revealed relentless (exponential) improvements in area and energy efficiency due to the research effort of our community. In weak PUFs, it was observed that the actual area and energy cost are dictated mostly by the ECC and, thus, the PUF native stability. The most promising approaches are mostly or fully digital in view of their amenability for technology scaling and automated design. The state of the art in strong PUFs is less mature and suffers from significantly worse energy and throughput, leaving considerable room for further innovation. Similar considerations hold for TRNGs, whose area and energy efficiency are exponentially improving, especially in mostly and fully digital solutions (e.g., coupled-ring-oscillator-based chaotic maps).

As another important trend, innovative solutions for lightweight private-key cryptography with subpicojoule/bit energy are being investigated. Area- and energy-efficient solutions adding flexibility for post-silicon geographical differentiation and future upgradeability are also being investigated. The currently large energy cost of flexibility leaves, again, interesting room for further innovation. Public-key cryptography has been shown to entail consumption three to six orders of magnitude larger than that of private-key cryptography, which mandates fundamental rethinking of existing security protocols. In the end, understanding the energy/area implications in tightly constrained systems is crucial not only for the design of SoCs but also for the proper choice of security protocols. Tighter interaction between chip design and protocols is demanded for next-generation IoT devices to simultaneously assure a targeted level of security at a minimum energy/area cost while enabling flexibility for future hardware patching.

Acknowledgments

The support of the Singapore National Research Foundation (SOCure grant NRF2018NCR-NCR002-0001) is acknowledged, as well the very valuable inputs and insights from Sachin Taneja and Anastacia Alvarez from the Green IC group of the Department of Electrical and Computer Engineering at the National University of Singapore.

References

[1] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. Sebastopol, CA: O'Reilly Media, 2014.

[2] M. Alioto, Ed., *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*. Cham, Switzerland: Springer, 2017.

[3] M. Alioto, "Perspectives on hardware security: Embedding it everywhere, continuously and inexpensively," YouTube, Apr. 26, 2018. [Online]. Available: https://www.youtube.com/watch?v=44qL_XcAg1c

[4] M. Alioto, "Hardware security: From basics to ASICs," presented at the IEEE Int. Solid-State Circuits Conf., San Francisco, CA, 2019.

[5] T. Addabbo, M. Alioto, A. Fort, S. Rocchi, and V. Vignoli, "A feedback strategy to improve the entropy of a chaos-based random bit generator," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 2, pp. 326–337, 2006.

[6] T. Addabbo, M. Alioto, A. Fort, A. Pasini, S. Rocchi, and V. Vignoli, "A class of maximum-period nonlinear congruential generators derived from the Rényi chaotic map," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 816–828, 2007.

[7] M. Alioto and S. Taneja, "Enabling ubiquitous hardware security via energy-efficient primitives and systems," presented at the IEEE Custom Integrated Circuits Conf., Austin, TX, 2019.

[8] National Agency for Information Systems Security, "Functionality classes and evaluation methodology for physical random number generator," Mar. 23, 2007. [Online]. Available: https://www.ssi.gouv.fr/archive/site_documents/certification/NOTE-05_Evaluation_AIS31_en.pdf

[9] M. N. Aman, S. Taneja, B. Sikdar, K. C. Chua, and M. Alioto, "Token-based security for the Internet of Things with dynamic energy-quality tradeoff," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2843–2859, 2019.

[10] A. Ballesil-Alvarez, W. Zhao, and 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 65nm," *Int. Solid-States Circuits Conf. Dig. Tech. Papers*, pp. 256–258, 2015.

[11] S. Balatti, S. Ambrogio, Z. Wang, and D. Ielmini, "True random number generation by variability of resistive switching in oxide-based devices," *IEEE J. Emerging Select. Topics Circuits Syst.*, vol. 5, no. 2, pp. 214–221, 2015. doi: 10.1109/JETCAS.2015.2426492.

[12] H. Bock, M. Bucci, and R. Luzzi, "An offset-compensated oscillator-based random bit source for security applications," in *Proc. Int. Workshop Cryptographic Hardware and Embedded Systems*, 2004, pp. 268–281.

[13] J. Brown et al., "A low-power and high-speed true random number generator using generated RTN," in *Proc. 2018 IEEE Symp. VLSI Technology*, pp. C95–C96.

[14] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo, "A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC," *IEEE Trans. Comput.*, vol. 52, no. 4, pp. 403–409, 2003.

[15] U. Banerjee, C. Juvekar, A. Wright, Arvind, and A. P. Chandrakasan, "An energy-efficient reconfigurable DTLS cryptographic engine for end-to-end security in IoT applications," in *Proc. 2018 IEEE Int. Solid-State Circuits Conf.*, pp. 42–43.

[16] S.-G. Bae, Y. Kim, Y. Park, and C. Kim, "3-Gb/s high-speed true random number generator using common-mode operating comparator and sampling uncertainty of D flip-flop," *IEEE J. Solid-State Circuits*, vol. 52, no. 2, pp. 605–610, 2017.

[17] M. Bucci and R. Luzzi, "Fully digital random bit generators for cryptographic applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 3, pp. 861–875, 2008.

[18] M. Bhargava and K. Mai, "An efficient reliable PUF-based cryptographic key generator in 65nm CMOS," in *Proc. 2014 Design, Automation & Test in Europe Conf. Exhibition*, pp. 1–6.

[19] R. Brederlow, R. Prakash, C. Paulus, and R. Thewes, "A low-power true random number generator using random telegraph noise of single oxide-traps," *2006 IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pp. 1666–1675, 2006.

[20] Bundesamt für Sicherheit in der Informationstechnik, "Cryptographic mechanisms: Recommendations and key lengths," May 29, 2018. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?_blob=publicationFile&v=8

[21] W. H. Choi et al., "A magnetic tunnel junction based true random number generator with conditional perturb and real-time output probability tracking," in *Proc. 2014 IEEE Int. Electron Devices Meeting*, pp. 12.5.1–12.5.4.

[22] F. Courbon, S. Skorobogatov, and C. Woods, "Reverse engineering Flash EEPROM memories using scanning electron microscopy," in *Proc. 2016 Int. Conf. Smart Card Research and Advanced Applications*, pp. 57–72.

[23] A. T. Do and X. Liu, "25 fJ/bit, 5Mb/s, 0.3 V true random number generator with capacitively-coupled chaos system and dual-edge sampling scheme," in *Proc. 2017 IEEE Asian Solid-State Circuits Conf.*, pp. 61–64.

[24] S. N. Dhanuskodi, A. Vijayakumar, and S. Kundu, "A chaotic ring oscillator based

random number generator," in *Proc. 2014 IEEE Int. Symp. Hardware-Oriented Security and Trust*.

[25] T. Figliolia, P. Julian, G. Tognetti, and A. G. Andreou, "A true random number generator using RTN noise and a sigma delta converter," in *Proc. 2016 IEEE Int. Symp. Circuits and Systems*, pp. 17–20.

[26] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. on Theory of Computing*, 1996, pp. 212–219.

[27] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. 9th ACM Conf. Computer and Communications Security*, 2002, pp. 148–160.

[28] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, 2009.

[29] P. Hämäläinen, T. Alho, M. Hännikäinen, and T. D. Hämäläinen, "Design and implementation of low-area and low-power AES encryption hardware core," in *Proc. 9th EUROMICRO Conf. Digital System Design*, 2006, pp. 577–583.

[30] M. Hamburg, P. Kocher, and M. E. Marson, "Analysis of Intel's Ivy Bridge digital random number generator," Rambus, 2012. [Online]. Available: https://www.rambus.com/wp-content/uploads/2015/08/Intel_TRNG_Report_20120312.pdf

[31] Green IC HWsecdb, "Database of primitives for hardware security." Accessed on: June 1, 2019. [Online]. Available: http://www.green-ic.org/hwsecdb

[32] B. Jun and P. Kocher, "The Intel random number generator," Rambus, Apr. 22, 1999. [Online]. Available: https://www.rambus.com/wp-content/uploads/2015/08/IntelRNG.pdf

[33] S. Jeloka, K. Yang, M. Orshansky, D. Sylvester, D. Blaauw, "A sequence dependent challenge-response PUF using 28nm SRAM 6T bit cell," in *Proc. 2017 Symp. VLSI Circuits*, pp. C270–C271.

[34] H. Kaul et al., "Ultra-lightweight 548–1080 gate 166Gbps/W–12.6 Tbps/W SIMON 32/64 cipher accelerators for IoT in 14nm tri-gate CMOS," in *Proc. 2018 IEEE Asian Solid-State Circuits Conf.*, pp. 183–186.

[35] D. J. Kinniment and E. G. Chester, "Design of an on-chip random number generator using metastability," in *Proc. 28th European Solid-State Circuits Conf.*, 2002, pp. 595–598.

[36] T.-K. Kuan, Y.-H. Chiang, and S.-L. Liu, "A 0.43pJ/bit true random number generator," in *Proc. 2014 IEEE Asian Solid-State Circuits Conf.*, pp. 33–36.

[37] S. S. Kumar, J. Guajardo, R. Maes, G.–J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," *in Proc. 2008 IEEE Int. Workshop Hardware-Oriented Security and Trust.*

[38] M. Kim, U. Ha, K. J. Lee, Y. Lee, and H.-J. Yoo, "A 82-nW chaotic map true random number generator based on a sub-ranging SAR ADC," *IEEE J. Solid State Circuits*, vol. 52, no. 7, pp. 1953–1965, 2017. doi: 10.1109/JSSC.2017.2694833.

[39] O. Kömmerling and M. G. Kuhn, "Design principles for tamper-resistant smartcard processors," in *Proc. USENIX Workshop on Smartcard Technology*, 1999.

[40] E. Kim, M. Lee, J.-K. Kim, "8Mb/s 28Mb/mJ robust true-random-number generator in 65nm CMOS based on differential ring oscillator with feedback resistors," in *Proc. 2017 IEEE Int. Solid-State Circuits Conf.*, pp. 144–145.

[41] W. Killmann and W. Schindler, "A proposal for: Functionality classes for random number generators," Bundesamt für Sicherheit in der Informationstechnik, Sept. 18, 2011. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf

[42] V. von Kaenel and T. Takayanagi, "Dual true random number generators for cryptographic applications embedded on a 200 million device dual CPU SoC," in *Proc. 2007 IEEE Custom Integrated Circuits Conf.*, pp. 269–272.

[43] J. S. Liberty et al., "True hardware random number generation implemented in the 32-nm SOI POWER7+ processor," *IBM J. Res. Develop.*, vol. 57, no. 6, pp. 4:1–4:7, 2013. doi: 10.1147/JRD.2013.2279599.

[44] H. Tschofenig and M. Pegourie-Gonnard, "Performance of state-of-the-art cryptography on ARM-based microprocessors," National Institute of Standards and Technology, 2015. [Online]. Available: https://csrc.nist.gov/csrc/media/events/lightweight-cryptography-workshop-2015/documents/presentations/session7-vincent.pdf

[45] National Institute of Standards and Technology, "Lightweight cryptography." Accessed on: June 1, 2019. [Online]. Available: https://csrc.nist.gov/Projects/Lightweight-Cryptography

[46] 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 for identification and authentication applications," in *Proc. 2004 Symp. VLSI Circuits*, pp. 176–179.

[47] J. Lee, D. Lee, Y. Lee, and Y. Lee, "A 445F^2 leakage-based physically unclonable function with lossless stabilization through remapping for IoT security," *in Proc. 2018 IEEE Int. Solid-State Circuits Conf.*, pp. 132–134.

[48] N. Liu, N. Pinckney, S. Hanson, D. Sylvester, and D. Blaauw, "A true random number generator using time-dependent dielectric breakdown," in *Proc. 2011 Symp. VLSI Circuits*.

[49] J. Li and M. Seok, "Ultra-compact and robust physically unclonable function based on voltage-compensated proportional-to-absolute-temperature voltage generators," *IEEE J. Solid-State Circuits*, vol. 51, no. 9, pp. 2192–2202, 2016.

[50] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications*. Cham, Switzerland: Springer, 2013.

[51] Microsemi Corporation, "PB0115: SmartFusion2 SoC FPGA product brief," Aug. 2018. Accessed on: June 1, 2019. [Online]. Available: https://www.microsemi.com/document-portal/doc_download/132721-pb0115-smartfusion2-soc-fpga-product-brief

[52] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Proc. Workshop on the Theory and Applications of Cryptographic Techniques*, 1994, pp. 386–397.

[53] S. K. Mathew et al., "μRNG: A 300–950 mV, 323 Gbps/W all-digital full-entropy true random number generator in 14 nm FinFET CMOS," *IEEE J. Solid-State Circuits*, vol. 51, no. 7, pp. 1695–1704, 2016. doi: 10.1109/JSSC.2016.2558490.

[54] N. Miura et al., "A 2.5ns-latency 0.39pJ/b 289μm2/Gb/s ultra-light-weight PRINCE cryptographic processor," in *Proc. 2017 Symp. VLSI Circuits*, pp. 256–257.

[55] S. K. Mathew, S. Srinivasan, and M. A. Anders, "2.4 Gbps, 7 mW all-digital PVT-variation tolerant true random number generator for 45 nm CMOS high-performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2807–2821, 2012. doi: 10.1109/JSSC.2012.2217631.

[56] S. K. Mathew et al., "A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS," *2014 IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, vol. 2, no. c, pp. 278–280, 2014.

[57] A. Mohanty, K. B. Sutaria, H. Awano, T. Sato, and Y. Cao, "RTN in scaled transistors for on-chip random seed generation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 8, pp. 2248–2257, 2017. doi: 10.1109/TVLSI.2017.2687762.

[58] S. K. Mathew et al., "53 Gbps nativeGF(2^4)2 composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high-performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 767–776, 2011.

[59] S. Mathew et al., "340 mV–1.1 V, 289 Gbps/W, 2090-gate nanoAES hardware accelerator with area-optimized encrypt/decrypt GF(2 4) 2 Polynomials in 22 nm tri-gate CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 1048–1058, 2015. doi: 10.1109/JSSC.2014.2384039.

[60] S. Satpathy et al., "A 4-fJ/b delay-hardened physically unclonable function circuit with selective bit destabilization in 14-nm trigate CMOS," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 940–949, 2017.

[61] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic PUFs from flip-flops on reconfigurable devices," in *Proc. 3rd Benelux Workshop on Information and System Security*, 2008, pp. 1–17.

[62] M. Matsumoto, S. Yasuda, R. Ohba, K. Ikegami, T. Tanamoto, and S. Fujita, "1200 μm^2 physical random-number generators based on SiN MOSFET for secure smartcard application," *2008 IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pp. 414–415, 2008.

[63] E. Barker and A. Roginsky, "Recommendation for cryptographic key generation," National Institute of Standards and Technology. Accessed on: June 1, 2019. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-133.pdf

[64] D. Nedospasov, J.-P. Seifert, C. Helfmeier, and C. Boit, "Invasive PUF analysis," in *Proc. 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 30–38.

[65] S. Poli, S. Callegari, R. Rovatti, and G. Setti, "Post-processing of data generated by a chaotic pipelined ADC for the robust generation of perfectly random bitstreams," in *Proc. 2004 IEEE Int. Symp. Circuits and Systems*, pp. 585–589.

[66] V. R. Pamula, X. Sun, S. Kim, F. ur Rahman, B. Zhang, and V. S. Sathe, "An all-digital true-random-number generator with integrated de-correlation and bias correction at 3.2-to-86 Mb/s, 2.58 pJ/bit in 65-nm CMOS," in *Proc. 2018 IEEE Symp. VLSI Circuits*, pp. C173–174.

[67] F. Pareschi, G. Setti, and R. Rovatti, "A fast chaos-based true random number generator for cryptographic applications," in *Proc. 2006 Proc. 32nd European Solid-State Circuits Conf.*, pp. 130–133.

[68], F. Pareschi, G. Setti, and R. Rovatti, "Implementation and testing of high-speed CMOS true random number generators based on chaotic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 12, pp. 3124–3137, 2010.

[69] Green IC, "Physically Unclonable Function database." Accessed on: June 1, 2019. [Online]. Available: http://www.green-ic.org/pufdb

[70] U. Rührmair et al., "PUF modeling attacks on simulated and silicon data," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[71] S. Rosenblatt et al., "Field tolerant dynamic intrinsic chip ID using 32 nm high-K/metal gate SOI embedded DRAM," *IEEE J. Solid State Circuits*, vol. 48, no. 4, pp. 940–947, 2013. doi: 10.1109/JSSC.2013.2239134.

[72] A. Rukhin et al. "A statistical test suite for random and pseudorandom number generators for cryptographic applications," National Institute of Standards and Technology, 2010. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf

[73] NXP Semiconductor, "NXP secure microcontroller SmartMX2 P60-Step-Up!" Jan. 2016. [Online]. Available: https://www.nxp.com/docs/en/brochure/75017695.pdf

[74] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Cham, Switzerland: Springer, 2010.

[75] R. Renner and S. Wolf, "Smooth Rényi entropy and applications," in *Proc. Int. Symp. Information Theory*, 2004, p. 233.

[76] S. Skorobogatov, "Local heating attacks on flash memory devices," in *Proc. 2009 IEEE Int. Workshop Hardware-Oriented Security and Trust*.

[77] S. Skorobogatov, "Flash memory 'bumping' attacks," in *Proc. Int. Workshop Cryptographic Hardware and Embedded Systems*, 2010, pp 158–172.

[78] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 6th ed. London: Pearson, 2014.

[79] S. Skorobogatov, "How microprobing can attack encrypted memory," in *Proc. 2017 Euromicro Conf. Digital System Design*.

[80] Y. Su, J. Holleman, and B. Otis, "A 1.6pJ/bit 96% stable chip-ID generating circuit using process variations," *2007 IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pp. 406–408, 2007.

[81] S. Satpathy et al., "13fJ/bit probing-resilient 250K PUF array with soft darkbit masking for 1.94% bit-error in 22nm trigate CMOS," in *Proc. 40th European Solid-State Circuits Conf.*, 2014, pp. 239–242.

[82] S. Satpathy et al., "A 4-fJ/b delay-hardened physically unclonable function circuit with selective bit destabilization in 14-nm trigate CMOS," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 940–949, 2017. doi: 10.1109/JSSC.2016.2636859.

[83] S. Satpathy et al., "An all-digital unified static/dynamic entropy generator featuring self-calibrating hierarchical Von Neumann extraction for secure privacy-preserving mutual authentication in IoT mote platforms," in *Proc. 2018 IEEE Symp. VLSI Circuits*, pp. 169–170.

[84] S. Taneja and M. Alioto, Ultra-low power crypto-engine based on Simon 32/64 for energy- and area-constrained integrated systems. 2018. [Online]. Available: https://arxiv.org/abs/1811.08507

[85] S. Taneja and M. Alioto, "Physically unclonable function design margin reduction via in-situ and PVT sensor fusion," in *Proc. European Solid-State Circuits Conf.*, 2019.

[86] S. Taneja, A. B. Alvarez, and M. Alioto, "Fully synthesizable PUF featuring hysteresis and temperature compensation for 3.2% native BER and 1.02 fJ/b in 40nm," *IEEE J. Solid-State Circuits*, vol. 53, no. 10, pp. 2828–2839, 2018. doi: 10.1109/JSSC.2018.2865584.

[87] C. Tokunaga, D. Blaauw, and T. Mudge, "True random number generator with a metastability-based quality control," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 78–85, 2008. doi: 10.1109/JSSC.2007.910965.

[88] J. von Neumann, "Various techniques used in connection with random digits," *J. Res. Natl. Bur. Stand. Appl. Math Series*, vol. 12, pp. 36–38, 1951.

[89] I. Verbauwhede, J. Balasch, S. S. Roy, and A. Van Herrewege, "Circuit challenges from cryptography," *2015 IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2015, pp. 428–429.

[90] Verayo, "Verayo PUF IP," 2013. [Online]. Available: http://www.verayo.com/tech.php

[91] M.-Y. Wu et al., "A PUF scheme using competing oxide rupture with bit error rate approaching zero," in *Proc. 2018 IEEE Int. Solid-State Circuits Conf.* pp. 130–131.

[92] W.-C. Wang, Y. Yona, Y. Wu, S. Diggavi, and P. Gupta, Implementation and analysis of stable PUFs using gate oxide breakdown. 2018. [Online]. Available: https://arxiv.org/abs/1808.01516

[93] T. Xu, J. B. Wendt, and M. Potkonjak, "Matched digital PUFs for low power security in implantable medical devices," in *Proc. 2014 IEEE Int. Conf. Healthcare Informatics*, pp. 33–38.

[94] K. Yang, D. Blaauw, and D. Sylvester, "An all-digital edge racing true random number generator robust against PVT variations," *IEEE J. Solid-State Circuits*, vol. 51, no. 4, pp. 1022–1031, 2016.

[95] K. Yang, Q. Dong, D. Blaauw, and D. Sylvester, "A physically unclonable function with BER <10−8 for robust chip authentication using oscillator collapse in 40nm CMOS," *2015 IEEE Int. Solid-State Circuits Conf. Dig, Tech. Papers*, 2015, pp. 254–255.

[96] K. Yang, D. Blaauw, and D. Sylvester, "Hardware designs for security in ultra-low-power IoT systems: An overview and survey," *IEEE Micro*, vol. 37, no. 6, pp. 72–89, 2017.

[97] K. Yang et al., "A 28nm integrated true random number generator harvesting entropy from MRAM," *in Proc. 2018 IEEE Symp. VLSI Circuits*, pp. C171–172.

[98] X. Xi, H. Zhuang, N. Sun, and M. Orshansky, "Strong subthreshold current array PUF with 265 challenge-response pairs resilient to machine learning attacks in 130nm CMOS," in *Proc. 2017 Symp. VLSI Circuits*, pp. 268–269.

[99] S. Yasuda, H. Satake, T. Tanamoto, R. Ohba, K. Uchida, and S. Fujita, "Physical random number generator based on MOS structure after soft breakdown," *IEEE J. Solid-State Circuits*, vol. 39, no. 8, pp. 1375–1377, 2004.

[100] K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw, and D. Sylvester, "A 23Mb/s 23pJ/b fully synthesized true-random-number generator in 28nm and 65nm CMOS," *2014 IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2014, pp. 280–281.

[101] W. Zhao, Y. Ha, and M. Alioto, "Novel self-body-biasing and statistical design for near-threshold circuits with ultra energy-efficient AES as case study," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 8, pp. 1390–1401, 2015.

[102] Y. Zhang, K. Yang, M. Saligane, D. Blaauw, and D. Sylvester, "A compact 446 Gbps/W AES accelerator for mobile SoC and IoT in 40nm," in *Proc. 2016 IEEE Symp. VLSI Circuits*.

[103] A.-R. Sadeghi and D. Naccache, Eds., *Towards Hardware-Intrinsic Security: Foundations and Practice*. Cham, Switzerland: Springer, 2010

[104] Microsemi Corporation, SmartFusion2 SoC FPGA PB0115 Product Brief, Aliso Viejo, CA, 2017.

[105] A. Alvarez, W. Zhao, and M. Alioto, "Static physically unclonable functions for secure chip identification with 1.9-5.8% native bit instability at 0.6-1 V and 15 fJ/bit in 65nm," *IEEE J. Solid-State Circuits*, vol. 51, no. 3, pp. 763–775, Mar. 2016.

[106] B. D. Choi and D. K. Kim, "Apparatus and method for generating identification key," U.S. Patent 10 032 729, July 24, 2018.

[107] J. Li, T. Yang, M. Yang, P. R. Kinget, and M. Seok, "An area-efficient microprocessor-based SoC with an instruction-cache transformable to an ambient temperature sensor and a physically unclonable function," *IEEE J. Solid-State Circuits*, vol. 53, no. 3, pp. 728–737, 2018

[108] B. Karpinskyy, Y. Lee, Y. Choi, Y. Kim, M. Noh, and S. Lee, "Physically unclonable function for secure key generation with a key error rate of 2E-38 in 45nm smart-card chips," *ISSCC Dig. Tech. Papers*, San Francisco, CA, 2016, pp. 158–160

[109] Y. Cao, C. Q. Liu, C. H. Chang, "A low power diode-clamped inverter-based strong physical unclonable function for robust and lightweight authentication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 11, pp. 3864–3873, Nov. 2018

[110] Y. Zhang et al., "Recryptor: A reconfigurable in-memory cryptographic Cortex-M0 processor for IoT," *Symp. VLSI Circuits*, June 5–8, 2017. doi: 10.23919/VLSIC.2017.8008501.

[111] M. Hutter, J. Schilling, P. Schwabe, and W. Wieser, "NaCl's Crypto_box in Hardware," *Int. Workshop CHES*, 2015, pp. 81–101. doi: 10.1007/978-3-662-48324-4_5.

[112] M. Hutter, M. Feldhofer, and J. Wolkerstorfer, "A cryptographic processor for low-resource devices: Canning ECDSA and AES like sardines," *WISTP LNCS*, vol. 6633, pp. 144–159, 2011

[113] S. Satpathy et al., "220mV-900mV 794/584/754 Gbps/W reconfigurable GF(24)2 AES/SMS4/Camellia symmetric-key cipher accelerator in 14nm Tri-Gate CMOS," *Symp. VLSI Circuits*, June 18–22, 2018. doi: 10.1109/VLSIC.2018.8502262.

About the Author

Massimo Alioto (malioto@ieee.org) is with the Department of Electrical and Computer Engineering at the National University of Singapore, where he leads the Green IC group and the Integrated Circuits and Embedded Systems area. He received his M.Sc. degree in electronics engineering in 1997 and his Ph.D. degree in electrical engineering in 2001, both from the University of Catania, Italy. He has held positions at the University of Siena, Italy; Intel Labs; the University of Michigan, Ann Arbor; the University of California, Berkeley; and EPFL-Lausanne. He has authored more than 270 publications and three books and is editor-in-chief of *IEEE Transactions on Very Large Scale Integration Systems* and a Technical Program Committee member of the IEEE International Solid-State Circuits Conference. His research interests include self-powered wireless integrated systems, widely energy-scalable systems, data-driven integrated systems, and hardware security. He is a Fellow of the IEEE.