# Hardware Designs for Security in Ultra-Low-Power IoT Systems: An Overview and Survey

**This article presents a survey of state-of-the-art hardware designs optimizing the tradeoffs between security, power, and costs in ultra-low-power systems like the Internet of Things. The authors analyze the connections between hardware specs and system demands to bridge the gap between research conducted in different communities. They also identify open problems in designing future ultra-low-power and secure hardware.**

**Kaiyuan Yang**
*Rice University*

**David Blaauw, Dennis Sylvester**
*University of Michigan*

The emergence of the Internet of Things (IoT) and pervasive computing are expected to enable physical things in the world to collect, process, and exchange data over the Internet. The blending of physical and cyber worlds will open up opportunities to revolutionize healthcare, transportation, infrastructure, and manufacturing industries (see Figure 1). The fundamental technology enablers are ubiquitous ultra-low-power (ULP) and ultra-low-cost edge devices equipped with sensors, actuators, computers, and network connectivity. In particular, ULP systems that can operate on batteries or even on harvested energy for years will enable many disruptive applications, such as implanted and wearable medical and fitness devices, environmental monitors for ecosystem study and protection, and industrial applications.

At the heart of all the ubiquitous applications in Figure 1 is a huge amount of personal, sensitive, or confidential data to be processed and transmitted. Therefore, security and privacy issues are among the most important challenges faced by this technology. Securing these ULP systems poses additional difficulties beyond conventional computer system and network security due to strictly limited computing resources, stringent power budgets, severe cost pressures, and the devices' physical accessibility to attackers. Security within ULP systems must be improved and optimized at every system stack. This article focuses on the lower stacks of the system, namely the hardware building blocks.


Published by the IEEE Computer Society
0272-1732/17/$33.00 

The identification and authentication of physical items are among the most fundamental requirements for secure IoT systems. Common examples include RFIDs for supply chain management, smart cards for owner verification, and wireless sensor nodes for secure data transmission. Identification can be established with any form of public identifier, such as physical marks or electronic IDs stored in nonvolatile memory (NVM) devices. Comparatively, authentication is much more demanding; it requires one entity of a protocol (the verifier) to be assured of the claimed identity of the other entity (the prover)—that is, to distinguish genuine physical things from counterfeited ones and prevent impersonation attack in networks. One-way or mutual authentication should be implemented depending on the targeted applications. A common approach to authentication relies on challenge-and-response protocols, in which the verifier asks a question and the prover must provide a valid answer to be authenticated. The questions and answers are agreed on in advance. The most common implementation of such a protocol is based on cryptographic primitives and secret keys. However, implementing these two primitives in IoT devices faces challenges of severe power and cost budgets, as well as physical attacks ranging from direct probing to side-channel monitoring. Therefore, a new security primitive aiming at secure key storage and lightweight authentication, called *physically unclonable function* (PUF), has emerged in recent years. The essential idea of PUF is to employ manufacturing variations as entropy sources to generate a random mapping function unique to each fabricated instance, which was first envisioned with optical and silicon demonstrations in 2002.[1,2] Over the years, however, it has been shown that PUFs face several critical issues related to their reproducibility, physical security, and vulnerability to modeling attacks.

In addition to identification and authentication of IoT devices, the secrecy and integrity of sensitive data being transmitted within the network, usually wirelessly, are also critical. Until now, cryptographic primitives have been the only practical methods to achieve the security requirements. High power consumption is the main challenge to implementing all kinds of cryptographic primitives in IoT devices. Figure 2 shows the energy costs of encryption compared


**Figure 1.** Internet of Things (IoT) applications and their expected market share. (Source: McKinsey Global Institute.)

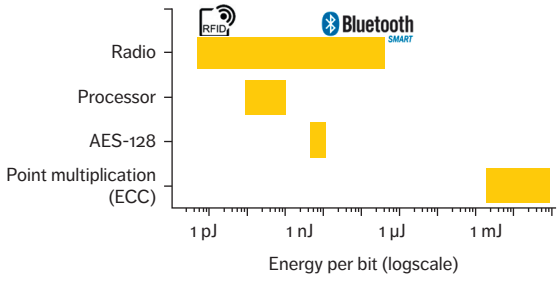


**Figure 2.** Energy efficiency of common IoT building blocks.

to other building blocks of a typical IoT system. Similar to entity authentication, these designs face vulnerability to physical attacks. Attackers can exploit power and electromagnetic (EM) radiation information of the physical implementation to reveal the secret keys.

In this article, we present a survey of the state-of-the-art hardware designs optimizing the tradeoffs between security, power, and costs (including design and manufacturing). Most of the designs have silicon prototypes and measurement results. Open questions in designing future ULP secure hardware are discussed as well.

## Cryptography-Based Entity Authentication

Figure 3 shows common challenge-and-response authentication protocols using cryptographic

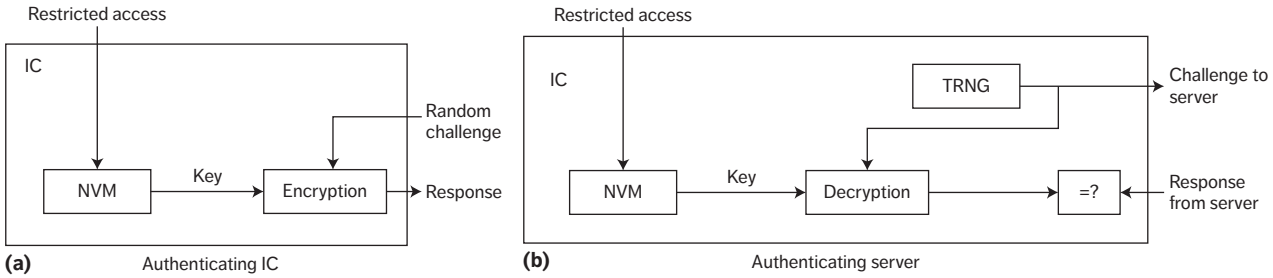


**Figure 3.** Basic authentication protocol using cryptographic primitives.[3]

primitives and secret keys. There are two types of solutions: block ciphers or hash functions with secret keys shared by the verifier and prover, and public-key ciphers with secret keys protected by provers. Both of these require hardware blocks for random number generation, cryptographic computation, and secret key storage. Novel implementations of these blocks are necessary to keep them within power and cost budgets while keeping them secure from potential attacks.

## Secure Crypto Engine

Cryptographic primitives such as block ciphers, public-key ciphers, and hash functions are the most widely used building blocks in secure systems. As Figure 2 shows, running these algorithms in software is not practical for ULP processors in IoT devices, because of the large latency, low energy efficiency, and limited memory space. Therefore, hardware accelerators for ULP devices are critical to reducing overall power consumption, but the available power and area budget limit the use of existing high-throughput and high-efficiency accelerators targeting server applications.[4] Recent research efforts have focused on developing lightweight accelerators for cryptographic algorithms that consume less power and area without the loss of energy efficiency. We selected the most widely used block cipher, Advanced Encryption Standard (AES), for a case study. Many of the design techniques can be adopted by other algorithms as well.

**Lightweight AES engine.** AES is a block cipher working on 128-bit input blocks. It takes 10/12/14 rounds for 128/196/256 key lengths. Within each round, the data is processed in four steps, including AddRoundKey (mixing input data and round key by XOR), SubBytes (nonlinear operation based on an SBox), ShiftRow (cyclically shifting the four rows by 1/2/3/4 bytes), and MixColumn (modular polynomial multiplication with a constant array). Early design efforts focused on high-speed designs using pipelined and loop-unrolled architectures. One of the fastest and most efficient designs was developed by Sanu Mathew and colleagues.[4] However, the specs of such designs are not suitable for ULP systems, and their architectures have delay overhead when used in cipher modes with feedbacks (such as CBC-MAC), which are widely used for authenticated encryption protocols suitable for IoT systems.

One of the earliest lightweight AES engines with complete measurement results and significantly improved power and area costs was presented in 2006.[5] It uses an 8-bit iterative datapath to save area and power, and it employs an SBox calculated in a composite field $GF(2^4)^2$ in runtime, instead of directly storing the $GF(2^8)$ lookup table in the ROM. This technique was first introduced in 2001,[6] and was optimized over the years to achieve better efficiency and a smaller footprint. Almost all recent AES designs have adopted SBox in the composite field, and it is shown to be beneficial to both ULP and high-performance designs. Mathew and colleagues performed an exhaustive search of optimal polynomials to construct the composite field,[7] presenting a 22-nm lightweight AES engine using only 1,947 gates for encryption. The results show that the worst polynomial choice will have 30 percent area overhead compared to the best one. Another innovation is removing the high power and area costs associated with ShiftRow byte permutations.[7] Mathew and colleagues moved this step to the start of each round by rescheduling the input data loaded to the data registers according to the ShiftRow rules.

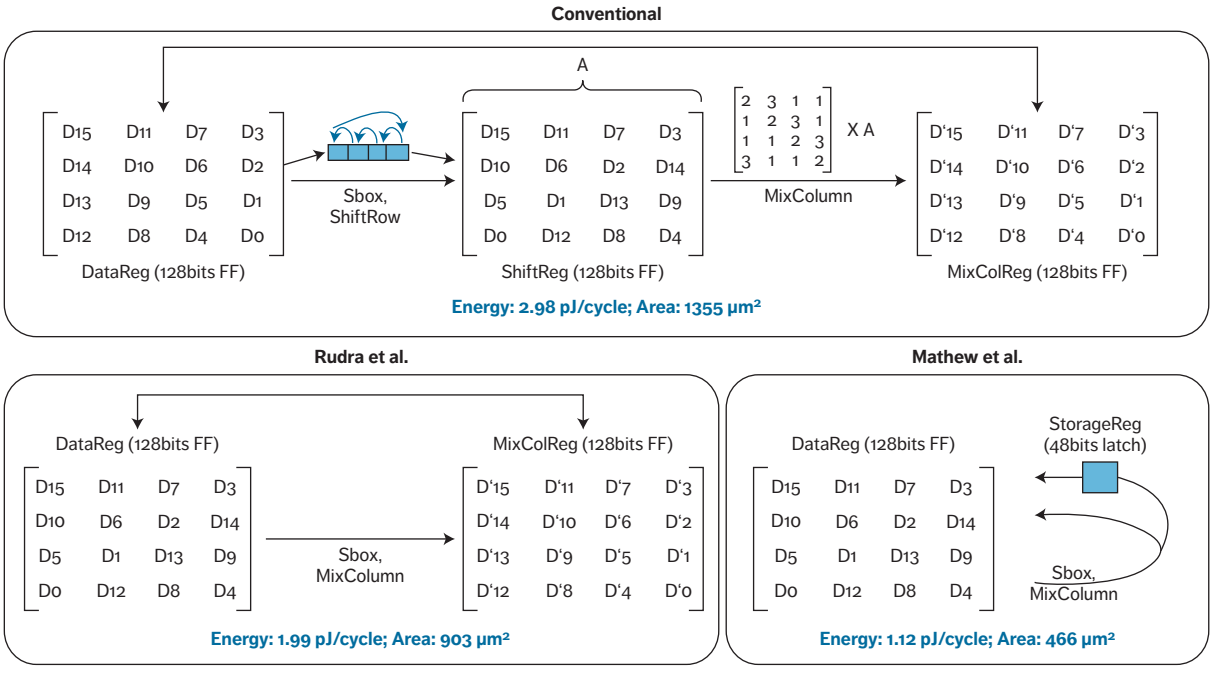


**Figure 4.** Register elimination in the Advanced Encryption Standard (AES) datapath.[8]

Yiqun Zhang and colleagues proposed the latest lightweight AES design,[8] observing that intermediate registers take about 50 percent of total power and area in an 8-bit iterative AES design. They proposed the following techniques to reduce the number of registers (see Figure 4):

- removing ShiftRow, similar to Mathew's work[7];
- reducing MixColumn registers from 128 to 48 bits by rescheduling the data update sequence;
- replacing data, key, and intermediate registers with latches to save area and power; and
- optimizing shift registers by shifting only one-hot addresses instead of all the data registers.

All of these techniques exploit the fixed and known data access patterns of cipher computations and can be applied to other ciphers. Table 1 summarizes the design metrics of the aforementioned AES accelerators.

**Resistance to side-channel attacks.** At the same time, research has shown that physical implementations of secure ciphers can leak information about the secret keys being used for encryption. Researchers have proposed various attack algorithms to break the key from power consumption and EM radiations. Differential power analysis (DPA) is a powerful attack that does not require knowledge about the detailed implementation of the victim hardware.[9] This type of noninvasive attack is a growing concern for IoT devices because adversaries can easily get hold of a device and measure its power and EM information with low-cost devices. This is in contrast to invasive attacks that require expensive equipment to deploy. Therefore, these side-channel attacks are more likely to target low-cost commercial systems that represent the majority of IoT devices.

There are two categories of defense against side-channel attacks: relying on new physical implementations, and using algorithm-level random masking. Although both serve the purpose of reducing the signal-to-noise ratios adversaries can get, they have distinct properties. Physical hiding can be evaluated only heuristically through measurements, but it can be helpful to defend against almost any attack algorithms. On the other hand, algorithmic masking by adding random variables and transforming computations can be provably secure

**Table 1. Performance summary of state-of-the-art lightweight AES accelerator.**

| Design specifications | P. Hamalainen et al. (EUROMICRO 06)[5] | | S. Mathew et al. (JSSC 15)[7] | | Y. Zhang et al. (VLSI 16)[8] | |
|---|---|---|---|---|---|---|
| Technology | 130 nm | | 22 nm | | 40 nm | |
| Voltage (V) | N/A | | 0.9 | 0.43 | 0.9 | 0.47 |
| Power (mW) | 17.98 | 3.9 | 13 | 0.45 | 4.39 | 0.1 |
| Throughput (Mbps) | 232 | 104 | 432 | 83.6 | 494 | 46.2 |
| Efficiency (pJ/b) | 77.5 | 37.5 | 31 | 5.38 | 8.85 | 2.24 |
| No. of gates | 3,200 | 3,900 | 1,947 | | 2,228 | |

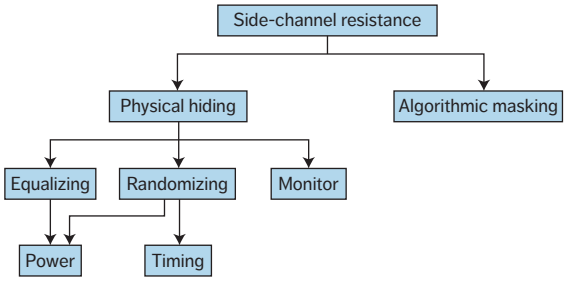


**Figure 5.** Taxonomy of side-channel defenses.

against certain types of attacks, but the determined nature of the masking algorithms makes them potentially vulnerable to higher-order attacks. Figure 5 shows a taxonomy of defenses. We focus on the progress of physical hiding in this article.

Physically hiding side-channel information leakage has been approached in different ways, including using logic gates that consume the same power for different transitions, a power management unit that randomizes the power consumption seen from outside the chip, and on-chip monitors detecting malicious probing. Ideally, the first approach can have the strongest protection against side-channel attacks by closing the source of side-channel information leakage. However, differential logic cells such as Differential Cascode Voltage Switching Logic and even specially optimized logic gates like Sense Amplifier Based Logic[10] cannot fully equalize the current consumptions of different transitions because of parasitics. Therefore, we can evaluate these designs' security levels only in terms of the signal-to-noise ratio, or by the number of measurements to disclosure of the secret keys. These differential logic gates are also more complicated to design with and consume more than twice the energy compared to conventional CMOS logic gates.

To reduce the power consumption, researchers adopted charge-recycling adiabatic logic, which was originally developed for high-efficiency and use-differential logic states, for resistance to side-channel attacks. The concept was first proposed in 2006,[11] and was demonstrated with a complete silicon implementation until 2015.[12] The results show that the adiabatic AES core requires 200+ times more power traces to find the correct key in a DPA attack, and it consumes only 70 percent of the power, compared to a baseline implementation with standard CMOS logic gates. However, the area overhead is about twice that of the baseline area. Thus, adiabatic logic might be the best option for now to physically hide side-channel leakage. The power overhead can be almost negligible, even compared with optimized AES designs, but the area overhead caused by differential cells and off-chip inductors can prevent their application in low-cost devices.

The second category of defense targets equalizing or randomizing the power

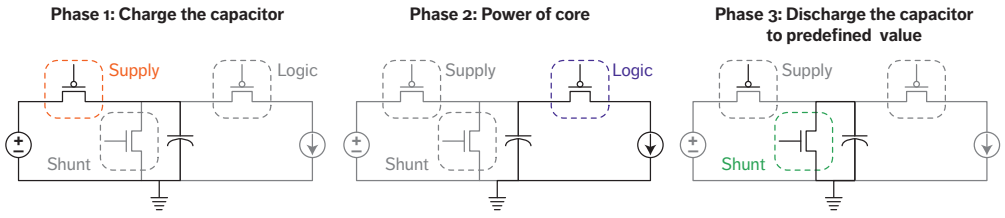


**Figure 6.** Operating principles of switched-capacitor current equalizer.[13]

consumption measured externally. It aims at defending against adversaries with limited resources and motivations to physically probing the chip for side-channel attacks. As discussed previously, this represents the major scenario that a side-channel attack will be carried out. One of the earliest proposals in this direction involves the use of a switched-capacitor current equalizer (see Figure 6).[13] The equalizer has three phases, controlled by closing one of the three switches to recharge the capacitor, supply power, and discharge the capacitor to a predefined level. Three equalizers work in a staggered fashion to ensure continuous operation of the crypto core. As expected, keys are not disclosed even after 10 million measurements, when only the equalizer's power input is exposed to adversaries. This design incurs 33 percent power overhead and 25 percent area overhead to the baseline. To further reduce the power overhead, a recent effort adds a control loop randomization block into an integrated buck voltage regulator to randomize, instead of equalizing the power drawn from the external source. This design adds a mere 5 percent power overhead and 103 gates area overhead to the baseline while being able to resist Correlation Power Analysis (an improved version of conventional DPA attack) and Test Vector Leakage Assessment.[14]

These defenses are effective only against power side-channel attacks. Researchers have shown that EM radiation can leak as much information and EM probes can even collect localized data to reduce noise.[15] To defend against EM attacks, researchers proposed an EM probe monitor based on a LC oscillator implemented on top of a protected circuit.[16] A coil made by top-layer metal is used as a sensor for EM probes. It is based on the observation that EM probes getting close to the chip surface will reduce the inductance of the coil and therefore can be detected by monitoring the frequency of an LC oscillator built with the coil as L. Calibration and referencing techniques are developed for the monitor to detect probes greater than 0.1 mm away from the chip surface. This EM monitor adds 9,000 $\mu m^2$ area overhead and consumes 17 $\mu W$ in 180-nm CMOS.

## Random Number Generation

Random numbers are critical to cryptographic systems to prevent replay attacks and key guesses. Two types of random number generators are widely used: *pseudorandom number generators* (PRNGs), which use a fixed algorithm and initial random seed to generate a sequence of numbers that can be approximated as random numbers; and *true random number generators* (TRNGs), which harvest entropy from physical noise sources, do not require an initial seed, and do not present any periodicity. Although many PRNGs are designed to be indistinguishable by adversaries from a truly random sequence without knowing the input seed, the seed's security and randomness become a concern in IoT devices because of the limited randomness the device can use (such as user input) and the physical accessibility by attackers. On the other hand, the main issues with TRNGs are high power, high cost, and potential vulnerability to external disturbance and attack.

An intuitive approach to on-chip TRNGs is to amplify resistor thermal noise directly and quantize it into digital bits.[17] However, such a design requires several high-performance analog blocks to mitigate nonideal effects (for example, comparator offset or reference variation) that lead to biased, low-entropy outputs. These designs generally consume higher static power, occupy larger area, and have less technology portability. Therefore, recent research

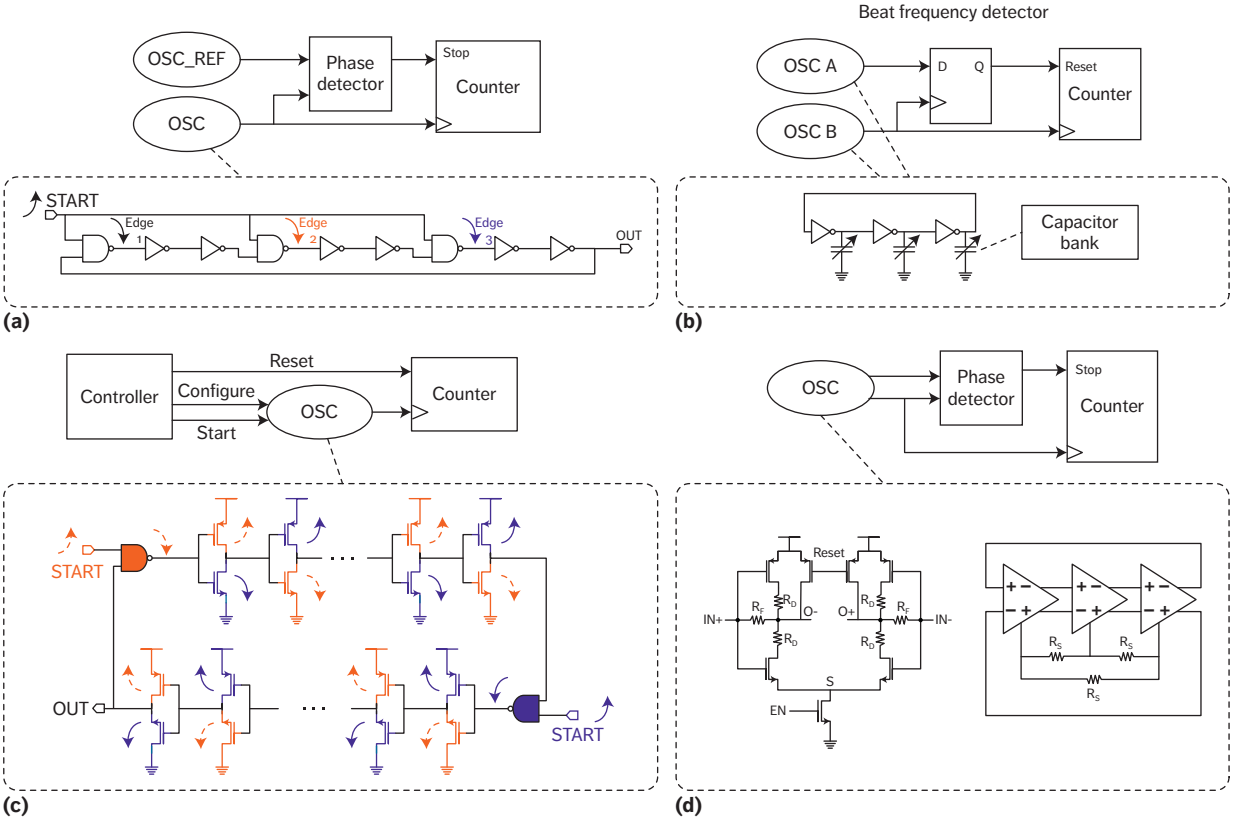


**Figure 7.** Simplified diagrams for state-of-the-art true random number generators (TRNGs), based work by the following authors: (a) K. Yang et al. (ISSCC 14)[23]; (b) Q. Tang et al. (CICC 14)[24]; (c) K. Yang et al. (JSSC 16)[25]; and (d) E. Kim et al. (ISSCC 17).[26]

efforts have focused on digital implementations of TRNG that exploit thermal noise in metastable circuits[18,19] and oscillators.[20] While metastability-based TRNGs offer high speed and efficiency because of the fast transition between metastable and stable states, the transition is easily affected by device variations and environmental conditions so that the generated bits are deterministically biased without complicated postprocessing or calibration steps.[21] Comparatively, oscillator-based TRNGs offer a simpler design to achieve higher raw entropy at the cost of slower speed. They have also been found to be vulnerable to a supply injection attack that injection-locks the oscillator in a TRNG to an external oscillator to reduce its jitter.[22] State-of-the-art TRNGs have focused on improving the speed and efficiency of oscillator-based TRNGs while achieving high-entropy raw outputs[23–26] and providing lightweight quality check and entropy improvement to existing TRNGs.[27]

The keys to further improving TRNGs are to decouple the output bias from process variation and environmental conditions, to automatically stop TRNG operation, and to implement a runtime quality check for TRNGs. This leads to the development of edge-chasing TRNGs that use the phase differences of multiple oscillations in one or multiple oscillators for random number generation. Kaiyuan Yang and colleagues introduced this concept in 2014,[23] in which three edges were injected into the same ring oscillator to oscillate independently, as shown in Figure 7a. In these designs, all three oscillations have exactly the same frequency because they happen in the same physical oscillator, but they have different phases because of their initial phase and independently accumulated noise. The fluctuation of phase differences among the three oscillations is determined by random noise and accumulated over time. Therefore, given enough time, two of the three oscillations will meet and cancel each other

because of their opposite phase. The time it takes for this first-hit-and-collapse event to happen is decided by random noise and therefore used as the proposed TRNG's entropy source. To achieve a uniform distribution of output bits, it has been shown that if the first-hit-and-collapse time is quantized into small enough bins, the least significant bits of the time can be a good approximation to a uniform distribution. Various designs have adopted similar techniques over the years to convert a distribution of time into uniform bits.[23–26] Because this design is not affected by process variations,[23] it can be synthesized with commercial standard cells and placement-and-routing tools (with manually defined rules) and was verified in both 65-nm and 28-nm CMOS technologies. However, this design does not provide runtime quality check and self-tuning capabilities to avoid entropy degradation and denial-of-service attacks caused by supply injection at certain frequencies. The authors proposed to use low-pass filters to protect the TRNG core against supply injection attacks,[23] which can also be applied to most other designs.

A different design with a similar concept was proposed in 2014.[24] The authors used the chasing time of two oscillations in different oscillators with precalibrated small delay difference. As Figure 7b shows, OSC_A is calibrated during start-up to be slightly faster than OSC_B, and they are started simultaneously until OSC_A runs one more cycle and overtakes OSC_B. In this way, the chasing time is bounded within a smaller range, with the average value decided by the deterministic delay difference, so that the TRNG speed is more constant compared to our previous work[23] and provides a knob for tuning the amount of entropy being accumulated.

Yang and colleagues proposed an improved design[25] that combined these previous approaches[23,24] by integrating two oscillations into one even-stage ring oscillator to save area and power. As Figure 7c shows, the two injected edges travel different paths in the even-stage oscillator, which emulates the two oscillators in Tang's work[24] and avoids the complexity of detecting the chase time, because the two oscillations will cancel each other when they meet. In Tang's work,[24] the calibration of the two oscillators is done with capacitor banks, occupying a relatively large area and offering limited resolution. A different calibration technique is employed by Yang and colleagues,[25] who use the intrinsic process variations for fine-grained tuning. Multiple copies of identically designed delay cells are implemented in parallel, and a random search will go over different configurations to find one that falls into the desired operating range. This approach is simpler and requires less area, but can take more trials to set up than the binary search in Tang's work.[24] Yang also provides an analytical model assuming a normal distribution of jitter,[25] which also applies to Tang's design.[24] The chasing of two oscillations is modeled by a random walk with constant drift, and the first-hit-and-collapse time is shown to follow an inverse Gaussian distribution. Mean and variance of the distribution can also be solved by analytical expressions, which helps researchers understand and optimize these designs. These two values are also directly related to the TRNG's operating conditions, which can be used as monitors of the physical random generation process to improve the TRNG's robustness to environmental variations and even deliberate supply injection attacks. Yang and colleagues describe a runtime calibration loop based on these monitors and experimentally verify its robustness against –40 to 120°C and 0.6 to 0.9 V variations.[25] They also show that supply injection attacks can be monitored and thwarted by retuning the oscillator to run at a different frequency.

Eunhwan Kim and colleagues offer the latest improvement aiming at resistance against supply injection attack and simplified startup process (see Figure 7d).[26] They eliminate the precalibration step[25] by using differential delay cells and forcing the differential paths to start with the same phase. The time for the transition from 0° to the normal 180° phase difference is affected by random noise, and the average time is decided by resistor Rs in Figure 7d, similar to previous work.[24,25] The multiple resistors are the key to the robustness against process variations and power injection attack by adding feedback and limiting oscillation amplitude in delay cells. Table 2 provides a summary of the TRNGs.

Although the raw entropy of TRNGs has been significantly improved, postprocessing algorithms to further improve and guarantee

**Table 2. State-of-the-art low-power TRNGs.**

| Design specifications | K. Yang et al. (ISSCC 14)[23] | | Q. Tang et al. (CICC 14)[24] | K. Yang et al. (JSSC 16)[25] | E. Kim et al. (ISSCC 17)[26] |
|---|---|---|---|---|---|
| Technology | 28 nm | 65 nm | 65 nm | 40 nm | 65 nm |
| Voltage (V) | 0.9 | 0.9 | 0.8 | 0.9 | 1.08 |
| Power (mW) | 0.54 | 0.046 | 0.13 | 0.046 | 0.289 |
| Throughput (Mbps) | 23.16 | 2.8 | 2 | 2 | 8.2 |
| Efficiency (pJ/b) | 23 | 57 | 66 | 23 (11 at 0.6 V) | 36 |
| Area ($\mu m^2$) | 375 | 960 | 6,000 | 836 | 920 |
| Operating voltage range | N/A | N/A | 0.8 to 1.2 | 0.6 to 1 | 1.08 to 1.44 |
| Resistance to power injection attack | Need filter | Need filter | N/A | Yes | Yes |
| Pretuning | No | No | Yes | Yes | No |

**Table 3. Comparison of postprocessing methods for TRNG.**

| Design specifications | AES-CBC | SHA-256 | S.K. Mathew et al. (JSSC 16)[27] |
|---|---|---|---|
| Full-entropy throughput | 7 cycles/bit | 7.25 cycles/bit | 8 cycles/bit |
| No. of gates | 32,000 | 19,000 | 4,900 |
| Energy/full-entropy bit | 49 pJ | 34 pJ | 9 pJ |

randomness are of great interest to commercial products to avoid both the potential critical failure of the entropy source and strong physical attacks, and to adopt legacy TRNG designs. According to a National Institute of Standards and Technology recommendation,[28] block ciphers like AES in CBC-MAC mode and certain HMAC functions are ideal for conditioning random bits. For example, the complete TRNG system in Intel CPUs targeting desktop and server applications includes on-chip health and wellness tests and conditioning circuits based on counter-mode AES.[21] However, these designs require significantly larger area and power than the TRNG entropy source core, rendering them unsuitable for ULP IoT and wearable devices.[27] Intel recently developed a lightweight TRNG system employing three independent entropy sources based on metastability and the Barak-Impagliazzo-Wigderson randomness extractor with an 8-bit datapath.[27] The prototype in 14-nm CMOS costs only a fraction of power and area compared to conventional approaches (see Table 3) while maintaining close-to-ideal Shannon entropy and min-entropy across 0.4 to 0.95 V supply variations. The efficiency can be further improved to 3 pJ/bit when operating it at near-threshold 0.4 V.

In summary, a number of novel TRNG designs have been proposed in recent years that achieve better randomness with smaller area, lower power, less complexity, and better design portability. Researchers have also studied intentional attacks such as power supply injection. However, the quality of the generated random bits has been verified only with certain statistical tests; more theoretical analysis and modeling of the designs are expected to facilitate the adoptions of these designs and the development of future low-power, high-entropy TRNGs. At the same time, a plethora of existing randomness extraction algorithms used in cryptography should be studied and optimized for lightweight TRNG conditioning.

## Secret-Key Storage

For authentication and encryption, it is necessary to securely store a digital key on chip. In this section, we describe two types of secret-key storage using nonvolatile memory and PUFs. We present recent progress in the design of both types of key storage.

**Nonvolatile memory.** Conventionally, the secret keys are usually stored in on-chip or stand-alone NVMs, including one-time programmable memory (such as ROM, electronic fuse, and antifuse) and nonvolatile random-access memory (such as electrically erasable programmable read-only memory and flash memory). However, a wide range of invasive (depackaging and probing) and semiinvasive (depackaging only) attacks can be used to read the data stored in these memories. Additionally, most of these memories require extra fabrication steps, which is not desirable for low-cost IoT systems and which cannot scale together with CMOS technologies. In responding to these needs, new memory technologies and designs are introduced for security applications.

In 2017, TSMC reported an antifuse technology using only standard 10-nm FinFET transistors.[29] The data is programmed by gate oxide breakdown during the enrollment phase. Each memory cell comprises just two FinFETs with 0.028 $\mu m^2$ area in 10-nm technology. For side-channel resistance, each bit is stored in two cells with complimentary values so that power consumption during read will not reveal information about the stored keys. This technique also improves the read margin for reliability. This design has overcome most of the security and cost concerns of using conventional NVMs for key storage. The only potential drawbacks are that it cannot destroy its own storage when being tampered, and it can be easily duplicated once a key is exposed. Additionally, it is still vulnerable to certain high-resolution, high-accuracy invasive and semi-invasive attacks, but should be secure enough for most applications.

**Weak PUF for key generation.** In addition to new NVMs, a drastically different key storage method has emerged over the years that relies on hardware-intrinsic process variations to generate and store secret keys. This concept was first proposed for chip identification,[30] but has been renamed as "weak PUFs" as a category of the prevailing PUF concept[2] and increasingly used for security. Because the keys are not stored in digital formats and are sensitive to invasive attacks, they are believed to be more secure than conventional key storage solutions. At the same time, PUFs are completely designed with CMOS transistors so that they can benefit from technology scaling and be easily migrated to different technologies, ranging from cost-sensitive to high-performance applications.

Almost all weak PUFs are designed with a differential structure to generate a response by comparing the characteristics of a differential pair, such as the voltage, current, and delay. Because process variation follows a normal distribution, PUFs are likely to have unreliable responses when the difference between the two arms is small. To solve this problem for security applications, helper data is generated during the initial enrollment phase and is used by the reproduction unit to recover the correct response in subsequent authentication sessions. Jeroen Delvaux and colleagues provide an in-depth overview of helper data algorithms to ensure reproducibility and uniformity of PUF key generation.[31] By replacing NVM in Figure 2 with weak PUF and a reproduction unit, we can achieve authentication using weak PUF. Although error-correction codes are widely used to ensure reliable key generation,[32–34] the information leakage and power and area costs associated with the correction are not negligible. To push the boundary for optimization, researchers have been building

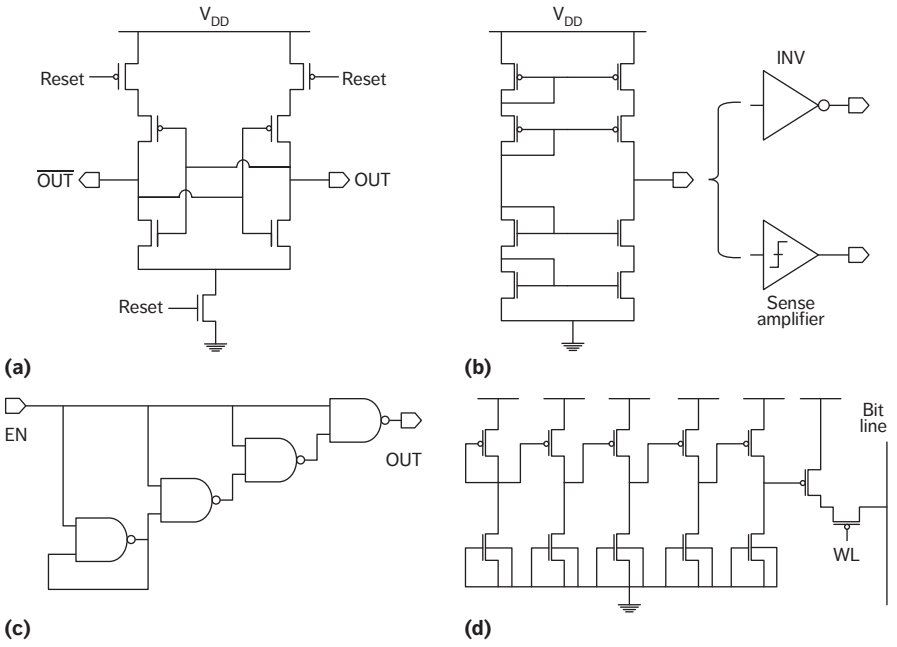


**Figure 8.** Circuit diagrams of state-of-the-art weak physically unclonable functions (PUFs), based on the following work: (a) S. Mathew et al. (ISSCC 14)[33] and Y. Su et al. (JSSC 08)[36]; (b) A. Alvarez et al. (ISSCC 15)[37]; (c) B. Karpinskyy et al. (ISSCC 16)[34]; (d) K. Yang et al. (ISSCC 17).[38]

custom PUF cells that outperform conventional designs using static RAM (SRAM) and oscillators in every aspect. This section presents some state-of-the-art custom PUF cell designs that can potentially alleviate the concerns about unreliable PUF responses.

Output reproducibility across process, voltage, and temperature (PVT) variations and density of the array are two critical metrics directly related to the security and cost of a PUF. The most popular PUF designs use the random power-up state of standard SRAM array[35] because SRAM IPs are widely available and already included in many systems. However, some off-the-shelf SRAMs are biased toward the "1" state,[35] and therefore require postprocessing to improve uniformity. On the basis of the same working principle as SRAM, researchers have designed custom PUF cells based on cross-coupled inverters,[33,36] which include reset switches that bring the structure to a metastable state for evaluation (see Figure 8a). Without further postprocessing techniques, these custom PUF cells do not achieve much better stability (around 6 to 8 percent bit-error rate [BER] at nominal condition) and occupy a larger cell area. However, they don't require manipulation of the power rail and can save a significant amount of power.

Recent custom PUFs with new circuit structures significantly improve the reproducibility and uniformity of native PUF outputs while saving area and power. In 2015, Anastacia Alvarez and colleagues presented a current mirror-based PUF cell (see Figure 8b).[37] Having a static operation and local quantization to PUF output greatly suppresses the noise effects on bit reproducibility, achieving around a 0.3 percent BER (that is, more than 20 times improvement over the SRAM PUF), but at the cost of a large cell. To further reduce area and improve reproducibility, Bohdan Karpinskyy and colleagues introduced a PUF cell using serially connected NAND gates.[34] As Figure 8c shows, the first stage has input and output pins shorted, which forces its output to stay at the transition voltage of the NAND gate. This design does not have an explicit differential structure, but the comparison happens between the transition voltage of the first and second stage. The large gain of the NAND gate around the transition voltage is used to reliably amplify their difference to digital PUF outputs. However, this PUF cell is sensitive to supply voltage and could experience

**Table 4. State-of-the-art weak PUFs.**

| Design specifications | | S. Mathew et al. (ISSCC 14)[33] | Y. Su et al. (JSSC 08)[36] | A. Alvarez et al. (ISSCC 15)[37] | B. Karpinskyy et al. (ISSCC 16)[34] | K. Yang et al. (ISSCC 17)[38] |
|---|---|---|---|---|---|---|
| Technology | | 130 nm | 22 nm | 65 nm | 45 nm | 180 nm |
| PUF cell area/bit ($F^2$) | | 1,092 | 9,628 | 6,036 | 2,613 | 553 |
| Total area/bit ($F^2$) | | 1,767 | N/A | ~36,450 | N/A | 843 |
| Native unstable bits (no. of evaluations) | | N/A | 30% (5,000) | 1.73% (400) | N/A | 1.67% (2,000) |
| Bit-error rates (nominal condition) (%) | | 3.04 | 8.3<br>0.97* | N/A | 0.1† | 0.13 |
| Tested operating conditions | Temperature (°C) | 0 to 80 | 25 to 50 | 25 to 85 | –25 to 85 | –40 to 120 |
| | Supply (V) | 0.9 to 1.2 | 0.7 to 0.9 | 0.7 to 1 | N/A | 0.8 to 1.8 |
| Bit errors per 10°C (%) | | 0.68 | N/A | 0.47 | 0.15 | 0.2 |
| Bit errors per 0.1 V (%) | | 1.82 | 0.49* | 1.27 | N/A | 0.2 |
| PUF core energy (fJ/bit) | | 930 | 13 | 15 | N/A | 11.3 at 1.2 V<br>1.51 at 0.8 V |
| Normal inter-PUF Hamming distance | | 0.506 | ~0.49 | 0.5014 | 0.498 | 0.499 |

** After stabilizing techniques including burn-in, 15-bit temporal majority voting, and dark bits masking. † With 2-bit glitch detector to remove.*

a large short-circuit current during operation. The latest PUF design extends Karpinskyy's ideas by replacing the NAND gates with two-transistor amplifiers (see Figure 8d).[38] This amplifier is biased at the deep subthreshold region to achieve a very high gain (more than 40) with ultra-low power consumption (about 5 pW). Similar to Karpinskyy's work,[34] the difference of switching voltages between the first and second stages is amplified by four cascading two-transistor amplifiers. The designers add two more transistors the same as the 8T SRAM cell's read port for reading, so that the PUF can be arranged in a crossbar array for maximum density, throughput, and efficiency. Table 4 provides a summary of the PUFs' design specs.

Although the advances in PUF cell designs help release the burden on error correction and uniformity, these postprocessing techniques are indispensable. At the same time, researchers have shown that with advanced backside imaging systems and Focused Ion Beams, PUF outputs of 600 nm SRAM cells can be read and edited by adversaries with standard university failure analysis equipment.[39] There is no theoretical barrier to apply the same attack to the more-advanced silicon PUFs mentioned here, although the ultra-low power consumption can potentially increase the requirements on imaging.[37,38] Even though this cloning attack is demonstrated with only 600-nm PUF, researchers should rethink the physical security of PUFs, especially when compared with new NVMs like the 10-nm antifuse discussed earlier. The main difference between the PUFs we studied and NVMs is the former's volatile nature, which is contradictory to reproducibility and intentionally removed in NVMs. For future weak PUF–based solutions, efforts should be made not only on lower costs with reliable operation, but also on capabilities to detect physical attacks actively or passively and

respond to these attacks by using the volatile nature.

## Hardware Designs for PUF-Based Entity Authentication

PUFs are usually categorized into strong and weak PUFs.[40] Both types of PUFs can be modeled as a challenge-and-response function. The difference between them is related to the scalability of the function. Weak PUFs usually have a challenge space linearly related to their area so that only a limited number of challenge-and-response mappings is possible in practice. Strong PUFs, on the other hand, provide a large challenge space that usually exponentially increases with PUF area, and therefore a huge number of mappings can be generated. Because of a large number of random mappings, strong PUFs can be used in challenge-response protocols for authentication as well as for key generation, whereas a weak PUF can be used only for reusable secret keys. This section focuses on using strong PUFs for lightweight authentications.

If the challenge-and-response mappings are truly random, strong PUFs are ideal for secure and lightweight authentication for ULP devices. However, the limited number of random variables in strong PUF circuits and a relatively simple combination of these random variables cannot remove the correlations between different challenge and response pairs. This issue was first envisioned in the original silicon PUF proposal[1] and later proved to be a very effective attack against the most popular arbiter-based strong PUF.[41] Figure 7 shows the arbiter-based PUF,[42] which uses two delay lines with $N$ multiplexers in between to reconfigure the two delay paths. The $N$ inputs to multiplexers are used as challenges to the PUF, and the racing of the two delay paths (judged by an arbiter) is used for PUF outputs. This design proposes an effective method to create an exponential challenge space with limited resources. However, because the challenge-and-response function can be approximated with a linear model, machine learning algorithms like linear regression and evolution strategies can easily find the random variables in PUF with a few challenge-response pairs (CRPs).[41] To defend against modeling attack and overcome other non-idealities in existing strong PUFs, researchers have suggested several authentication protocols using strong PUFs. Delvaux and colleagues present an excellent description and comparison of 19 strong PUF protocols in literature.[3] Eight of them are identified as promising solutions and categorized into two groups. Protocols in the first group work in a similar fashion as weak PUF-based authentication (see Figure 3). Cryptographic computation is still required to improve security, and all the strong PUF versions can be simplified to weak PUF versions. Researchers claim that strong PUFs are more secure against physical attacks, because a modeling attack is required to duplicate the device. This requires a longer attack time, but the strong PUF versions are not theoretically more secure than the weak PUF versions. The burden returns to the strong PUF implementation. The second group of potential protocols (PUF obfuscation) is closer to the original challenge-and-response PUF proposal[2] and keeps the lightweight property by eliminating cryptographic primitives. To satisfy these requirements, slender PUF,[43] noise bifurcation PUF,[44] and lockdown protocol[45] all require the use of a benign model of the strong PUF stored in the server and a TRNG on chip. For these protocols, the design of PUF circuits is even more complicated, because they require modeling by the owner and resistance to modeling attacks. This is possible only by using compound PUFs that have access to the internal simple PUFs during initial enrollment.

As you can see, protocol designs solve only part of the problem; better strong PUF designs are necessary to complement the protocols and help achieve better protection against practical attackers. The fundamental contradiction and tradeoff in strong PUFs are between complexity of the function and reproducibility. When the function is more complicated and nonlinear, a small perturbation of the random variables will significantly change the final response. This seems an impossible mission, because only a combination of them can achieve strong security—for example, a reproducible NVM key and complicated cryptographic primitives (assuming no physical attacks). Fortunately, we can optimize the PUF design in two directions following the two groups of authentication protocols.

### Reproducible but Learnable PUFs

One direction is following the second group by constructing a compound PUF. Researchers have shown both empirically[41] and theoretically[46] that XORing the outputs of enough independent learnable PUFs can make the computing requirement intractable for practical attackers. However, the noise associated with each PUF is accumulated in the XOR PUF, which limits the number of PUFs that can be XORed. According to the accurate modeling of the BER of the XOR PUF (equation 6 in work by Meng-Day Yu and colleagues[45]), it can be approximated as the number of PUFs multiplying the BER of each PUF, when the BER of each PUF is small. Therefore, by reducing the BER of a single PUF to half, the number of PUFs can be doubled in XOR PUF while keeping the same false-acceptance rate and false-rejection rate. In addition, the time and training data required to perform a modeling attack on XOR PUF is growing exponentially with the number of XORs. Thus, the goal here is to improve the BER of a single PUF, not considering the complexity of the mapping function.

A recent progress in this direction replaces the delay lines in an arbiter PUF with a ring oscillator.[47] Delay cells are configurable by input challenges to create a large challenge space. Similar to the edge-chasing TRNG,[25] two edges are inserted to opposite positions of an even-stage oscillator and chase each other. During this process, the mismatch between them is increasing linearly with time, while noise is increasing as a square root function of time. In this way, the PUF's entropy source (process variation) is amplified relative to noise to achieve better reproducibility. In addition, the time for the chasing to finish can be measured by a counter and used to indicate the amount of mismatch between the two edges. This in-situ monitor can accurately monitor the PUF under a varying environment and make decisions about the confidence of this specific CRP. By using it, unreliable responses can be excluded in runtime to significantly reduce the BER. The measurement results of a 40-nm prototype show that the BER can be reduced to less than $10^{-8}$ when 30 percent of the CRPs are discarded.[47] To further lower power, improve efficiency, and improve the BER, delay cells are biased at the near-threshold region. Future work in this direction must carefully consider side-channel attacks, which have been shown to be effective against slender PUF, controlled PUF, and XOR PUFs by using reliability of response and power side channels.[48,49] Although the concern has been alleviated by enforcing the number of accesses to a PUF in lockdown protocol,[45] other defenses are worth investigating.

### Difficult-to-Learn PUFs

The second direction is related to the first group of strong PUF protocols. Improving a single PUF's resistance to modeling attacks will increase the difficulty of attacking these protocols with combined physical and modeling attacks. However, the reproducibility of these new PUFs must be kept low enough to avoid other problems. Two designs targeting modeling-resistant strong PUFs were published in 2017.[50,51] In the former, reconfigurable subthreshold transistors connected in serial and parallel are used to create a nonlinear mapping.[50] In the latter, challenges are changed to sequences of inputs and the PUF is changed from combinational logic to sequential logic for nonlinearity.[51] The design is based on a commercial 6T SRAM array initialized to its power-up value, similar to power-up SRAM PUF.[35] By sequentially shorting different rows, the final state of the last accessed row depends on all the previously accessed rows and the access sequence. By choosing more rows from an array, a large challenge space can be achieved. Both works show similar resistance to linear regression and SVM-based machine learning attacks with up to 10,000 training data, while keeping comparable BERs compared to conventional SRAM and arbiter PUFs.[50,51] These designs are promising but require more rigorous attacks that are designed with knowledge of the PUF.

## Data Security

The demands for data security can be achieved only by cryptographic primitives, including symmetric key ciphers for encryption, hash functions for integrity, and public key ciphers for signature and key exchange. We have discussed these systems' building blocks, including cipher engines, random number generators, and key storage.

For resource-constrained devices, ASIC accelerators can provide the best possible efficiency, power, and area. Many defenses against side-channel attacks are also easier to integrate together with ASIC implementations.

However, one thing to notice is that there will be a wide range of IoT devices and communication standards in the ecosystem, and therefore flexibility of the security algorithm and protocol is important. This represents a different optimization space compared to a pure ASIC design that can exploit fixed operations. Some recent works propose the use of in-memory computing[52] and a flexible-bit-width Galois Field arithmetic logic unit with SIMD instructions[53] to accelerate the most demanding computation in many cryptographic and even error-correction algorithms. They achieve 5 to 20 times improvement over software implementation and within a few times to state-of-the-art ASIC designs.

ULP devices are expected to support a wide range of new and disruptive applications like the IoT. The power and cost budget and physical attack threats demand new hardware and system designs to ensure the security of these devices. In this article, we discussed the need for better hardware blocks to support entity authentication and data security. We presented a survey of recent hardware designs matching these needs in order to show the state of the art. We also identified open problems and future directions for ULP hardware designs for security. We showed that the selection of protocols and hardware design is strongly dependent on specific applications—for example, systems that already require encryption engines are more suitable for weak PUF-based authentication protocols. Also, certain stereotypes about the physical security of PUFs and NVMs need to be reconsidered and studied because of new attacks and defenses. ■

## References


1. B. Gassend et al., "Silicon Physical Random Functions," *Proc. 9th ACM Conf. Computer and Communications Security*, 2002, pp. 148–160.
2. R. Pappu, "Physical One-Way Functions," *Science*, vol. 297, no. 5589, 2002, pp. 2026–2030.
3. J. Delvaux et al., "A Survey on Lightweight Entity Authentication with Strong PUFs," *ACM Computing Surveys*, vol. 48, no. 2, 2015, p. 26:1–26:42.
4. S.K. Mathew et al., "53 Gbps Native Composite-Field AES-Encrypt/Decrypt Accelerator for Content-Protection in 45 nm High-Performance Microprocessors," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, 2011, pp. 767–776.
5. P. Hamalainen et al., "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core," *Proc. 9th EUROMICRO Conf. Digital System Design*, 2006, pp. 577–583.
6. A. Rudra et al., "Efficient Rijndael Encryption Implementation with Composite Field Arithmetic," *Cryptographic Hardware and Embedded Systems*, Ç.K. Koç, D. Naccache, and C. Paar, eds., Springer, 2001, pp. 171–184.
7. S. Mathew et al., "340mV-1.1V, 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, 2015, pp. 1048–1058.
8. Y. Zhang et al., "A Compact 446 Gbps/W AES Accelerator for Mobile SoC and IoT in 40nm," *Proc. IEEE Symp. VLSI Circuits*, 2016.
9. P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Advances in Cryptology*, 1999, pp. 388–397.
10. K. Tiri, M. Akmal, and I. Verbauwhede, "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards," *Proc. 28th European Solid-State Circuits Conf.*, 2002, pp. 403–406.
11. M. Khatir et al., "A Secure and Low-Energy Logic Style using Charge Recovery Approach," *Proc. ACM/IEEE Int'l Symp. Low Power Electronics and Design*, 2008, pp. 259–264.
12. S. Lu, Z. Zhang, and M. Papaefthymiou, "1.32GHz High-Throughput Charge-Recovery AES Core with Resistance to DPA Attacks," *Proc. Symp. VLSI Circuits*, 2015, pp. C246–C247.

13. C. Tokunaga and D. Blaauw, "Secure AES Engine with a Local Switched-Capacitor Current Equalizer," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2009, p. 64–65,65a.
14. B.J. Gilbert Goodwill et al., "A Testing Methodology for Side-Channel Resistance Validation," *Proc. Non-invasive Attack Testing Workshop*, 2011.
15. K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," *Cryptographic Hardware and Embedded Systems*, 2001, pp. 251–261.
16. N. Miura et al., "A Local EM-Analysis Attack Resistant Cryptographic Engine with Fully-Digital Oscillator-Based Tamper-Access Sensor," *Proc. Symp. VLSI Circuits*, 2014.
17. C.S. Petrie and J.A. Connelly, "A Noise-Based IC Random Number Generator for Applications in Cryptography," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, 2000, pp. 615–621.
18. S.K. Mathew et al., "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, 2012, pp. 2807–2821.
19. 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, 2008, pp. 78–85.
20. M. Bucci et al., "A High-Speed Oscillator-Based Truly Random Number Source for Cryptographic Applications on a Smart Card IC," *IEEE Trans. Computers*, vol. 52, no. 4, 2003, pp. 403–409.
21. M. Hamburg, P. Kocher, and M.E. Marson, *Analysis of Intel's Ivy Bridge Digital Random Number Generator*, tech. report, Cryptography Research, 2012.
22. A.T. Markettos and S.W. Moore, "The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators," *Cryptographic Hardware and Embedded Systems*, Springer, 2009, pp. 317–331.
23. K. Yang et al., "A 23Mb/s 23pJ/b Fully Synthesized True-Random-Number Generator in 28nm and 65nm CMOS," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2014, pp. 280–281.
24. Q. Tang et al., "True Random Number Generator Circuits Based on Single- and Multi-phase Beat Frequency Detection," *Proc. IEEE Custom Integrated Circuits Conf.*, 2014, pp. 1–4.
25. 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, 2016, pp. 1022–1031.
26. E. Kim, M. Lee, and J.J. Kim, "8Mb/s 28Mb/mJ Robust True-Random-Number Generator in 65nm CMOS based on Differential Ring Oscillator with Feedback Resistors," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2017, pp. 144–145.
27. 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, 2016, pp. 1695–1704.
28. M.S. Turan et al., *Recommendation for the Entropy Sources Used for Random Bit Generation*, report 800-90B, Nat'l Inst. Standards and Technology, 2016.
29. S.Y. Chou et al., "A 10 nm 32Kb Low-Voltage Logic-Compatible Anti-fuse One-Time-Programmable Memory with Anti-tampering Sensing Scheme," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2017, pp. 200–201.
30. K. Lofstrom, W.R. Daasch, and D. Taylor, "IC Identification Circuit using Device Mismatch," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2000, pp. 372–373.
31. J. Delvaux et al., "Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis," *IEEE Trans. Computer-Aided Design of Integrated Circuits Systems*, vol. 34, no. 6, 2015, pp. 889–902.
32. R. Maes, A.V. Herrewege, and I. Verbauwhede, "PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator," *Cryptographic Hardware and Embedded Systems*, 2012, pp. 302–319.
33. 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," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2014, pp. 278–279.
34. B. Karpinskyy et al., "Physically Unclonable Function for Secure Key Generation with a Key Error Rate of 2E-38 in 45 nm Smart-Card Chips," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2016, pp. 158–160.
35. 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. Computers*, vol. 58, no. 9, 2009, pp. 1198–1210.
36. Y. Su, J. Holleman, and B.P. Otis, "A Digital 1.6 pJ/bit Chip Identification Circuit Using Process Variations," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, 2008, pp. 69–77.
37. A. Alvarez, W. Zhao, and M. Alioto, "15fJ/b Static Physically Unclonable Functions for Secure Chip Identification with <2% Native Bit Instability and 140x Inter/Intra PUF Hamming Distance Separation in 65nm," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2015, pp. 256–257.
38. K. Yang et al., "A 553F$^2$ 2-Transistor Amplifier-Based Physically Unclonable Function (PUF) with 1.67% Native Instability," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2017, pp. 146–147.
39. C. Helfmeier et al., "Cloning Physically Unclonable Functions," *Proc. IEEE Int'l Symp. Hardware-Oriented Security and Trust*, 2013, pp. 1–6.
40. C. Herder et al., "Physical Unclonable Functions and Applications: A Tutorial," *Proc. IEEE*, vol. 102, no. 8, 2014, pp. 1126–1141.
41. U. Rührmair et al., "Modeling Attacks on Physical Unclonable Functions," *Proc. 17th ACM Conf. Computer and Communications Security*, 2010, pp. 237–249.
42. J.W. Lee et al., "A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications," *Proc. Symp. VLSI Circuits*, 2004, pp. 176–179.
43. M. Majzoobi et al., "Slender PUF Protocol: A Lightweight, Robust, and Secure Authentication by Substring Matching," *Proc. IEEE Symp. Security and Privacy*, 2012, pp. 33–44.
44. M.-D. Yu et al., "A Noise Bifurcation Architecture for Linear Additive Physical Functions," *Proc. IEEE Int'l Symp. Hardware-Oriented Security and Trust*, 2014, pp. 124–129.
45. M.D. Yu et al., "A Lockdown Technique to Prevent Machine Learning on PUFs for Lightweight Authentication," *IEEE Trans. Multi-Scale Computer Systems*, vol. 2, no. 3, 2016, pp. 146–159.
46. F. Ganji, S. Tajik, and J.-P. Seifert, "Why Attackers Win: On the Learnability of XOR Arbiter PUFs," *Trust and Trustworthy Computing*, 2015, pp. 22–39.
47. K. Yang et al., "A Physically Unclonable Function with BER <$10^{-8}$ for Robust Chip Authentication using Oscillator Collapse in 40nm CMOS," *Proc. IEEE Int'l Solid-State Circuits Conf.*, 2015, pp. 254–255.
48. G.T. Becker, "On the Pitfalls of Using Arbiter-PUFs as Building Blocks," *IEEE Trans. Computer-Aided Design of Integrated Circuits Systems*, vol. 34, no. 8, 2015, pp. 1295–1307.
49. G.T. Becker, "The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs," *Cryptographic Hardware and Embedded Systems*, 2015, pp. 535–555.
50. X. Xi et al., "Strong Subthreshold Current Array PUF with 265 Challenge-Response Pairs Resilient to Machine Learning Attacks in 130nm CMOS," *Proc. IEEE Symp. VLSI Circuits*, 2017, pp. C268–C269.
51. S. Jeloka et al., "A Sequence Dependent Challenge-Response PUF using 28nm SRAM 6T Bit Cell," *Proc. IEEE Symp. VLSI Circuits*, 2017, pp. C270–C271.
52. Y. Zhang et al., "Recryptor: A Reconfigurable In-Memory Cryptographic Cortex-M0 Processor for IoT," *Proc. IEEE Symp. VLSI Circuits*, 2017, pp. C264–C265.
53. Y. Chen et al., "A Programmable Galois Field Processor for the Internet of Things," *Proc. 44th Ann. Int'l Symp. Computer Architecture*, 2017, pp. 55–68.

**Kaiyuan Yang** is an assistant professor in the Department of Electrical and Computer

Engineering at Rice University. His research interests include energy-efficient integrated circuit and system design and hardware security. Yang received a PhD in electrical engineering from the University of Michigan, Ann Arbor. He is a member of IEEE. Contact him at kyang@rice.edu.

**David Blaauw** is a professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. His research interests include the design of millimeter-scale computing systems and energy-efficient near-threshold computing. Blaauw received a PhD in computer science from the University of Illinois at Urbana-Champaign. He is an IEEE Fellow. Contact him at blaauw@umich.edu.

**Dennis Sylvester** is a professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. His research interests include the design of millimeter-scale computing systems and energy-efficient near-threshold computing. Sylvester received a PhD in electrical engineering from the University of California, Berkeley. He is a cofounder of Ambiq Micro, a fabless semiconductor company developing ultra-low-power mixed-signal solutions for compact wireless devices. He is an IEEE Fellow. Contact him at dmcs@umich.edu.