

## MIT Open Access Articles

*Performance Metrics and Empirical Results  
of a PUF Cryptographic Key Generation ASIC*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** (Mandel) Yu, Meng-Day et al. "Performance Metrics and Empirical Results of a PUF Cryptographic Key Generation ASIC." IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2012. 108–115.

**As Published:** <http://dx.doi.org/10.1109/HST.2012.6224329>

**Publisher:** Institute of Electrical and Electronics Engineers (IEEE)

**Persistent URL:** <http://hdl.handle.net/1721.1/73113>

**Version:** Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

**Terms of use:** Creative Commons Attribution-Noncommercial-Share Alike 3.0



# Performance Metrics and Empirical Results of a PUF Cryptographic Key Generation ASIC

Meng-Day (Mandel) Yu<sup>\*</sup>, Richard Sowell<sup>\*</sup>, Alok Singh<sup>\*</sup>, David M’Raïhi<sup>\*</sup>, Srinivas Devadas<sup>†</sup>

<sup>\*</sup>Verayo, Inc., San Jose, CA, USA  
{myu, rsowell, asingh, david}@verayo.com  
<sup>†</sup>MIT, Cambridge, MA, USA  
devadas@mit.edu

**Abstract**— We describe a PUF design with integrated error correction that is robust to various layout implementations and achieves excellent and consistent results in each of the following four areas: Randomness, Uniqueness, Bias and Stability. 133 PUF devices in 0.13  $\mu\text{m}$  technology encompassing seven circuit layout implementations were tested. The PUF-based key generation design achieved less than 0.58 ppm failure rates with 50%+ stability safety margin. 1.75M error correction blocks ran error-free under worst-case V/T corners ( $\pm 10\%$  V,  $125^\circ\text{C}/65^\circ\text{C}$ ) and under voltage extremes of  $\pm 20\%$  V. All PUF devices demonstrated excellent NIST-random behavior (99 cumulative percentile), a criterion used to qualify random sources for use as keying material for cryptographic-grade applications.

**Keywords** - Physical Unclonable Function (PUF); Error Correction; Key Generation; ASIC; NIST Randomness

## I. INTRODUCTION

### A. Background and Motivation

Physical Unclonable Functions (PUFs) implemented in silicon devices are used to produce output bits that are a function of manufacturing variations. These accumulated bits correspond to a *hardware biometric* signature that can be used to identify silicon devices based on Hamming distance comparisons. Identification is performed by comparing a *regenerated* sequence of PUF output bits on a silicon device against a previously *provisioned* sequence [1, 4, 6-7, 9, 17]. *Inter-class* variation is derived using pair-wise Hamming distance comparisons between two sequences of PUF output bits from different PUF devices. *Intra-class* variation is a measure of the amount of PUF noise present by comparing the Hamming distance between a provisioned sequence of PUF output bits and a regenerated sequence, possibly under a significantly different environmental condition than during provisioning.

In the recent years, there have been numerous works regarding the use of PUFs for *cryptographic key generation* [1-2, 7, 10-13, 16, 18-21]. A *reliability algorithm* is added to the PUF circuit to account for PUF noise, which typically increases with increasing change in environmental conditions (e.g., voltage, temperature) between a provisioning condition, where a reference snapshot of the PUF output bits sequence is taken, and a regeneration condition. While there have been several error correction schemes developed under the assumption of a particular PUF noise model, there are relatively few works where comprehensive PUF key generation results, i.e., ones obtained empirically from a PUF + reliability algorithm

implementation, have been demonstrated under extreme environmental variations.

The current work describes a PUF architecture and reliability algorithm combination robust to various PUF circuit layout implementations, where each PUF circuit produces “random-looking” raw PUF output bits and further these bits can be reliably and efficiently error corrected. We note while it is possible to produce a PUF with random-looking bits, for example by applying bit-wise XOR of two or more manufacturing-variation-derived bits to produce a composite PUF output bit, these PUF bits may not error correct reliably and efficiently. We obtained excellent and consistent empirical results in the areas of *Randomness*, *Uniqueness*, *Bias*, and *Stability* across seven distinct PUF circuit-level implementations (constituting 133 PUF devices), including ones derived using a Standard-Cell ASIC design flow as well as ones derived using a full-custom Custom-Cell ASIC design flow. This helps to accelerate widespread deployment by reducing performance sensitivities associated with the specifics of a particular layout implementation.

### B. Our Contribution

This paper makes the following contributions:

- *PUF circuit-level designs*: Standard-Cell, Custom-Cell, and low-power designs are described that include techniques to reduce area and power.
- *Extensive characterization of PUF key generation ASIC implementation*: Randomness, Uniqueness, Bias, and Stability.
- *PUF architecture + reliability algorithm combination robust to various layout implementations*: Consistent results for 133 PUF devices regardless of layout implementation specifics.
- *Large stability safety margin*: 50%+ unused error correction capacity for 133 PUF devices tested under extreme environmental variations, after an aggregate of 1.75M+ tests.

We present comprehensive experimental results through direct empirical testing of an integrated PUF + reliability algorithm implementation under high environmental variations, including worst-case voltage-temperature corners. We introduce *stability safety margin*, computed as the proportion of error correction capacity *remaining* under some specified environmental conditions, for a certain number of test runs.

This is important to help account for scaling issues associated with a large-scale deployment, to provide a safety margin for very large population sizes, manufacturing skews, aging, radiation damage, etc. We also demonstrate *consistent results* in the areas of Randomness, Uniqueness, Bias, and Stability from a variety of PUF circuit implementation layouts, including Standard-Cell designs as well as Custom-Cell designs. This is important to achieve widespread deployment, as some ASIC design houses may not have Custom-Cell layout and design capabilities. Finally, this work “starves” the power rail of PUF circuits to greatly reduce power consumption while preserving the PUF key generator performance. We note that in the present work, we focus on the description and presentation of results on four of these PUF circuit implementations comprising 76 PUF devices. The results here are representative of the larger data set.

### C. Related Works

Gassend et al. introduced silicon PUFs in [6] [7], where the notion of Inter- and Intra-device variation was used to measure the quality of a silicon PUF circuit for an authentication application. Gassend [7] pioneered the use of error correction with silicon PUFs using a 2-D error correction Hamming code. Suh [18] took a more robust approach to account for environmental noise using a single-stage BCH(255) code. Bösch [2] introduced a two-stage error correction approach. Maes [11-12] introduced the use of soft-decision error correction coding. Yu [20] used an information-theoretically secure Index-Based Syndrome coding approach to achieve robust error correction as an alternative to Code-Offset Syndrome [5] used in the previous approaches [2, 7, 11-12, 18]. Paral [13] used yet another alternative to Code-Offset Syndrome, specifically a pattern matching technique, to derive stable PUF key bits.

TABLE I. PUF KEY GENERATION RESULTS COMPARISON

	<i>Temperature</i>	<i>Voltage</i>	<i>V/T Corners</i>	<i>Stability Safety Margin</i>	<i>NIST STS</i>
Bösch	-20°C to 80°C	n/a	n/a	n/a	n/a
Maes	n/a	n/a	n/a	n/a	n/a
Yu	-55°C to 125°C	±10%	2-corners	50%	n/a
Seli.	-40°C to 80°C	±10%	n/a	24%	n/a
Paral	-25°C to 85°C	n/a	n/a	n/a	n/a
This Work	-65°C to 125°C	±20%	4-corners	50%+	Yes

Table I contains a summary of published PUF key generation results. Bösch [2] developed error correction schemes using a PUF noise profile obtained from [8], which did not explicitly account for voltage effects. Maes [11-12] did not specify the temperature and voltage ranges covered by the PUF noise profile used. Yu [20-21] lacked a 4-corner analysis and NIST randomness data. Selimis in [16] lacked worst-case voltage-temperature analysis, and under single parameter (e.g., temperature only or voltage only) analysis achieved a 24% stability safety margin under the assumption that an error correction scheme can correct up to a quarter of the PUF bits

being noisy (flipped), i.e., a fractional Hamming distance of 0.25, which is the theoretical limit for a conventional single-stage error correction scheme such as a BCH code [3] [14]. Paral [13] lacked voltage data, though the results corresponded to an RFID deriving power from its antenna implying some voltage fluctuation. The current work accounts for a wide temperature range, a wide voltage variation, and contains a 4-corner voltage-temperature (V/T) analysis. Additionally, a 50%+ stability safety margin remains even under these extreme conditions for 1.75M test runs covering 133 PUF devices, illustrating the empirical robustness of the error correction scheme used; there is plenty of error correction capacity or headroom in the error correction algorithm to correct additional errors, e.g., due to factors not explicitly accounted for in these tests. PUF output bit sequences are subjected to extensive NIST Randomness Statistical Test Suite (STS) testing, yielding consistent results for 133 PUF devices.

## II. ARCHITECTURAL CHOICES

### A. An Empirically Robust Combination

The current work analyzes and leverages prior results to derive an empirically robust combination of PUF architecture + reliability algorithm choices. One of the design goals was to derive a PUF key generation design that has a high tolerance to a variety of PUF layout specifics. In this section, we describe the choices made and the rationale.

#### i. PUF Architecture Chosen and Rationale

For the PUF architecture, we chose a k-sum PUF architecture based on [20], shown in Figure 1. This was considered superior to an approach where pair-wise PUF ring oscillators were compared [19], since the summation structure allows PUF oscillator pairs that are closer to each other in frequency (the noisy pairs) to not affect the overall PUF output bit as much due to the summation (averaging) process. The approach was also considered superior to an Arbiter PUF [7] [10] [13] approach in that a traditional Arbiter PUF does not produce “soft-decision” information at the PUF output to indicate the strength of the “1” or the “0” produced. The approach was also considered a better choice than a PUF based on initial SRAM values (memory PUF) [2, 9, 11-12, 17] in that soft-decision information can be readily obtained from (the sum of) oscillator comparisons at a higher resolution.

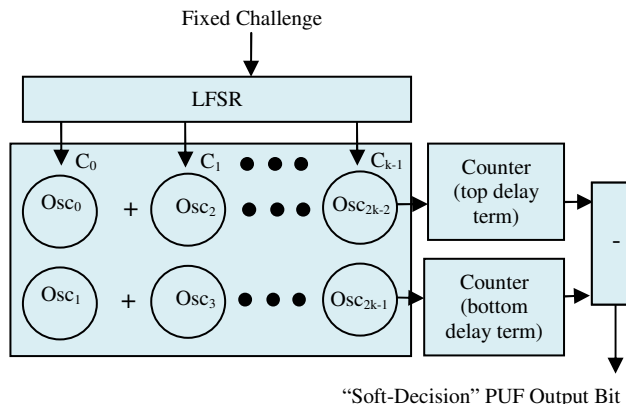


Figure 1: k-sum PUF

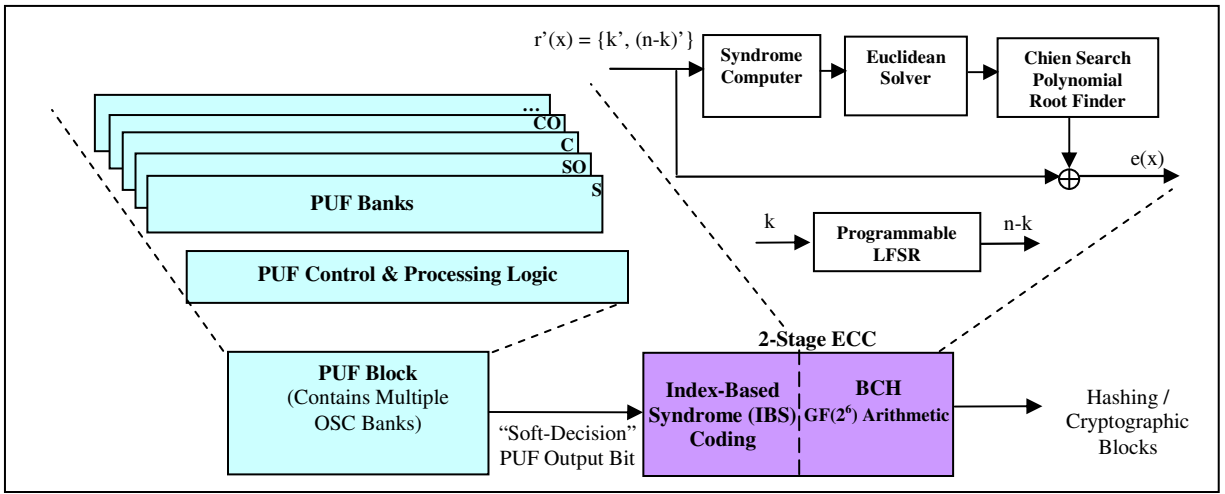


Figure 2: PUF Cryptographic Key Generation ASIC Block Diagram

### ii. Reliability Algorithm Chosen and Rationale

For the reliability algorithm, the two-stage approach in [20], as opposed to the single-stage approach in [21], was chosen, so that an empirical *stability safety margin* measure can be readily derived to help account for unknowns in a large-scale deployment. Index-Based Syndrome (IBS) coding [20] was used instead of Code-Offset [5], to achieve additional coding gains inherent in the index-coding scheme, *on top of* the noise-reduction associated with the oscillator summation (averaging) process. In IBS, an index points to a strong representation of a “1” bit or a “0” bit, where each bit can be either a data bit (the “k” portion of an ECC block) or a parity bit (the “n-k” portion of an ECC block). We use a 4-bit index, selecting the strongest representation out of 16 “Soft Decision” PUF Output Bit choices. We select either the *maximum* difference of top and bottom delay terms (ref: Figure 1), with the difference represented as a *signed* (e.g., a 2’s complement) value, or the *minimum* difference of top and bottom delay terms, depending on whether we want to encode a “1” bit or a “0” bit. For the BCH stage, a BCH ( $n = 63, k = 30, t = 6$ ) code using a three-module architecture, consisting of a Syndrome Computer, Euclidean Solver, and Chien Search finite field factoring algorithm, was used.

As shown in Figure 2, the user selects one or more PUF banks to generate a cryptographic key (four of these PUF bank implementations are presented). A typical PUF bank consists of 128 oscillators and requires 768 NAND2 equivalent gates; shared PUF control and processing logic is used for all the banks. An incremental bank growth can provide for more keying material. The output of the PUF is processed by a two-stage error correction block, using Index-Based Syndrome (IBS) coding and BCH coding [20]. The error correction is empirically robust and provably secure since the hardware algorithms integrated into the ASIC use secure constructs from [20] and [21]. Downstream cryptographic functions, such as hash function and AES, use the PUF-derived key for cryptographic applications.

### B. Circuit-Level Implementations

The ASIC architecture used a “banked” approach to allow different implementation types to be compared. The

implementations include circuit innovations that improved area and power. The simulated power measurements in the typical PVT dropped from 309.7  $\mu\text{W}$  per PUF oscillator (“S”) to 49.9  $\mu\text{W}$  (“CO”), which is approximately a factor of 6.2. These circuits are further described below.

A reference starting design (Figure 3) is the **Standard-Cell PUF (“S”)**, which uses 36 standard-cells, INVX1. These are arranged in a 9x4 configuration, where standard-cell inverter gates are used as dummy loads to slow the oscillation. One of the inversion stages is implemented with a NAND gate to start-stop the oscillator. The oscillator output is buffered to prevent output feedback noise from negatively affecting the oscillation.

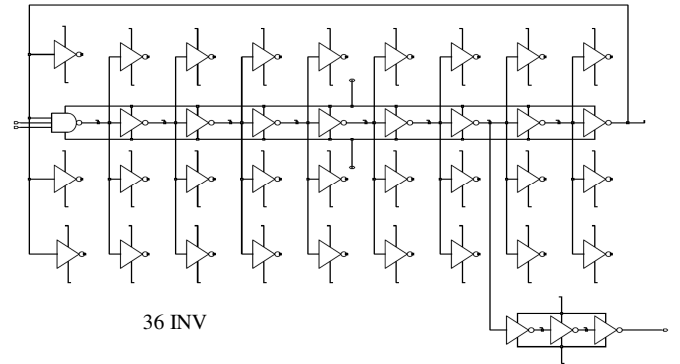


Figure 3: Standard-Cell PUF (“S”)

The reference standard-cell design was optimized for area (and power) as shown in Figure 4. The **Standard-Cell Optimized PUF (“SO”)** uses an innovative technique where standard-cell gates are used to starve the ring oscillator power rails. The standard-cell gates are arranged in cascade, where the output of one standard-cell gate is fed into the power rail of the next. This requires creative manipulation of Standard-Cell ASIC layout tools. The number of inverter equivalents was reduced from a reference value of 36 to 9. Simulation showed a 2.6x reduction in power consumption compared to the reference (“S”) implementation, from 309.7  $\mu\text{W}$  to 116.8  $\mu\text{W}$  per PUF oscillator. A custom level-shifter was designed to give the ring oscillator output a full-range voltage swing. This implementation achieved excellent results in each of the four

categories of testing: Randomness, Uniqueness, Bias, and Stability; the results looked relatively similar to the other implementations, while achieving power/area savings.

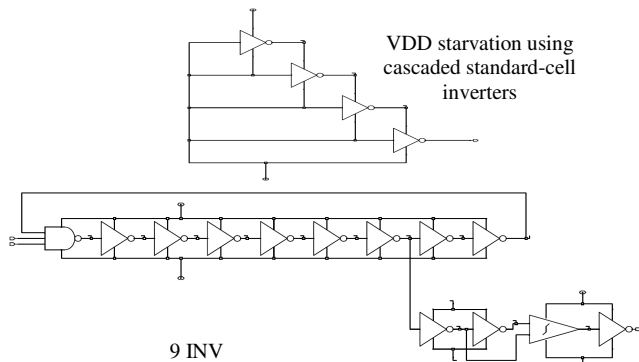


Figure 4: Standard-Cell Optimized PUF (“SO”)

In Figure 5, we move to a full-custom PUF circuit layout implementation approach, to compare the savings. After simulating various topologies, the **Custom-Cell PUF (“C”)** design was derived, using a triple-stacked PMOS/NMOS structure to increase input loading. The simulated power per oscillator is 72.96  $\mu\text{W}$ , which is a 4.2x reduction from the reference “S” implementation.

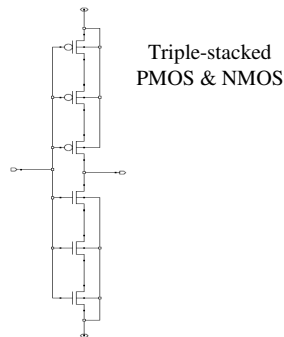


Figure 5: Custom-Cell PUF (“C”)

In Figure 6, we implemented a power/area-optimized variant for the Custom-Cell PUF shown in Figure 5, by adding power rail starvation using a network of pass-gate PMOS transistors. The resulting **Custom-Cell Optimized PUF (“CO”)** achieved a simulated power per PUF oscillator of 49.91  $\mu\text{W}$ , which is a 6.2x reduction from the reference “S” implementation.

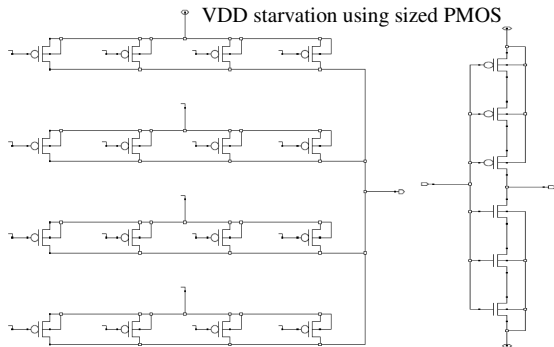


Figure 6: Custom-Cell Optimized PUF (“CO”)

As it will be shown in Section III, the Randomness, Uniqueness, Bias, and Stability results do not vary appreciably across all four implementations.

### III. PERFORMANCE METRICS AND EMPIRICAL RESULTS

Using PUFs as the basis for *cryptographic key generation* requires significant analysis beyond the Intra-class and Inter-class metrics used for PUF-based authentication. It is important to analyze the *Randomness* of PUF output bits in addition to PUF *Uniqueness* (i.e., Inter-class variation). PUF *Bias* (the proportion of 1s in a PUF output bit sequence) needs to be analyzed; a non-negligible bias reduces the guessing entropy of the PUF output bit sequence. Finally, *Stability* of the PUF error correction mechanism needs to be analyzed under various environmental conditions, including worst-case corners. A total of 133 PUF devices, comprising seven PUF circuit layout implementations, each underwent empirical tests in all four areas, namely, *Randomness*, *Uniqueness*, *Bias*, and *Stability*, all producing consistent and excellent results. This section focuses on the presentation of results for 76 PUF devices, encompassing four PUF circuit implementations.

#### A. Randomness

In Figure 7, the rightmost column on the top row (“RND”) represents the NIST pass rate distribution when *NIST-recommended random bits* are applied to the NIST statistical tests for randomness [15]; note that the top of each plot shows the min/max pass rates. The distribution obtained for the four PUF implementations closely resembles the results from “RND”, showing a degree of indistinguishability between PUF randomness test results vs. random bits test results. The bottom row of Figure 7 shows the minimum pass rates for *each* of the 15 NIST tests (92%+) as well as the Cumulative p-values (99.9%+ pass) and Cumulative Proportions (99%+ pass). NIST testing and indistinguishability are common techniques used for entropy source analysis and cryptographic analysis, and we leverage those techniques on PUF output bits from a PUF Key Generation ASIC.

#### B. Uniqueness

For PUF Uniqueness testing (Figure 8), all four PUF implementations produced a Gaussian distribution for the Inter-class variations. Furthermore, as the number of PUF comparisons increases from 34K to 1.1M, obtained by applying more challenges to each PUF device, the  $\mu$  (statistical mean) converges to 0.5, and  $\sigma$  (standard deviation) does not flare out. We are able to show  $\mu$  convergence (based on law of large numbers, and specifically that a sample mean approaches the true mean for random processes as sample size increases) and  $\sigma$  convergence (student-t converges to Gaussian) based on empirical ASIC data as the basis for PUF uniqueness extrapolation to very large sample sizes. One conclusion that one could draw from these results is that as the number of PUF devices and number of PUF response comparisons increase to a very large number, the standard deviation  $\sigma$  should not get worse and flare out, and the statistical mean  $\mu$  should not deviate from an ideal value very close to 0.5, providing a measure of assurance for the uniqueness of PUF-derived values.

### C. Bias

For the PUF bias tests (Figure 9), all four PUF types produce bias distributions that are well within  $\pm 1\%$  of ideal, beyond which the bias becomes cryptographically significant in that NIST randomness tests would likely, and readily, fail. In cryptographic applications, one is concerned not only about average security, but also worst-case security. The *average bias* is within 1% of 1% from ideal (0.50003). The *worst-case bias* is at 0.49698, within half a percent of ideal. For the bottom row of Figure 9, the bias mean is shown on the bottom of each plot corresponding of each of the four PUF implementations, and the worst-case bias value is shown on top.

### D. Stability

For PUF stability testing (Figure 10), all four PUF implementations produced highly stable PUF output bits with 50%+ stability safety margin under all the conditions tested. A total of 1.75M+ error correction blocks (63 bits each) were provisioned under nominal voltage (1.2V) and room temperature (25°C), and subsequently regenerated under voltage extremes of  $\pm 20\%$  V, and also regenerated under the four VT corners of  $\pm 10\%$  V, -65°C/125°C. No error bits were observed after full error correction, and no more than three bits of errors remain after first-stage error correction (IBS decoding).

There are three rows in the figure. The first row corresponds to provisioning under nominal conditions (1.2V, 25°C), and regeneration under a high level of voltage stress of  $\pm 20\%$  V while maintaining the same temperature. The second row corresponds to provisioning under the same nominal conditions, and regeneration under the slow-fast and fast-slow voltage-temperature corners. The third row corresponds to provisioning under the same conditions and regeneration under the slow-slow and fast-fast voltage-temperature corners. We note that six regeneration conditions (two per row) are shown in the row legend. Each plot contains two histograms representing the raw PUF noise, with the fitted Gaussian  $\mu$ ,  $\sigma$  displayed below each plot. The spike at zero bits represents the post error correction result; since all the errors were corrected, this is also the sample size for a particular combination of test condition. The pair of triplets near the top of each plot corresponds to the maximum error bits observed i. at the raw PUF output, ii. after first-stage ECC (IBS decoding), iii. after second-stage ECC (BCH decoding) for each of the two regeneration conditions. The middle number for each triplet indicates how many bits out of the six-bit BCH error correction capability were used, from which a *stability safety margin* (bits of error correction capability unused, out of 6 bits) measurement can be derived.

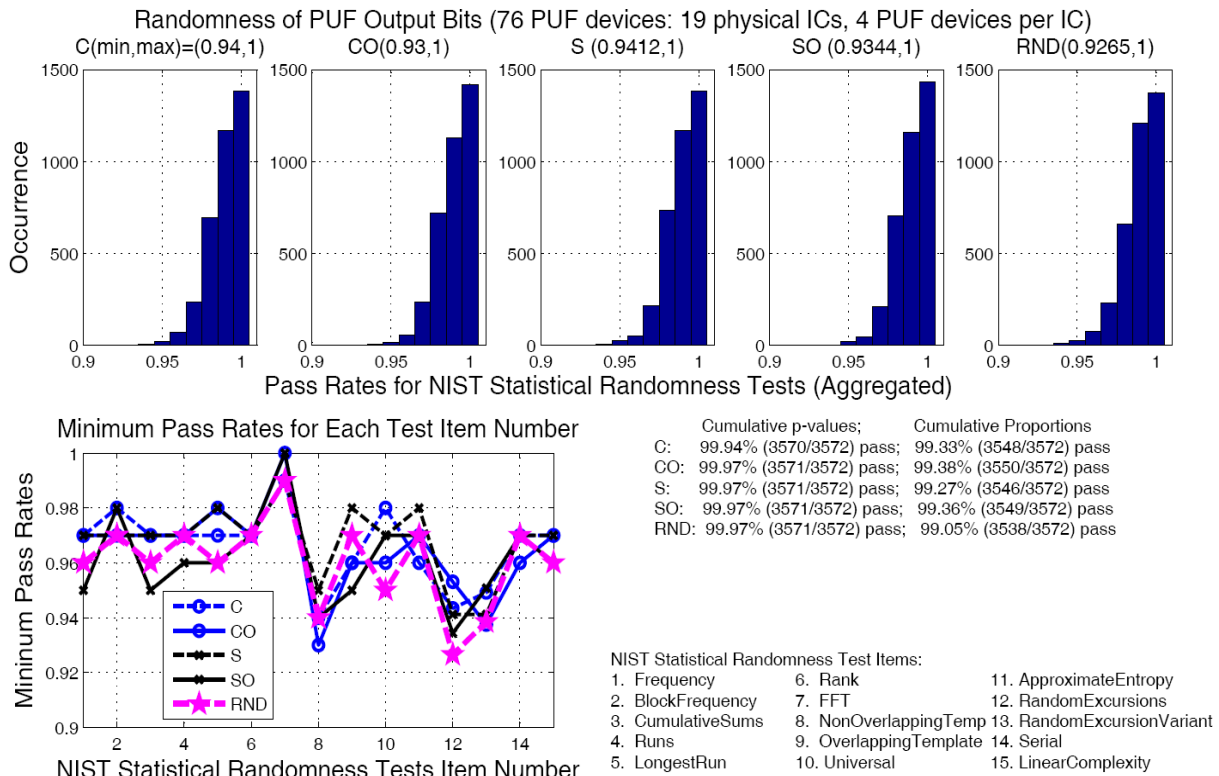


Figure 7: PUF Randomness

Note: The top row shows the NIST randomness test pass rates for the 4 PUF implementations as well as the test results for NIST-recommended random bits (the right most "RND" column). The top of each plot shows the min, max pass rates. The bottom row shows the minimum pass rates for each of the 15 NIST randomness test items.

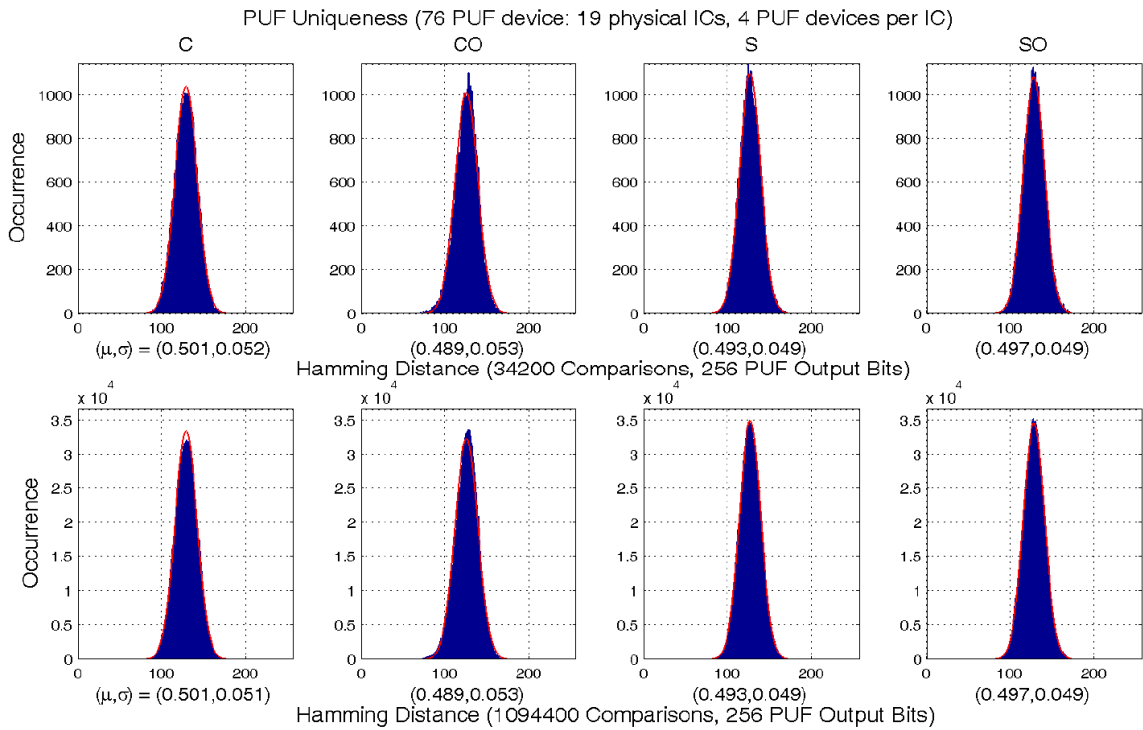


Figure 8: PUF Uniqueness

Note: The top row contains the inter-class PUF distribution for the 4 PUF implementations with 34k comparisons. When the number of comparisons is increased to 1.1M (bottom row), both the  $\mu$  and  $\sigma$  values do not increase and show convergence.

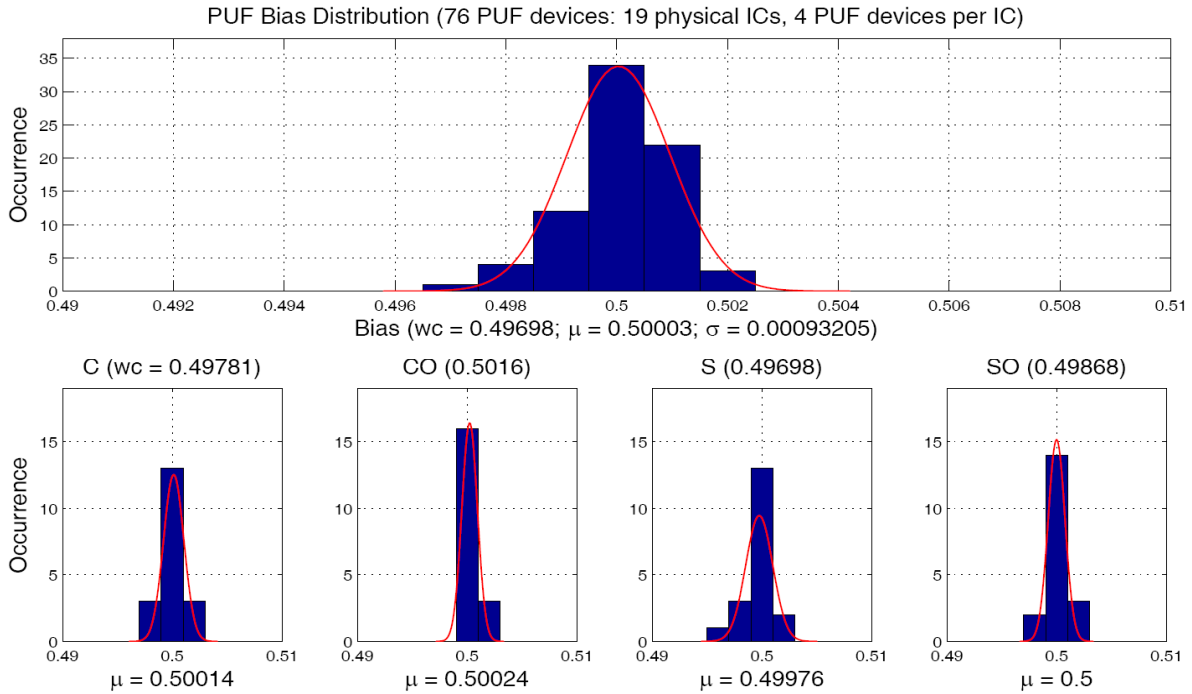
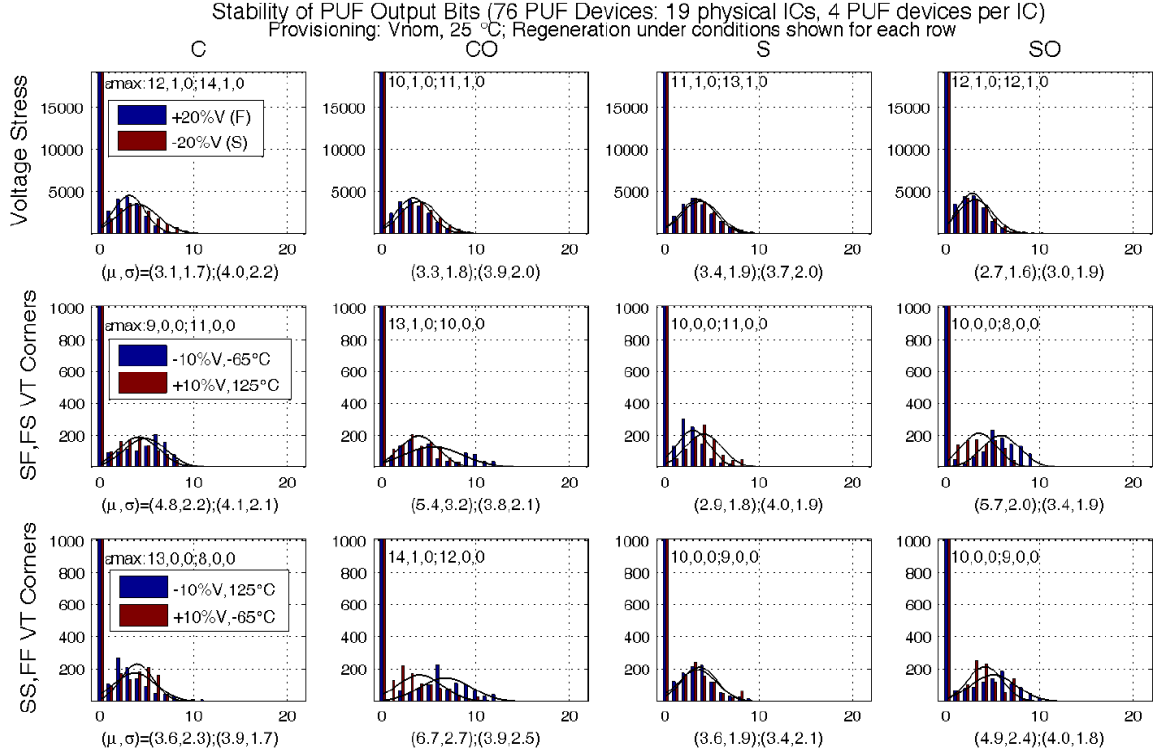


Figure 9: PUF Bias

Note: The top plot contains the PUF bias distributions for 76 PUF devices. The bottom row contains plots for the 4 PUF implementation types, with the bias mean shown at the bottom of each plot, and the worst-case bias value shown on top.



Note: For all cases, provisioning was performed at 1.2V, 25°C. Six regeneration conditions are shown. The top row shows regeneration under +/- 20% voltage. The middle row shows regeneration under fast-slow and slow-fast voltage-temperature corners. The bottom row shows regeneration under slow-slow and fast-fast voltage-temperature corners. Each plot contains two histograms representing the raw PUF noise, with the fitted Gaussian  $\mu$ ,  $\sigma$  displayed below each plot. The spike at zero bits represents the post error correction result; since all the errors were corrected, this is also the sample size. The pair of triplets near the top of each plot are the maximum error bits observed at the raw PUF output, after first-stage ECC (IBS decoding) and after second-stage ECC (BCH decoding) for each of the two regeneration conditions. The error correction block size is 63 bits. The middle number for each triplet indicates how many bits out of the six-bit BCH error correction capability is used, from which a *stability safety margin* (amount of correction capacity unused, out of 6 bits) measurement can be derived.

TABLE II. PUF IMPLEMENTATION COMPARISONS, 0.13 $\mu$ m ASIC

	$Pwr^a$ ( $\mu W$ )	$Area^a$ ( $\mu m^2$ )	<i>NIST Randomness</i>			<i>Uniqueness</i>	<i>Bias</i>	<i>Stability</i>
			<i>min-pass</i> (%)	<i>Cum p-val</i> (%)	<i>Cum prop</i> (%)	<i>mean</i>	<i>wc</i>	<i>margin<sup>b</sup></i> <i>wc noise<sup>c</sup></i>
S	310	200	94.00	99.94	99.33	0.501	0.49781	83% 0.22
SO	117	65	93.00	99.97	99.38	0.489	0.50160	83% 0.22
C	73	60	94.12	99.97	99.27	0.493	0.49698	83% 0.21
CO	50	43	93.44	99.97	99.36	0.497	0.49868	83% 0.19

a. for each oscillator inversion ring b. *stability safety margin* c. worst-case fractional hamming distance before any error correction

#### IV. CONCLUSIONS

We presented empirical PUF key generation test results in the context of a PUF ASIC implementation with integrated

error correction. Specifically, a total of 133 PUF devices comprising seven PUF circuit layout implementations were designed, implemented, and tested. Four metrics specific to PUF key generation were defined, and empirical data was obtained from 0.13  $\mu$ m ASICs. The choice of an oscillator



summation PUF architecture coupled with a two-stage index-based error correction approach produced consistent *Randomness, Uniqueness, Bias, and Stability* results across all these implementations, of which four representatives are summarized in Table II. Since the results are only slightly layout dependent, and the design can in fact be implemented in a Standard-Cell approach (albeit less optimized), we conclude that the PUF key generator design is highly portable across different design methods (Standard vs. Custom Cell) and particular layout choices (power starved vs. none), which eases adoption. The high (50%+) *stability safety margin*, quantifiable-by-design for this particular PUF key generation design, means that excess error correction capability can be relied upon to help account for the unexpected in large-scale deployments. Future work includes explicit analysis of process scaling.

## REFERENCES

- [1] F. Armknect, R. Maes, A.-R. Sadeghi, F.-X. Standaert, C. Wachsmann, "A formal foundation for security features of physical functions," *IEEE Symposium on Security and Privacy* 2011.
- [2] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, P. Tuyls, "Efficient helper data key extractor on FPGAs," *Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2008*, Lecture Notes in Computer Science (LNCS) vol. 5154, pp. 181-197.
- [3] R. C. Bose, D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, no. 1, pp. 68-79, Mar. 1960.
- [4] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, V. Khandelwal, "Design and implementation of PUF-based 'unclonable' RFID ICs for anti-counterfeiting and security applications," *IEEE International Conference on RFID 2008*, pp. 58-64.
- [5] Y. Dodis, L. Reyzin, A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," *Eurocrypt 2004*, Lecture Notes in Computer Science (LNCS) vol. 3027, pp. 523-540.
- [6] B. Gassend, D. Clarke, M. van Dijk, S. Devadas, "Silicon physical random functions," *Proc. ACM Conference on Computer and Communications Security (CCS) 2002*, pp. 148-160.
- [7] B. Gassend, "Physical random functions," M.S. thesis, Dept. EECS, Massachusetts Institute of Technology (2003).
- [8] J. Guajardo, S. Kumar, G.-J. Schrijen, P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," *Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2007*, Lecture Notes in Computer Science (LNCS) vol. 4727, pp. 63-80.
- [9] D. Holcomb, W. Bursleson, K. Fu, "Initial SRAM state as a fingerprint and source of true random numbers for RFID tags," *IEEE International Conference on RFID 2007*.
- [10] D. Lim, "Extracting secret keys from integrated circuits," M.S. thesis, Dept. EECS, Massachusetts Institute of Technology (2004).
- [11] R. Maes, P. Tuyls, I. Verbauwhede, "A soft decision helper data algorithm for SRAM PUFs," *IEEE International Symposium on Information Theory (ISIT) 2009*.
- [12] R. Maes, P. Tuyls, I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs," *Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2009*, Lecture Notes in Computer Science (LNCS) vol. 5747, pp. 332-347.
- [13] Z. Paral, S. Devadas, "Reliable and efficient PUF-based key generation using pattern matching," *IEEE Symposium on Hardware-Oriented Security and Trust (HOST) 2011*.
- [14] J. Proakis, "Digital Communications," 3<sup>rd</sup> Edition, McGraw-Hill Press, 1995.
- [15] A. Rukhin et al., "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *NIST Special Publication 800-22 Rev1a*, 2010.
- [16] G. Selimis, M. Konijnenburg, M. Ashouei, J. Huisken, H. de Groot, V. van der Leest, G.-J. Schrijen, M. van Hulst, P. Tuyls, "Evaluation of 90nm 6T-SRAM as Physical Unclonable Function for secure key generation in wireless sensor nodes," *IEEE International Symposium on Circuits and Systems (ISCAS) 2011*.
- [17] Y. Su, J. Holleman, B. Otis, "A 1.6pJ/bit 96 (percent) stable chip ID generating circuit using process variations," *IEEE International Solid-State Circuits Conference (ISSCC) 2007*, pp. 200-201.
- [18] G. Suh, "AEGIS: a single-chip secure processor," Ph.D. thesis, Dept. EECS, Massachusetts Institute of Technology (2005).
- [19] G. Suh, S. Devadas, "Physical Unclonable Functions for device authentication and secret key generation," *IEEE/ACM Design Automation Conference (DAC) 2007*, pp. 9-14.
- [20] M. Yu, S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Design and Test of Computers*, Special Issue on Verifying Physical Trustworthiness of ICs and Systems, vol. 27, no. 1, pp. 48-65, Jan./Feb. 2010.
- [21] M. Yu, D. M'Raihi, R. Sowell, S. Devadas, "Lightweight and secure PUF key storage using limits of machine learning," *Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2011*, Lecture Notes in Computer Science (LNCS) vol. 6917, pp. 358-373.