An Efficient Reliable PUF-Based Cryptographic Key Generator in 65nm CMOS

Mudit Bhargava, Ken Mai
Department of Electrical and Computer Engineering, Carnegie Mellon University
{*mbhargav,kenmai*}@*ece.cmu.edu*

***Abstract*—Physical unclonable functions (PUFs) are primitives that generate high-entropy, tamper resistant bits for use in secure systems. For applications such as cryptographic key generation, the PUF response bits must be highly reliable, consistent across multiple evaluations under voltage and temperature variations. Conventionally, error correcting codes (ECC) have been used to improve response reliability, but these techniques have significant area, power, and delay overheads and are vulnerable to information leakage. In this work, we present a highly-reliable, PUF-based, cryptographic key generator that uses no ECC, but instead uses built-in self-test to determine which PUF bits are reliable and only uses those bits for key generation. We implemented a prototype of the key generator in a 65nm bulk CMOS testchip. The key generator generates 1213 bits in an area of <50kμm^2 with a measured bit error rate of $< 5 * 10^{-9}$ in both the nominal and worst case corners (100k measurements each). This is equivalent to a 128-bit key failure rate of $< 10^{-6}$. The system can generate a 128-bit key in 1.15μs. Finally, we present a realization of a "strong"-PUF that uses 128 of these highly reliable bits in conjunction with an Advanced Encryption Standard (AES) cryptographic primitive and has a response time of 40ns and is realized in an area of 84kμm^2.**

I. Introduction

A Physical Unclonable Function (PUF) is a die-specific random function or a *silicon biometric* that is unique for every instance of the die. PUFs derive their randomness from uncontrolled random variations in the IC fabrication process (usually undesirable) to create practically unclonable functions even if the original design files are compromised. PUFs are increasingly used as building blocks in many secure systems for applications such as device identification/authentication [1]–[8] and secret key generation [2]–[4], [9]–[13]. PUFs provide an attractive alternative to *storing* of random secret bits in volatile or non-volatile memory (which are vulnerable to attacks [14]) by instead *generating* these random bits every time the PUFs are evaluated.

Most PUF implementations do so by amplifying some electrical characteristic (e.g., delay, threshold voltage) from two nominally identical circuit components in the PUF core. These electrical differences, especially when small, often flip polarity across environmental variations (voltage and temperature), in the presence of ambient noise, or over device aging, resulting in some bits of the raw PUF response being inconsistent/unreliable. Previous hardware studies have shown that for some designs >25% of the PUF response bits may be unreliable across environmental variations [15], [16]. Since electrical differences of larger magnitude require larger variations to flip polarity, a PUF bit is more reliable when generated by amplifying a larger electrical difference.

Although some applications like identification and authentication can be designed to tolerate a few errors in the response without significant loss of security, all applications can benefit from more reliable PUFs, and applications such as key generation require the PUF response to be perfectly reliable. The conventional method to improve PUF reliability is to use powerful error correction codes (ECC) to correct the raw response from the PUF core, with a typical targeted failure rate for a 128-bit key $< 10^{-6}$. This technique requires a PUF enrollment operation, prior to the in-field use of PUFs for key generation. During enrollment, the ECC encoder uses a large number of raw response bits to compute the helper data and the secret key. The helper data is assumed to be public information is typically stored in a non-volatile memory on-die. During in-field operation, the helper data is loaded back on chip and is used by the ECC decoder to compute the same secret key from the re-generated raw PUF response.

Unfortunately, ECC implementations generally have significant VLSI overheads, which scale up quickly as the number of bits of correction increases. To generate a 128-bit key with a targeted key error rate $< 10^{-6}$, ECC implementations typically require 3k-10k PUF raw response bits (with bit error rate of 15%) to generate the key. This is equivalent to using 23-80 raw bits to generate a single reliable bit [17]–[22]. The helper data generated for this case will be typically 3k-15k bits. Large helper data require larger off-chip storage as well as longer time to load the bits during decoding of the key. Further, the helper data has been shown to be a source of information leakage requiring careful design [17], [18]. These overheads, however, reduce significantly if the errors in the raw response bits are reduced. For example, the BCH coding in [23] requires 26.7 raw response bits to generate a reliable bit if the raw response bits exhibit 15% errors but requires only 3.68 raw response bits if the errors reduce to 6%.

In this work, we detail an efficient, reliable, sense amplifier PUF-based key generator design that achieves a 128-bit key error rate of $< 10^{-6}$ without using any error correction coding (ECC). Additionally, we describe a realization of an efficient "strong-PUF" built using our key generator. Finally, we present measured results from a custom testchip prototype of our key generator and strong PUF in 65nm bulk CMOS. Measurements are done across a wide range of voltage (±200mV of nominal VDD) and temperature (-20°C to 85°C) variations.

II. Key Generator Design

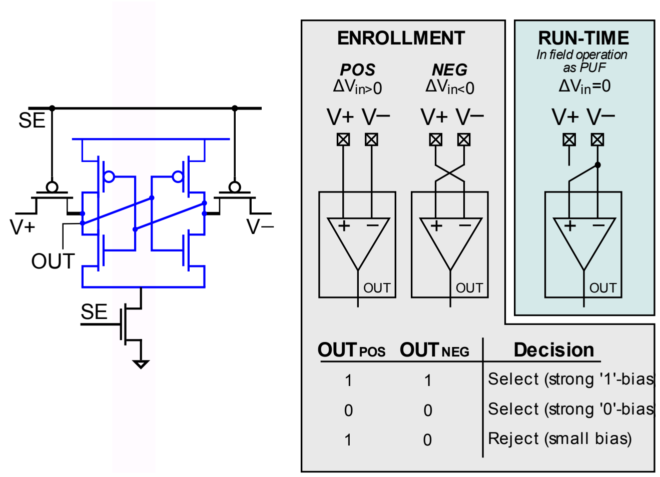
Our efficient, high-reliability, ECC-less cryptographic key-generator uses a sense amplifier (SA) based PUF core surrounded by built-in self-test (BIST) logic to pre-characterize and selectively use only the reliable bits. We first describe the design and operation of a SA PUF and how they enable reliability characterization. Then we describe the BIST logic that characterizes the reliability in a self-contained manner. Then, we describe the design of a "strong"-PUF that uses the bits generated from the key generator in conjunction with a standard one-way cryptographic primitive (AES).

Sense Amplifier (SA) PUF. SAs are clocked circuits that amplify small differential voltages into full swing digital values and are commonly used in memory read paths and as voltage comparators (Figure 1(a)). Under ideal conditions, an ideal SA would correctly amplify even the smallest of input differential voltages. In practice, however, variations in the devices of an

978-3-9815370-2-4/DATE14/©2014 EDAA

SA may result in an offset (or bias), a measure of the natural tendency of the SA to resolve to a particular polarity. In typical SA implementations, correct operation is ensured by providing the SA inputs with a voltage difference larger than the offset voltage ($|V_{OFFSET}|$). Offset of a SA results from a combination of systematic and random variations. Systematic variations can be due to manufacturing gradients and layout asymmetries [24], and can be minimized by symmetric layout of matched devices. Random variations are a result of random uncertainties in the fabrication process such as random dopant fluctuation (fluctuations in the number and location of dopants in the transistor channel) [25] and gate line-edge roughness [26]. The effects can be mitigated by using larger devices [27].

SAs can be used as PUFs by evaluating them while providing a zero differential input voltage [28]. To maximize randomness and reliability, SAs as PUF cores should be built using a regular layout and by using small-sized devices [28]. A regular layout will minimize any systematic variations (high randomness) [24] and small-sized devices will maximize the variations in the device characteristics (high reliability) [27]. Hardware measurements have previously shown that SA PUFs designed this way have good randomness and uniqueness characteristics [29].

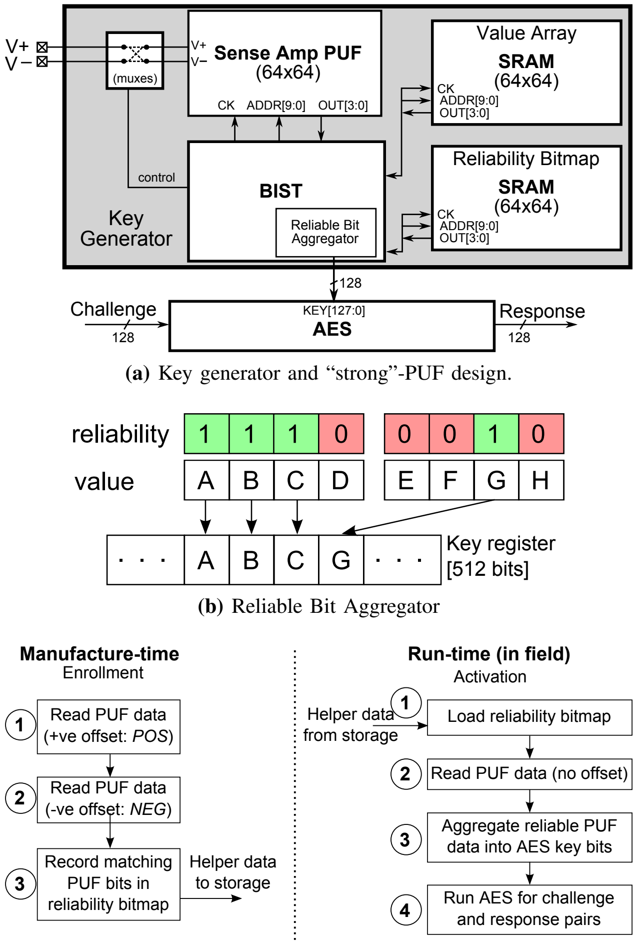


(a) Latch-style sense amplifier **(b)** Enrollment and run-time operations

Figure 1: (a) Latch-style sense amplifier schematic (bistable portion highlighted in blue) (b) Enrollment (pre-characterization) and in-field operation (as PUF).

Reliability Characterization for SA PUFs. The magnitude of the offset voltage of a SA ($|V_{OFFSET}|$) is a good indicator of its reliability for use in a PUF. A SA with high $|V_{OFFSET}|$ (i.e., a strong bias to resolve to a particular polarity) will likely resolve to the same polarity across environmental variations and over aging. Measured hardware results have shown high reliability from SA PUFs with $|V_{OFFSET}|$ >50mV [15], [28].

Figure 1(b) shows the two-phase enrollment operation used to characterize the reliability of a SA PUF. If a large number of SAs are arrayed with their inputs shorted across all of them (i.e., a common V+ and a common V–), then the entire array of SAs can be characterized together. In the first phase (*POS*), the inputs are configured such that $\Delta V_{IN} = (V+) - (V-)$ and in the second phase (*NEG*), the connections are reversed so $\Delta V_{IN} = (V-) - (V+)$. As shown in Figure 1(b), at the end of enrollment, a SA is selected as a potentially reliable one if the output of the SA is consistent (either 1 or 0) for both the phases. A consistent output of a SA is an indicator that its $|V_{OFFSET}| > \Delta V_{IN}$ (i.e., an external ΔV_{IN} was insufficient to make a SA flip its preferred polarity) and hence the SA has a high probability to resolve to a consistent polarity when $\Delta V_{IN} = 0$ across different and noisy environmental conditions. V+ and V– are kept fixed at a voltage differential (ΔV_{IN}) that provides sufficient robustness over environmental variations. Previous studies have shown that a ΔV_{IN} ~50mV results in selection of ~50% of SAs which have extremely high reliability [15] but this will vary with SA design and process technology.



(a) Key generator and "strong"-PUF design.

(b) Reliable Bit Aggregator

(c) Flowchart of key generation and "strong"-PUF realization.

Figure 2: (a) A self-contained BIST controlled sense amplifier (SA) PUF based key generator. The BIST automatically generates the reliability bitmap in the enrollment phase which is then used at run-time to select reliable bits from the SA PUF array. The first 128 of these bits are used as the key in an AES primitive to realize a reliable and secure "strong"-PUF. The input and output of the AES primitive are treated as the challenge and response respectively. (b) Aggregation of reliable PUF bits into the AES key. Letters used instead of bits in value for clarity. A total of 512 bits stored in the key register for test purposes; the first 128 of these are used for AES key. (c) Flowchart of key generation and then the subsequent realization of a "strong"-PUF.

Self-contained Key Generator. Figure 2(a) shows the top-level schematic of our self-contained BIST-controlled key generator. It consists of a 64x64 SA PUF array and two 64x64 SRAM arrays. The SA PUF array is arranged and designed much like a typical SRAM array and each row of the SA array is activated by a rising sense enable (SE) signal (Figure 1(a)) which is implemented like the word-line (WL) signal in an SRAM array. Figure 2(c) describes the key generator operation. During the first phase (*POS*) of enrollment (pre-characterization), the values from the SA PUF array (OUT_{POS} in Figure 1(b)) are read and temporarily stored in the SRAM array named 'Value Array'. In the second phase (*NEG*), the values from the SA PUF array (OUT_{NEG} in Figure 1(b)) are compared with OUT_{POS}) by simultaneously accessing the 'Value Array'. If $OUT_{POS} = OUT_{NEG}$ for a SA, suggesting its

$|V_{OFFSET}| > \Delta V_{IN}$, its location is marked as potentially reliable by storing a '1' at the corresponding location in the Reliability Bitmap SRAM array. The written word into the Reliability Bitmap is generated by a bit-wise XNOR of OUT_{POS} and OUT_{NEG}. The end product of enrollment is the completely filled Reliability Bitmap array which is equivalent to the helper data of typical ECC schemes. The bitmap is used during run-time for key generation in the field. The bitmap can be public information and stored in on-die non-volatile memory or off-chip, but if stored off-chip, additional security measures may be necessary.

Figure 2(c) shows the execution steps at run-time when the key generator is used in the field. First, the reliability information is loaded into the Reliability Bitmap array. Then the SA PUF array is activated with $\Delta V_{IN} = 0$ (Figure1(b)) while the corresponding reliability information is also read from the Reliability Bitmap array. As shown in Figure 2(b), the Reliable Bit Aggregator aggregates the SA values from the first N reliable locations (as per the Reliability Bitmap) to generate and stores a reliable N-bit key in registers. In our design, we have a provision of aggregating and storing N=512 bits for test purposes and only the first 128 bits are used as a key to an AES.

Note that the helper data (Reliability Bitmap block) carries no information about polarity of the bits but only the physical location of the potentially more reliable bits. Hence they do not leak any information about the bits unless there is a location-based correlation found in the bits generated from the SA PUFs in the array. However, the biggest contributor of V_{OFFSET} (and hence the polarity of bit) is *local* random variations in the devices of a SA (e.g., random dopant fluctuations and line edge roughness) and hence the bits of the array can be assumed to largely independent.

"Strong"-PUFs. A strong-PUF is defined as a PUF with extremely large number of challenge-response pairs [23], [30]. Most PUF implementations (including ROs, SRAM, SAs) only generate a small number of random bits. As discussed in [30], designing a *true* strong-PUF is challenging and will possibly be impractical for most applications. We propose the use of random bits as a key in a secure one-way cryptographic function to realize a strong-PUF. In our design, we use 128 reliable bits from the Reliable Bit Aggregator and use them as a key in an AES, a standard one-way encryption primitive which is widely used and considered extremely secure, to design a *practical* strong-PUF (Figure 2(a)). The 128-bit input and output to the AES can then be considered the challenge-response pair (CRP) of the strong-PUF thus realized. The key bits used are unique across chips and hence the responses of this strong-PUF will be unique across chips.

III. Testchip Description

Prototype Testchip. A prototype of the key generator and strong PUF was designed and fabricated on a 65nm bulk CMOS custom ASIC testchip. The entire design has an area of 0.119 mm^2 (Figure 3). The total area of the testchip is 2.5mm x 2.2mm, with 130 I/O pads and several unrelated projects.

The SRAMs and the SA PUFs are custom designed arrays, each having 1024 words and 4bits/word. The BIST and the AES primitive are synthesized using a commercial standard-cell library. Additional scan chains (not shown in Figure 2(a)) are used for back-door access to the two SRAM arrays for test purposes. These would be neither required nor desired from a security standpoint in a production of our key generator.

Table I shows the area consumption of the key generator and the realized "strong"-PUF and the gate equivalents (GE) for the synthesized blocks. The total on-die area of the design is 118.8k μm^2. The custom designed 4096 bit SA PUF and two SRAM arrays (both 64x64 column-mux 16 arrays) take 8.1% (9.7k μm^2) and 13.0% (15.4k μm^2) of the overall area respectively. The remaining 78.9% of area is synthesized logic. The synthesized logic is composed of the key-generator logic (IO-Ctrl and Test Ctrl) and the AES block. The key-generator logic fits in a total of 3963 GEs and in an area of 24k μm^2. The total area of the key generator (SA/SRAM arrays and control/IO logic) is 49.1k μm^2. The AES was built using 10487 GEs in an area of 34.4k μm^2. The total on-die area of the synthesized logic is more than the sum of the area of individual synthesized blocks because of placement and routing overhead as well as test circuits.

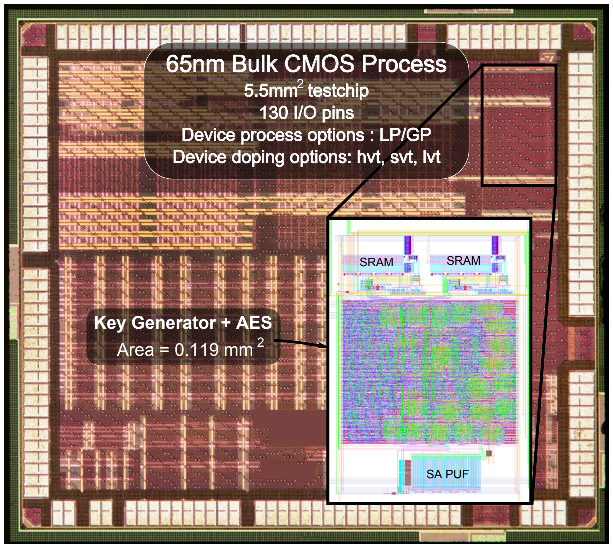


Figure 3: Die shot and layout capture of the key generator system, and the realization of a "strong"-PUF using an AES primitive which supports 128-bit challenge-response pairs.

Design Block	GE	Area on Chip
Synthesized logic	14,452	93,700 μm^2†
→AES	10,487	34,436 μm^2
→IO-Ctrl	1,493	10,844 μm^2
→Test-Ctrl	2,470	13,198 μm^2
Value/Reliability SRAMs	–	2 x 7,700 μm^2
Sense Amp PUF	–	9,700 μm^2
Total area	–	118,800 μm^2

Table I: Area of the key generator. † The total area of the synthesized logic is more than the sum of the area of different synthesized blocks as it includes the place and route overhead as well as several scan-flops for test purposes.

IV. Experimental Results

In this section we provide measured results of the key generator from our 65nm CMOS testchip. First, we present the reliability results followed by the area and speed measurements of the key generator and the "strong"-PUF.

Reliability of Generated Key. The key generator creates a Reliability Bitmap that is used to select a set of bits for use in-field. A higher ΔV_{IN} will result in a smaller set of selected SAs, but one with higher expected reliability. Hence, reliability for our design will be a function of the chosen ΔV_{IN}. Figure 4(a),(b) show the measured outputs from the SA PUF array after the *POS* and the *NEG* phases in enrollment for different values of ΔV_{IN}. Figure 4(c) shows the reliability bitmap that is generated after the two phases. Each square in Figure 4 consists of 64x64 pixels that represent the contents of a 64x64 array. A ‘1’ in the SA PUF output is represented by a white pixel and a ‘0’ in the PUF output is represented by a black pixel. We repeated the *POS* and *NEG* phases 100 times, and some of the PUF outputs are noisy, i.e., they sometimes resolve to a ‘0’ and sometimes to a ‘1’. Such SA PUFs are represented by a grey pixel and the ratio of ‘1’s and ‘0’s is indicated by the greyscale value. The reliability bitmap is generated using the first evaluation of the SA PUF. A white pixel in the Reliability Bitmap represents a bit location to be selected (reliable) and a black pixels represent the locations of the unselected (unreliable) bits. We see that for a higher ΔV_{IN}, the SA PUFs tend to generate more 1’s in the *POS* phase (more white pixels in Figure 4(a)) and more 0’s in the *NEG* phase (more black pixels in Figure 4(b)). Also, for higher ΔV_{IN}, there is a more rigorous selection of reliable bits, as is seen by the less number of white pixels in the Reliability Bitmap.

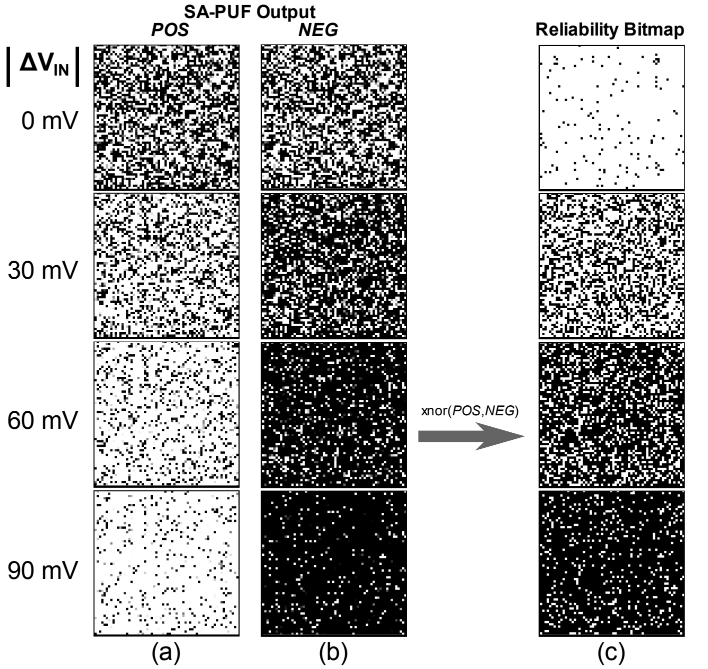


Figure 4: SA PUF response map in the (a) *POS* and the (b) *NEG* phases.The percentage of ‘1’s and ‘0’s is ∼ 50% for ΔV_{IN}=0mV. With an increasing ΔV_{IN}, the response maps in the two phases show a bias towards a higher percentage of ‘1’s and ‘0’s in the *POS* and the *NEG* phase respectively. (c) The reliability bitmap shows a reduced number of selected SAs for higher ΔV_{IN}.

This is further illustrated in Figure 5 which shows the percentage of SA locations that are selected during enrollment for different values of ΔV_{IN}. For example, for ΔV_{IN}=50mV, 38.4% (1573 of 4096 bits) were selected, but for ΔV_{IN}=120mV, only 3.7% (151 of 4096 bits) were selected.

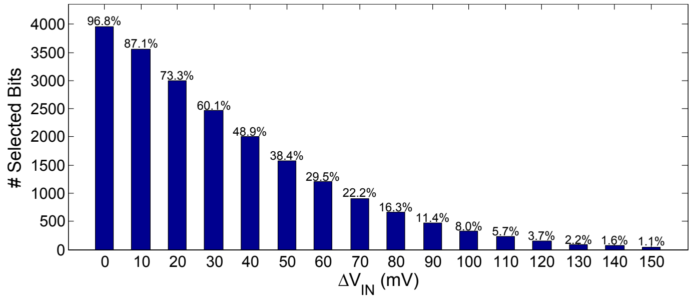


Figure 5: Percentage of bits (of the 4096 bits in the SA PUF array) that are selected in the enrollment stage i.e. percentage of 1’s in the Reliability Bitmap for various ΔV_{IN}

We first took 10,000 measurements (run-time operation as shown in Figure 1(b) and Figure 2(b)) at all combinations of voltage: 1.0V, 1.2V, 1.4V and temperature: -20°C, 27°C, 85°C. We then took 150,000 measurements at the corner with worst reliability as measured from the initial 10,000 runs (found to be 1.0V, 85°C). We also take 100,000 measurements at the nominal corner (1.2V, 27°C). Any selected SA that resolves inconsistently in any of the measurements is considered an unreliable SA and every inconsistent SA increases the bit error count by 1. The BER is then ratio of bit error count, averaged across all runs, to the total number of selected bits. For a key to be error free, all the bits of the key must be error free. We assume the secrecy rate to be 0.75 (i.e., entropy per bit = 0.75), using the results from previous studies on bi-stable based PUFs [23], [31]. Hence, to obtain a 128-bit key, $\lceil 128/0.75 \rceil = 171$ bits need to be generated which can be later compressed (privacy amplification) to 128-bits of perfect entropy. Hence for this study, we define the key error rate $KER = 1 - (1 - BER)^{171}$.

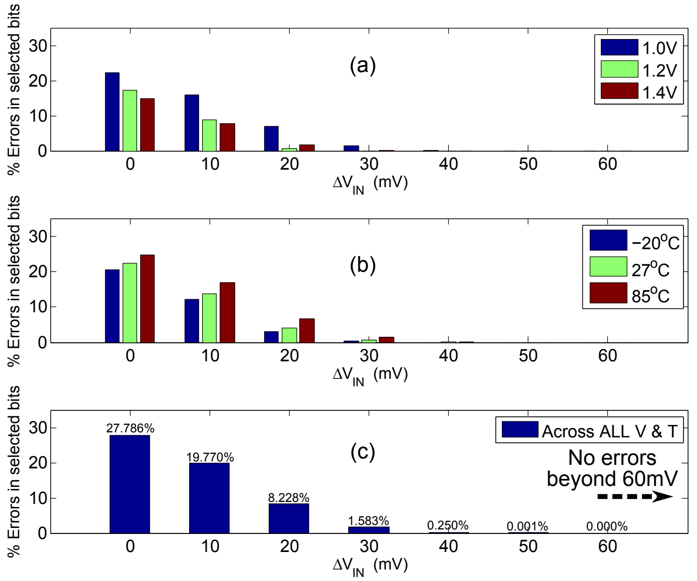


Figure 6: Percentage Error bits in the selected set of SAs. For the error measurements, the SA PUF array was evaluated 10,000 times at all combination of voltage (1.0V, 1.2V, 1.4V) and temperature (-20°C, 27°C, and 85°C) (nominal: voltage=1.2V; temperature = 27°C), hence a total of 90,000 evaluations. (a) Errors across temperature variations while keeping the voltage constant. (b) Errors across voltage variations while keeping the temperature constant. (c) Errors across all voltage and temperature variations.

Figure 6 shows the measured errors in the selected set of SAs for different ΔV_{IN}. As expected, a higher ΔV_{IN} results in reduced number of errors. Figure 6(a) shows the errors across temperature variations at constant voltages of operation. Figure 6(b) shows the errors across voltage variations when operated at constant temperatures. Figure 6(c) shows the errors across all voltage and temperature variations as measured from the first batch of 10,000 measurements across all 9 corners. We find that when enrollment is done using $\Delta V_{IN} = 60mV$, no errors were found in any of the 1213 selected bits in the first batch of measurements. We then performed an additional 140,000 and 90,000 measurements at the worst case corner (1.0V, 85°C) and the nominal corner respectively, that took over 20 days to generate, and still found no errors in any of the runs.

The measured BER, the model, and the worst case bit error rate (BER_{WC}) are shown in Figure 7. BER_{WC} is defined as follows. If no errors were found in N runs of S selected bits, then we pessimistically assume that the first error would occur in the very next measurement and define our $BER_{WC} = 1/(N*S+1)$. The measured BER is at very least lower than BER_{WC}. We were able to achieve a good curve fit by choosing the following model: $log(y) = ae^{bx}$ where $y = BER$ and $x = \Delta V_{IN}$, but must exist an error floor due to various error phenomena such as energetic particle strikes, thermal noise, etc. However, at such error levels, the PUF would likely not be any more unreliable than any other part of the digital circuitry.

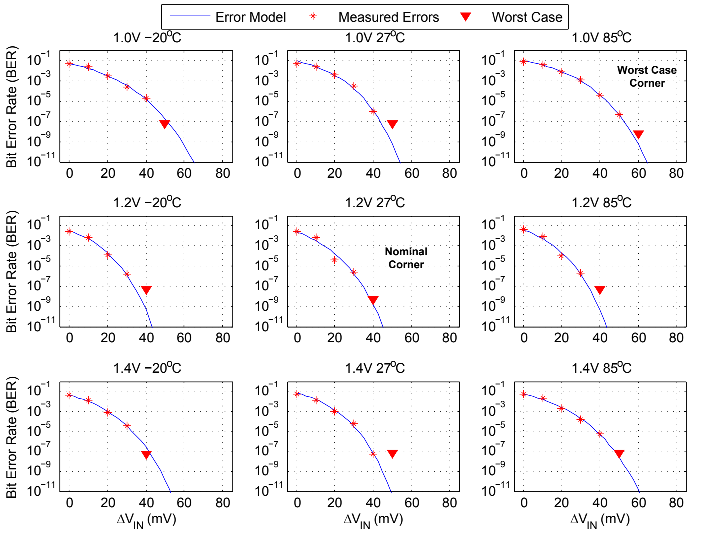


Figure 7: Measured bit error rate (BER) and modeling of errors beyond the measured range. The measured errors are from 150,000 measurements in the worst case corner, 100,000 measurements in the nominal corner, and 10,000 measurements at all other corners. If no errors are observed in N measurements of S selected bits, then the worst case bit error rate is pessimistically defined as 1 error in N*S measurements. The following error model was a very good fit: $log(y) = ae^{bx}$ where $y = BER$ and $x = \Delta V_{IN}$.

Uniqueness of SA Bits Across Dies. Uniqueness is a measure of how uncorrelated the response bits are across chips, and ideally the response bits should differ with a probability of 0.5. The Hamming distance (HD) of a k-bit response from ideally unique chips should follow a binomial distribution with parameters $\mathcal{N} = k$ and $p = 0.5$ and the mean of the HD distribution should be equal to k/2. For uniqueness measurements, we create 256 16-bit words from the 4096 SA raw bits from 15 different chips for HD computation. Figure 8 shows that the pair-wise HD of response bits from three arbitrarily chosen chips. The HD distribution is close to ideal and the mean of HD (ideally 8.00 for k=16) for all pair-wise combinations taken from 15 measured chips (i.e., total 105 combinations) was found to be in the range 7.69–8.26.

Randomness of the Selected Bits. We compute the percentage of 1's in the selected set of bits for various ΔV_{IN} just to ensure that the selection process does not favor bits with a certain polarity. We find that the percentage of 1's falls from 48-50% to 43-45% as ΔV_{IN} is increased (Figure 9) from 0mV to 100mV. Such a deviation from the ideal 50%, as seen consistently across multiple chips, was unexpected. The underlying SA PUFs used in our designs have been shown in previous studies to have a bias close to the ideal 50% [15]. Further, measurements showed that SAs have a symmetric distribution of their offset voltages with a zero mean (i.e., the number of SAs with offset >ΔV_{IN} and ¡-ΔV_{IN} for any ΔV_{IN} should be same). A differential voltage drop in the global nets V+ and V– arising from a layout or PCB mismatch could possibly be causing the bias but we do not suspect the bias to be structural dependent.

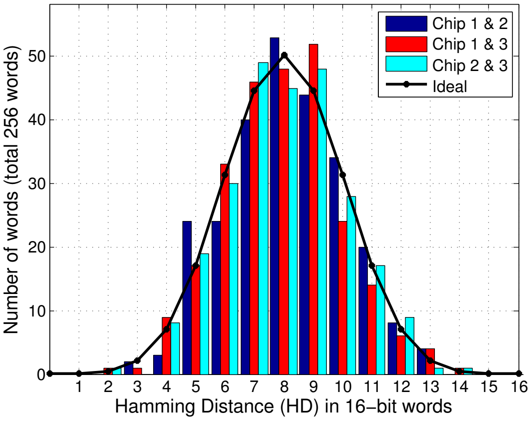


Figure 8: Histogram of Hamming distance (HD) of response words from the SAs across three chips. Also shown is the probability mass function of the HD in responses from ideally unique chips. For the HD comparison, 256 16-bit words are created from 4096 bits of the SA arrays. The pair-wise HD of response bits from the three chips is close to ideal with means of 8.16 7.93, and 7.99.

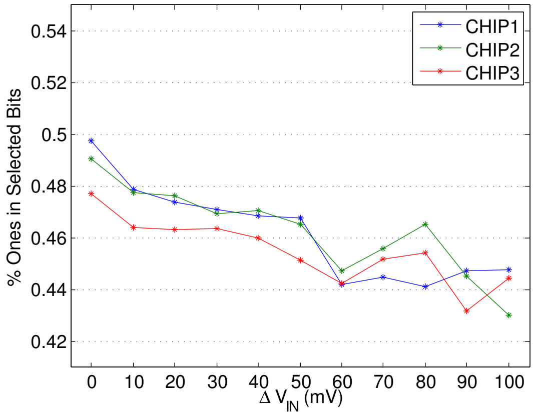


Figure 9: Percentage of 1's in the selected bits for various ΔV_{IN} for 3 chips.

Speed of Enrollment of SA PUFs. The enrollment needs to be done just once to extract the reliablity bitmap of the SA PUF array. For this design, the enrollment is a completely self-contained operation and requires no configuration except fixing the two signal pins V+ and V– to provide a sufficient ΔV_{IN} which could be done with an internal resistive ladder or other bias generator circuits. Enrollment requires reading all locations of the SA array twice, one each for the *POS* and the *NEG* phase with a break between the phases to allow the inputs of the SAs to settle to the switched voltages. Our design uses 4-bit words and hence requires 1024 cycles for each phase. From the simulations of the design with parasitic capacitances extracted, we estimated that the SA inputs take <250ns to settle. Although we tested our design in silicon at only 10 MHz due to test equipment limitations, in simulations the key generator is able to run at up to 250 MHz. At 250MHz, enrollment would take $\sim 8.5\mu s$ and at the measured frequency of 10MHz, enrollment took $\sim 205\mu s$. Note that the speed of enrollment could be further increased by increasing the word-size of the SA and the SRAM arrays from 4, but at the cost of increased I/O pins since this reliability information has to be sent off chip for storage.

Speed of Key Generation. At run-time, the key generator first loads the Reliability Bitmap (generated during enrollment and stored off-chip). The SA PUF is then evaluated while the

Reliable Bit Aggregator accumulates the reliable bits of the key by processing the output words from the SA PUF array and the Reliability Bitmap array, one word at a time. For our 4-bit word designs, this run-time key-generation operation takes a maximum of 2048 cycles - 1024 cycles to load the Reliability Bitmap, and 1024 cycles to aggregate the key bits. From the measured data, however, it was found that to aggregate 171 bits, it would require nominally accessing only 570 of the 4096 bits (or 143 cycles instead of 1024). At simulation speed of 250MHz, run-time key generation would take $1.15\mu s$ and at the measured speed of 10MHz, it would take $28.6\mu s$. Note, that it also means that the key could be generated by storing only $\sim$570 bits of reliability bitmap, i.e., <4.5 bits of helper data per generated bit. This is $\sim$5-25x lower than the size requirement of helper data in conventional ECC. Further, as discussed earlier in Section II, the reliability bitmap is significantly more resilient to information leakage as compared to the helper data in conventional ECC.

Speed of the "Strong"-PUF (response time). The AES primitive implemented in our system requires 10 cycles to generate a 128-bit response for a 128-bit challenge. At the simulated frequency of 250 MHz, this is 40ns per challenge/response pair.

V. Conclusions

In this work, we presented a highly reliable, self-contained, efficient key generator based around a sense amplifier PUF. Measurements from our 65nm custom ASIC testchip show that our key generator operates with a bit error rate $BER < 5 * 10^{-9}$, equivalent to a 128-bit key error rate of $< 10^{-6}$. Such low BER is conventionally only achievable using powerful, but costly, error correction codes (ECC). Our key generator eschews these costly ECC blocks and rather uses reliability pre-characterization and post-silicon PUF element selection of reliable bits. Finally, we leveraged this high reliability of the key generator, in conjunction with an AES cryptographic primitive, to build a reliable, efficient, and secure "strong" PUF.

Acknowledgment

This work was supported in part by National Science Foundation grant CNS1117755. We also thank Mark McCartney for his significant testchip design and implementation support.

References

[1] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2002, pp. 148–160.

[2] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in *Proceedings of 44th ACM/IEEE Design Automation Conference DAC '07*, 2007, pp. 9–14.

[3] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *Proceedings of Digest of Technical Papers VLSI Circuits 2004 Symp*, 2004, pp. 176–179.

[4] J. Guajardo, B. kori, P. Tuyls, S. Kumar, T. Bel, A. Blom, and G.-J. Schrijen, "Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions," *Information Systems Frontiers*, vol. 11, pp. 19–41, 2009.

[5] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, 2009.

[6] B. Gassend, D. Lim, D. Clarke, M. van Dijk, and S. Devadas, "Identification and authentication of integrated circuits," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 11, pp. 1077–1098, 2004.

[7] G. Hammouri, E. Ozturk, B. Birand, and B. Sunar, "Unclonable lightweight authentication scheme," in *Information and Communications Security*, ser. Lecture Notes in Computer Science, L. Chen, M. Ryan, and G. Wang, Eds. Springer Berlin / Heidelberg, 2008, vol. 5308, pp. 33–48.

[8] P. Koeberl, J. Li, R. Maes, A. Rajan, C. Vishik, and M. Wjcik, "Evaluation of a PUF Device Authentication Scheme on a Discrete 0.13um SRAM," in *Trusted Systems*, ser. Lecture Notes in Computer Science, L. Chen, M. Yung, and L. Zhu, Eds. Springer Berlin / Heidelberg, 2012, vol. 7222, pp. 271–288.

[9] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. VLSI Syst.*, vol. 13, no. 10, pp. 1200–1205, 2005.

[10] B. Skoric, P. Tuyls, and W. Ophey, "Robust Key Extraction from Physical Uncloneable Functions," in *Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, J. Ioannidis, A. Keromytis, and M. Yung, Eds. Springer Berlin Heidelberg, 2005, vol. 3531, pp. 407–422.

[11] Intrinsic-id: Products webpage: www.intrinsic-id.com/products.

[12] Verayo, Inc. Products webpage: www.verayo.com/products.

[13] NXP, Inc.: Press release (Feb 21, 2013): http://www.nxp.com/news/press-releases/2013/02/nxp-strengthens-smartmx2-security-chips-with-puf-anti-cloning-technology.html.

[14] K. Mai, "Introduction to hardware security and trust," ser. SpringerLink : Bücher. Springer New York, 2012, ch. Side Channel Attacks and Countermeasures.

[15] M. Bhargava, C. Cakir, and K. Mai, "Comparison of Bi-stable and Delay-based Physical Unclonable Functions from Measurements in 65nm bulk CMOS," in *Custom Integrated Circuits Conference, 2012. CICC '12. IEEE*, Sept 2012.

[16] R. Maes, V. Rozic, I. Verbauwhede, P. Koeberl, E. van der Sluis, and V. van der Leest, "Experimental evaluation of Physically Unclonable Functions in 65 nm CMOS," in *ESSCIRC (ESSCIRC), 2012 Proceedings of the*, Sept. 2012, pp. 486 –489.

[17] M.-D. Yu and S. Devadas, "Secure and Robust Error Correction for Physical Unclonable Functions," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 48–65, 2010.

[18] R. Maes, A. V. Herrewege, and I. Verbauwhede, "PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator," in *CHES*. Springer, 2012, pp. 302–319.

[19] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology - EUROCRYPT 2004*, ser. Lecture Notes in Computer Science, C. Cachin and J. Camenisch, Eds. Springer Berlin Heidelberg, 2004, vol. 3027, pp. 523–540.

[20] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," in *Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems*, ser. CHES '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 63–80.

[21] C. Bosch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient Helper Data Key Extractor on FPGAs," in *Cryptographic Hardware and Embedded Systems CHES 2008*, ser. Lecture Notes in Computer Science, E. Oswald and P. Rohatgi, Eds. Springer Berlin / Heidelberg, 2008, vol. 5154, pp. 181–197.

[22] M.-D. M. Yu, D. M'Raihi, R. Sowell, and S. Devadas, "Lightweight and secure PUF key storage using limits of machine learning," in *Proceedings of the 13th international conference on Cryptographic hardware and embedded systems*, ser. CHES'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 358–373.

[23] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Physical Unclonable Functions and Public-Key Crypto for FPGA IP Protection," in *Proceedings of Int. Conference Field Programmable Logic and Applications FPL 2007*, 2007, pp. 189–195.

[24] K. Agarwal and S. Nassif, "Characterizing Process Variation in Nanometer CMOS," in *Proceedings of 44th ACM/IEEE Design Automation Conference DAC '07*, 2007, pp. 396–399.

[25] R. W. Keyes, "Effect of randomness in the distribution of impurity ions on FET thresholds in integrated electronics," *IEEE J. Solid-State Circuits*, vol. 10, no. 4, pp. 245–247, 1975.

[26] P. Oldiges, Q. Lin, K. Petrillo, M. Sanchez, M. Ieong, and M. Hargrove, "Modeling line edge roughness effects in sub 100 nanometer gate length devices," in *Proceedings of Int. Conference Simulation of Semiconductor Processes and Devices SISPAD 2000*, 2000, pp. 131–134.

[27] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct. 1989.

[28] M. Bhargava, C. Cakir, and K. Mai, "Attack resistant sense amplifier based PUFs (SA-PUF) with deterministic and controllable reliability of PUF responses," in *Proceedings of IEEE Int Hardware-Oriented Security and Trust (HOST) Symp*, 2010.

[29] ——, "Reliability enhancement of bi-stable PUFs in 65nm bulk CMOS," in *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*, june 2012, pp. 25 –30.

[30] U. Ruhrmair, H. Busch, and S. Katzenbeisser, "Strong PUFs: Models, Constructions, and Security Proofs," in *Towards Hardware-Intrinsic Security*, ser. Information Security and Cryptography, A.-R. Sadeghi and D. Naccache, Eds. Springer Berlin Heidelberg, 2010, pp. 79–96.

[31] V. Leest, B. Preneel, and E. Sluis, "Soft Decision Error Correction for Compact Memory-Based PUFs Using a Single Enrollment," in *Cryptographic Hardware and Embedded Systems CHES 2012*, ser. Lecture Notes in Computer Science, E. Prouff and P. Schaumont, Eds. Springer Berlin Heidelberg, 2012, vol. 7428, pp. 268–282.