An Integrated TRNG-PUF Architecture based on Photovoltaic Solar Cells

Amit Degada
University of Kentucky

Himanshu Thapliyal
University of Kentucky

***Abstract*— The objective of the article is to present an integrated True Random Number Generator (TRNG) and Physically Unclonable Function (PUF) architecture using Photovoltaic solar cells. We illustrate that the Photovoltaic (PV) solar cell sensor response can be engineered into dynamic (TRNG) and static responses (PUF). The proposed prototype uses the iterative Von Neumann post-processing scheme to produce random bits with 34% better throughput compared to a single Von Neumann operation. The random bit quality was checked by statistical test suites from the National Institute of Science and Technology (NIST) and achieves an average p-value of 0.45 at all variations in light intensity. The PUF response achieves 92.13% reliability and 50.91% uniformity. The integrated TRNG-PUF architecture is beneficial for resource-constrained Cyber-Physical System (CPS).**

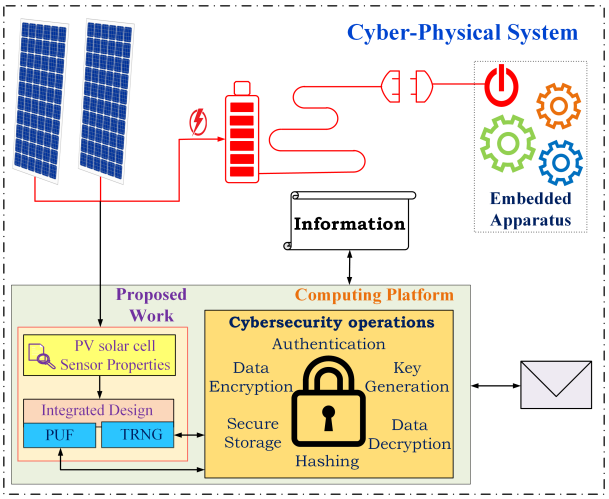


Fig. 1: Building security primitives from solar cells and sensors in CPS.

I. Introduction

The Cyber-Physical System (CPS) integrates sensors, computing platforms, and networking among constituent blocks. The application space of CPS includes many intelligent consumer electronics appliances such as in aerospace, smart-home, vehicles, manufacturing plants, healthcare, real-time traffic monitoring, chemical process control, environmental monitoring, and smart-grid [1], [2]. The inherently decentralized framework is a blend using networking technology and subsequently provides many vulnerable points to compromise security. Therefore, it is challenging to ensure confidentiality, integrity, authenticity, and availability across different physically integrated devices [3].

The use of renewable energy sources to fulfill energy requirements is a convenient way in the decentralized framework of CPS. The solar panels, as shown in Figure 1, are preferred to supply the energy need in many embedded apparatus in CPS. To build hardware security primitives, e.g. True Random Number Generator (TRNG) and Physically Unclonable Function (PUF) from sensors in CPS, could be a novel research direction. TRNG is useful to generate secret key, initialization vector, padding, nonce, and salt bits in cryptographic framework [4]. On the other hand, PUF exploits the manufacturing variation of the device to produce a reliable, unclonable unique ID that can be used to authenticate sensors and generate a secret key. The Photovoltaic (PV) solar cell panels are preferred way to harvest solar energy in CPS and thereby, PV solar sensors find commonplace in many CPS [5]. Therefore, designing TRNG and PUF using sensors (in our case PV solar cell sensor) and microcontroller-based computing platform is a novel research direction.

The integration of TRNG and PUF as inte-

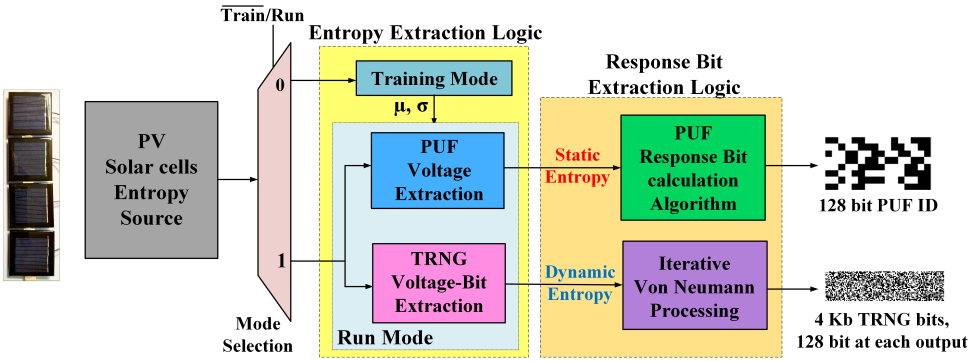


Fig. 2: Schematic of integrated TRNG-PUF architecture.

grated architecture is a challenging task because of the fundamental difference of the PUF and TRNG design. There are existing works that have demonstrated the integrated TRNG-PUF architecture based on Field Programmable Gate Array (FPGA) [6] and CMOS [7], [8]. However, to the best of our knowledge, there is no existing work on integrated TRNG-PUF design based on Photovoltaic (PV) solar cells. Therefore, this article proposes integrated TRNG-PUF architecture devised around a common entropy source of Photovoltaic (PV) solar cells. Further, the proposed architecture does not require additional hardware and can be ported across the existing framework.

II. Integrated TRNG-PUF Architecture

The proposed prototype shown in Figure 2 operates in two modes: (i) Training, and (ii) Run.

- The training mode learns the entropic nature of PV solar cells. The mean value (μ) and Standard Deviation (SD) (σ) of each solar cell voltage histogram is recorded. Additionally, the training mode sets an optimal sampling interval, a vital step to set optimum TRNG throughput. The detailed explanations are presented in the subsequent sections.
- The run mode segregates sensor response in either dynamic (large variation) response to produce TRNG output or static (stable) response to generate PUF output. The prototype has an option to enter in training mode before producing each TRNG/PUF response. The updated training information can reflect the change in response due to light intensity variations.

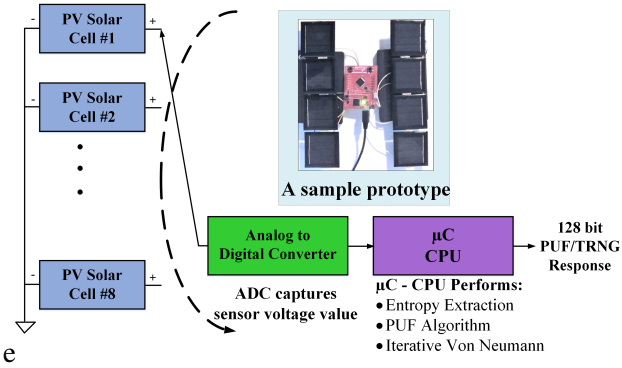
A. TRNG BITS Generation

The TRNG transform randomness in entropic source to generate random bits. The proposed prototype produces initial binary streams by comparing the successive voltage samples produced outside one SD (σ) around mean (μ) in voltage histogram. However, the natural random source has a high correlation between the successive samples, and post-processing becomes inevitable. There exist many techniques for post-processing, and among them, Von Neumann (VN) is particularly useful for limited computing power and small memory size. The proposed prototype implements the Iterative Von Neumann (IVN) approach to reduce wastage of initial binary streams ($\approx$ 76%) [9] in single Von Neumann block.

B. PUF BITS Generation

Over the years, many researchers have attempted to design the PUF using sensors. The electrical voltage of PV solar cells should have a predictable relationship with environmental conditions, e.g. ambient light for PV solar cells. Further, the response of many samples should settle to a static value. Additionally, the algorithm chosen should require less computing power and memory for IoT applications. One such algorithm is proposed in our earlier works [10] and the proposed prototypes in this research adapt the same method. The prototype produces a PUF response bit by calculating average voltage over one standard deviation (σ) around the mean (μ). This approach helps to reject outlier sample voltage response that typically arises naturally and thereby calculates a more stable voltage response.

C. Electrical Schematic of proposed prototype



e

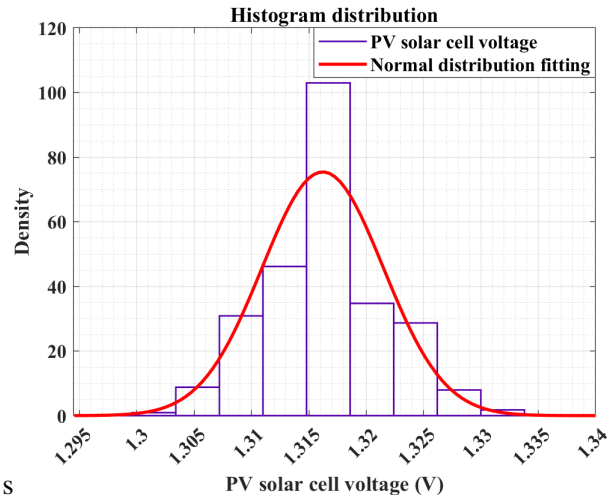
Fig. 3: An example electrical schematic.

Figure 3 is one of the possible ways to implement the proposed architecture and it can be explained in three steps. In step 1, the Analog-to-Digital Converter (ADC) samples eight PV solar cell sensors and converts voltage value into an equivalent digital reading. The next step implements training and run mode. In addition to that, CPU also implements an algorithm to produce 128-bit PUF response and IVN technique to produce 128-true random bits. In step 3, the TRNG and PUF response bits can be communicated further to perform cryptographic co-processor functions.

III. Entropy Extraction Logic

The PV solar cell is a p-n junction diode, and its output voltage depends on several variables. These variables include manufacturing process variations between sensors, the number of photons falling over the p-n junction, lifetime of electron-hole, doping of p and n-type material, area of the p-n junction and mobility of the charge. As these variables are random, we hypothesize that the photovoltaic solar cells could be a good entropic source.

A PV solar cell was connected to the ADC of the microcontroller. The output of the ADC is processed for analysis. The experiment setup was put under a light chamber that facilitates constant light source and isolation from the external light source. Figure 4 shows the histogram plot for the PV solar cell sensor for a total of 100,000 samples. The sensor follows the normal distribution. Important observations from Figure 4 are summarized as follow:



s

Fig. 4: PV solar cell sensor voltage histogram.

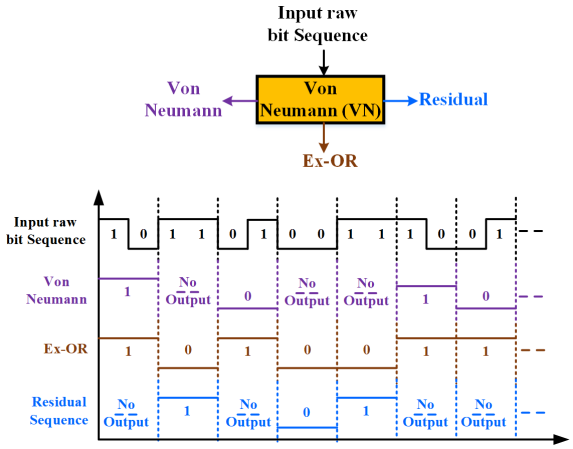
- The PV solar cell follows near-normal distribution for sampled voltages.
- The voltage samples within one SD (σ) around mean (μ) value can be utilized to calculate the average value, that would be relatively more stable. Later, it can be useful to calculate PUF response bits as per algorithm in [10].
- The successive voltage samples other than one SD (σ) around mean (μ) value would be useful to generate raw binary bits.

IV. Iterative Von Neumann (IVN) Processing for TRNG

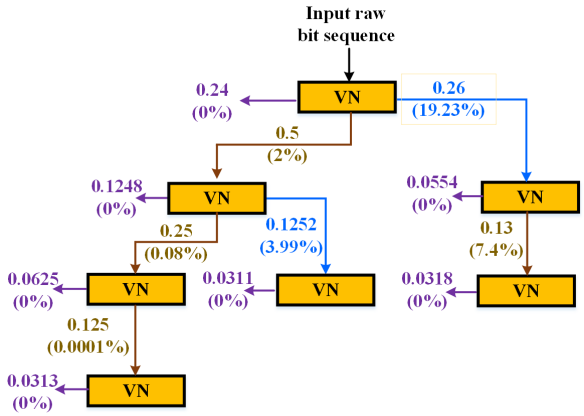
A practical entropic source produces random bits 1 and 0 with unequal probability p and q respectively with some bias n. The number of unbiased bits is equal to npq and is far less than achievable entropy bound. The bias makes the extraction of TRNG bits very difficult and depends upon sampling interval between two samples and environmental factors, such as external lighting, temperature, or humidity. The following equation originally described in [9] is used to calculate the bias:

$$n = \frac{|p - q|}{2} \times 100\% \quad (1)$$

The bias among initially generated raw bits arises due to a higher correlation between successive samples. High bias leads to rejection of raw bits, and therefore an optimum bias is desirable. The bias



(a) Von Neumann block and corresponding waveforms of output sequences.



(b) IVN tree structure using Von Neumann blocks. The value at output sequence is throughput with respect to reference value 1 (input bit sequence) and corresponding value in bracket indicates bias in percentage.

Fig. 5: Iterative Von Neumann (IVN) schematic [9].

Algorithm 1 Input bit stream bias adjustment

procedure BIAS-ADJUST($bits, S, \delta$)
  bits[ ] $\leftarrow$ Array of initial bits
  S $\leftarrow$ Set initial sampling Interval
  $\delta \leftarrow$ Set step value
  while $S \neq 10\%$ **do**
    Generate 1000 sample bits in bits[ ]
    Calculate bias n
    if N >10% **then**
      $S = S - \delta$
    else if N <10% **then**
      $S = S + \delta$

value 10% is a good balance between throughput and Shannon entropy per bit [9]. The Algorithm 1 is part of the training mode. It sets the bias value equals to 10% by adjusting sample interval (time difference between two successive samples). The process begins with generating 1000 raw bits and calculating the bias. Then, bias value is checked if it is >10% then subtract step value δ else add step value δ in initial step interval.

After setting the bias value 10%, the prototype switches to run mode to generate true random bits [9]. The initial input raw bit sequence is fed to the IVN tree structure realized using 7 Von Neumann blocks illustrated in Figure 5b. The Von Neumann debiasing, shown in Figure 5a, is a suitable technique for low-computing-power and low-memory devices due to its simpler operation. It rejects successive occurrence of bit sequence "11" or "00" in output Von Neumann sequence and accepts bit sequence "01" and "10" as a bit '0' and '1' respectively. However, a single Von Neumann can extract throughput only up to 24%, a substantial loss of the bits.

Therefore, it becomes important to process entropy present in discarded bits. As implied in the name IVN, we process the Ex-OR and residual sequence to extract entropy present in them. We made some design choices to accommodate prototype for computing resource constraint platforms. First, we limit the structure up to 7 VN blocks as additional VN blocks would not result in much throughput improvement. Second, the residual sequence is processed only at two blocks, where the bias in residual sequences is relatively higher. The final TRNG outcome is produced by concatenating the output from all Von Neumann sequences and has $\approx$ 33.69% better throughput than a single Von Neumann block.

V. A Specific Case Study

The TRNG and PUF are inherently orthogonal in operation, therefore, the metrics to measure the performance characteristic are quite different. Further, the change in light intensity can alter the electrical

parameters of the photovoltaic solar cell. Thus, the change in light intensity is a useful environmental condition to vary to test the performance. An ideal design should work well at every light intensity. The experimental set up was placed inside a light chamber that facilitates the change in light condition from light intensity 0 W/m^2 (extreme dark) to 90 W/m^2 (very bright sunlight).

A. PUF Performance Testing

The reliability and uniformity metrics are used to measure the performance of the proposed PUF prototype. The PV solar sensors and microcontroller set up were put inside the light chamber and PUF output bits were recorded in PC.

1) Reliability: The reliability metric is the measure of the deviation of the PUF bit response with the reference response. It uses the hamming distance and is a measure of the reproducibility of PUF response with reference response. The following equation was first used to calculate reliability, R of n-bit PUF response

$$\mathrm{R} = 100\% - \frac{1}{M}\sum_{m=1}^{M}\frac{HD\left(R_{ref}, R_{m,t}\right)}{n} \times 100\% \tag{2}$$

The light intensity at 50 W/m^2 (corresponding to normal room light intensity) was considered as the reference point. The Hamming Distance (HD) measures that how many bits are different between reference response, $\mathrm{R_{ref}}$ and response generated at different light condition $\mathrm{R_{m,t}}$. Figure 6 shows the measured reliability at different light intensity. The proposed design has worst-case reliability of 92.13% at light intensity 90 W/m^2 and average reliability of 92.13% for light intensity variation from 50 W/m^2 to at 90 W/m^2.

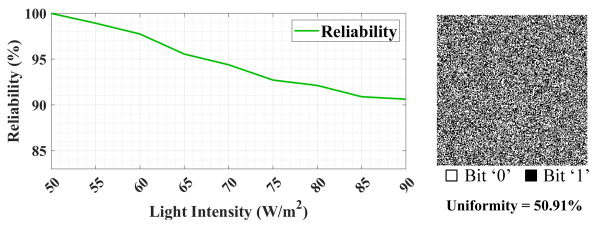


Fig. 6: Reliability and Uniformity as a measure of PUF performance metric.

2) Uniformity: The uniformity measures the proportion of 0 and 1 in PUF response. The ideal PUF response should have 50% uniformity, i.e. in 128-bit PUF response, the number of 0-bits and 1-bits should be 64.

The light intensity 50 W/m^2 was considered as reference and 12 different readings were taken at an interval of 1 hour. The worst-case uniformity is 47.66% and the best case uniformity is 50%. The average uniformity was measured at 50.91%, i.e. very close to the ideal value.

B. TRNG Performance Testing

The ideal TRNG should work independently of ambient light conditions. The quality of a random number is measured by different tests, e.g. NIST STS, DieHard, AIS, and TestU01. Among, them NIST STS is most widely used by researchers [4], [7], thus we preferred it in our work. The NIST STS is a collection of 15 tests that a true random sequence should satisfy.

The different tests in NIST STS check the number of occurrences of bits 1 and 0, find the run length, i.e. the number of consecutive occurrences of bit 1 or zero, checking linear dependence among the sub-string of, periodicity of occurrence in given length and ability to compress the sequence. Each test is measured by calculating "p-value", which indicates the confidence of randomness. A test with a p-value of $\geqslant 0.01$ indicates the test is passed and with 99% confidence. The nature of each random test is different and hence, the criteria for each test are also different. We collected a total of 100 random number sequences, where each sequence consists of 1 million random bits with light varied at the random interval to simulate the real-world scenario. Further, we exposed some sensors to light and some sensors were blocked during the data collection procedure. The minimum pass rate for each test other than random excursion and random excursion variant is 96 out of 100. The criteria to pass the random excursion and random excursion variant test are 65 out of 69 random bit sequences.

Table I lists the results of each NIST test in terms of the p-value, the proportion of the test passed and the result of the test as either pass/fail. The number in the bracket next to each test denotes, number of sub-tests. The proportion simply indicates how

TABLE I: NIST STS for TRNG performance evaluation. Result is 'pass', if p-value > 0.01.

Test name	Exhaustive Test		
	p-value	Proportion	Result
Frequency	0.845629	0.9900	**Pass**
Block frequency	0.451279	1.0000	**Pass**
Cumulative sums (forward)	0.152695	0.9900	**Pass**
Cumulative sums (reverse)	0.847926	0.9900	**Pass**
Runs	0.562478	1.0000	**Pass**
Longest run	0.384567	0.9900	**Pass**
Rank	0.747956	1.0000	**Pass**
FFT	0.859674	1.0000	**Pass**
Non-overlapping template (148)	0.501324	0.9937	**Pass**
Overlapping template	0.569541	0.9800	**Pass**
Universal	0.659841	0.9900	**Pass**
Approximate entropy	0.356947	1.0000	**Pass**
Random excursions (8)	0.846259	0.9927	**Pass**
Random excursions variant (18)	0.395846	0.9943	**Pass**
Serial-1	0.756185	0.9900	**Pass**
Serial-2	0.869416	1.0000	**Pass**
Linear complexity	0.231567	1.0000	**Pass**

many random sequences passed for the test, with 1 indicating all 100 or 69 random bit sequences have cleared the test. The proposed prototype passes all tests with a very high p-value, with the lowest 0.231567 and the highest 0.845629. The average p-value for all the tests is 0.45.

VI. Conclusion

The research work in this article proposes an integrated design of TRNG and PUF using PV solar cells and microcontroller. We have shown that the voltage response of PV solar cells can be engineered in static (stable) and dynamic (large variation) response. The segregation is based on dividing the PV solar cell histogram within or outside one SD (σ) around the mean voltage value (μ). The proposed prototype uses Iterative Von Neumann (IVN) structure which has $\approx$ 33.69% better throughput to generate true random bits. The proposed prototype achieves an average 92.13% reliability and 50.91% uniformity in PUF response. The integrated TRNG-PUF architecture can be beneficial in space-limited CPS.

Acknowledgment

The research was partially supported by National Science Foundation (NSF) grant: 1738662.

References

[1] S. P. Mohanty, "Advances in transportation cyber-physical system (t-cps)," *IEEE Consumer Electronics Magazine*, vol. 9, no. 4, pp. 4–6, 2020.

[2] S. P. Mohanty, "Consumer electronics is the driver of smart cars notes from the editor," *IEEE Consumer Electronics Magazine*, vol. 7, no. 5, pp. 3–55, 2018.

[3] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," *Computers in Industry*, vol. 100, pp. 212–223, 2018.

[4] M. Stipčević and Ç. K. Koç, "True random number generators," in *Open Problems in Mathematics and Computational Science*. Springer, 2014, pp. 275–315.

[5] G. Honan, N. Gekakis, M. Hassanalieragh, A. Nadeau, G. Sharma, and T. Soyata, "Energy harvesting and buffering for cyber physical systems: A review," *Cyber-Physical Systems: A Computational Perspective*, pp. 191–217, 2015.

[6] M. Varchola, M. Drutarovsky, and V. Fischer, "New universal element with integrated puf and trng capability," in *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, 2013, pp. 1–6.

[7] S. K. Satpathy, S. K. Mathew, R. Kumar, V. Suresh, M. A. Anders, H. Kaul, A. Agarwal, S. Hsu, R. K. Krishnamurthy, and V. De, "An all-digital unified physically unclonable function and true random number generator featuring self-calibrating hierarchical von neumann extraction in 14-nm tri-gate cmos," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1074–1085, 2019.

[8] S. Larimian, M. R. Mahmoodi, and D. B. Strukov, "Lightweight integrated design of puf and trng security primitives based on eflash memory in 55-nm cmos," *IEEE Transactions on Electron Devices*, vol. 67, no. 4, pp. 1586–1592, 2020.

[9] V. Rožić, B. Yang, W. Dehaene, and I. Verbauwhede, "Iterating von neumann's post-processing under hardware constraints," in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2016, pp. 37–42.

[10] C. Labrado and H. Thapliyal, "Design of a piezoelectric-based physically unclonable function for iot security," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2770–2777, 2019.

About the Authors

Amit Degada is currently pursuing his Ph.D. in Electrical and Computer Engineering at the University of Kentucky, Lexington, KY, USA. Contact him at amit.degada@uky.edu.

Himanshu Thapliyal is an Associate Professor and Endowed Robley D. Evans Faculty Fellow with the Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA. Contact him at hthapliyal@uky.edu.