Harnessing Uncertainty in Photoresistor Sensor for True Random Number Generation in IoT Devices

Amit Degada and Himanshu Thapliyal
VLSI Emerging Design And Nano Things Security Lab (VEDANTS-Lab)
Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA
Corresponding Author: hthapliyal@uky.edu

***Abstract*—Internet of Things (IoT) has facilitated the connection of many smart devices via internet. Recent cyberattacks have shown that resource constrained IoT nodes are easy prey that lead towards compromising the secrecy of the data and vulnerabilities could be exploited remotely to take control of safety-critical systems. Photoresistor sensors have applications in IoT systems, such as smart street lighting, intelligent cameras, light activated smart consumer electronics, smart home, smart healthcare, etc. Building hardware security primitives, such as True Random Number Generator (TRNG), based on the intrinsic properties of photoresistor would be a novel direction to develop cost-savvy IoT security primitives. Therefore, this paper proposes a TRNG prototype that is devised from uncertainty presents in photoresistor sensors. The proposed TRNG prototype does not require any complex interfacing for preprocessing the weak signal, thereby reducing the unnecessary delay and the recurring hardware cost. The proposed prototype employs the novel approach of additive scrambling that aids to sample sensors at a higher rate. The proposed TRNG has an average random bit generation rate of 8 kbps that is better than the recent work in the literature. The quality of randomness was validated by 15 test batteries of NIST STS test.**

***Index Terms*—Hardware security, Photo-sensor, Light Dependent Resistor (LDR), Cryptography, Cybersecurity, NIST test.**

I. Introduction

We are on the verge of experiencing significant improvement in quality of life with the inclusion of Internet of Thing (IoT) which empowers all kind of sensors nodes to interact and process the data via internet [1]. The lighter computing capability, smaller memory and limited power budget of sensor nodes makes the inclusion of additional cybersecurity primitives exceptionally troublesome in existing fabric. Therefore, one intriguing research direction is creating security solutions out of already present components such as sensors [2], [3]. A true random number generator (TRNG) is a hardware component that generates a string of random bits based on non-deterministic physical phenomenon as a source of randomness. TRNG is crucial in many cybersecurity operations, such as asymmetric and symmetric key generation, Digital Signature (DS) creation, initialization vector in block and stream ciphers and salt value generation for secure storage [4].

In the existing literature, free-running oscillators (FRO), ASIC and FPGA based TRNGs have been explored [4]. Considering the growing importance of the IoT nodes, TRNG designs using existing sensors and microcontrollers have also been explored by researchers in recent time. Figure 1 illustrates a generalized design of a sensor based TRNG. The existing sensor based TRNGs have explored sensors, such as accelerometer [5] [6] [7], fuel cell [8], hydrogen gas [9], inertial measurement unit (IMU) [10], ECG [11] and RFID [12]. The existing sensor TRNGs have a preprocessing module in order to sample-amplify-filter raw sensor signal [5] [8] [9] [11] [12] or to remove stationery patterns [6] or to add randomness in raw sensor signal [10], before being utilized to extract random bits by post-processing algorithms. The inherent properties of the sensors and preprocessing modules in existing sensors based TRNG makes them suffer from lower random bit generation rate. For example, the maximum average random bit generation rate in sensors based TRNG is 250 bps [7], to the best of our knowledge. As sensors based TRNG can be designed with minimal redesign costs and minimal performance, area, and power penalties, thereby a faster implementation needs to be developed.

Fig. 1: TRNG - General Schematic

In this work, we have identified photoresistor sensor as a potential candidate to design faster TRNGs. The photoresistor, also known as Light Dependent Resistor (LDR) and photocell sensor is a two terminal, low cost sensor, which exhibits change in electrical resistance proportional to ambient lighting conditions. The photoresistor sensor plays pivotal role in many IoT applications, such as in detecting change in lighting conditions to activate lights in smart homes and smart street lighting, phototaxic navigation in robotic tadpoles, control brightness and contrast in smart televisions, designing heartbeat sensors in smart healthcare, calculating shutter speed in smart cameras, light activated control circuitry in smart consumer electronics and designing detector for infrared astronomy, infrared spectroscopy and optical coding [13] [14]. Building hardware security primitives, such as True Random Number Generator (TRNG), based on intrinsic properties of photoresistor would be a novel direction to develop cost-savvy IoT security primitives.

Therefore, this paper proposes a TRNG prototype that is

devised from uncertainty present in photoresistor sensor. The proposed TRNG prototype does not require any complex interfacing for preprocessing, thereby reducing the unnecessary delay and the recurring hardware cost. The proposed prototype employs the novel approach of additive scrambling that aids to sample sensors at higher rate, consequently producing a faster random bit generation rate. The proposed TRNG has an average random bit generation rate of 8 kbps which to our knowledge is better than the recent work in the literature. Additionally, the proposed TRNG works satisfactorily from dark light to normal sunlight and sudden change in ambient light condition does not affect performance. We used the US based National Institute of Standard and Technology (NIST) recommended Statistical Test Suite (STS), also known as NIST SP 800-22 as a measure of randomness [15].

This paper is organized as follows. The section II gives background of photresistor sensors. Section III presents an evaluation of photoresistor sensor as source of randomness and describes electronic hardware and software framework of proposed photoresistor sensor based TRNG. The experimental setup and results are discussed in section IV. The section V concludes the paper.

II. Background

The key electrical characteristics for photoresistors to note are dark resistance, light resistance, rise time and fall time. The dark resistance of a photoresistor is measured at 10 seconds from removal of the light. The light resistance is electrical resistance measured at a light intensity of 10 lux. The photoresistor shows wide difference between both values, for example, photoresistor PDV-P8104 has dark resistance value 2 MΩ and light resistance has variation in values ranging from 27 kΩ to 60 kΩ. The rise time and fall time for photoresistors are time taken to settle to stable value for light and dark resistance respectively. Their values are typically in millisecond (for photoresistor PDV-P8104, rise time and fall time are 60 ms and 25 ms respectively [14]).

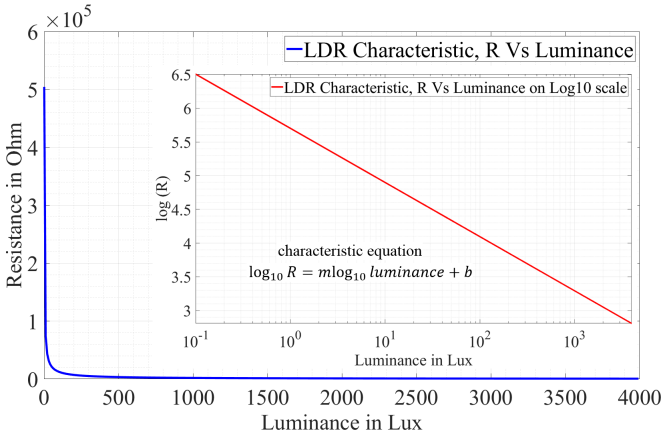

Fig. 2: LDR Characteristics, R Vs Luminance

To understand the property of photoresistor further, we modeled the characteristic of photoresistor in normal scale from known characteristic equation of logarithmic scale. We determined slope m= -1.245 Ω/lux from datasheet [14] and constant b = 7.10 Ω from many practical readings. It can be observed in figure 2 that the photoresistor has highly skewed electrical characteristic.

The Analog to digital Converter (ADC) in modern microcontroller used in IoT application supports high sampling frequency (for example, the Tiva™ C series TM4C123GH6PM microcontroller supports sampling rate from 125k to 1M samples/second). The non-linear characteristic of photoresistor and slow response time with respect to microcontroller ADC sampling rate would be a good choice to design TRNG.

III. Architecture of Proposed Photoresistor Based TRNG

In this section, we evaluate the uncertainty of photresistors as a measure of randomness and explains electronic hardware and software framework for proposed TRNG. We assume that microcontroller's CPU operation and its memory is attack resilient.

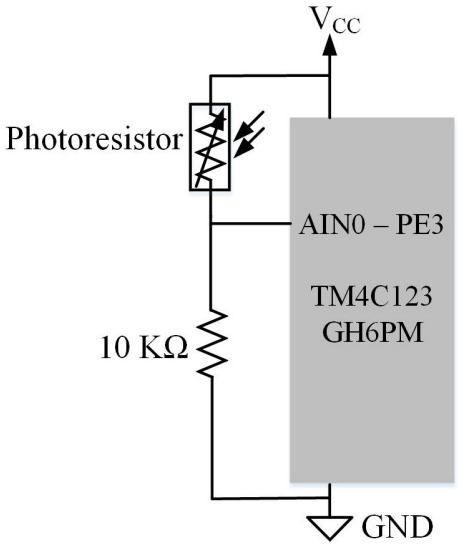

Fig. 3: Photoresistor-microcontroller Setup to study histogram of sampled voltage

A. Evaluation of Randomness in Photoresistor

A practical ADC in a microcontroller has Root Mean Square (rms), quantization and code-transition noise. Our software framework does not involve any steps to remove such ADC noises. Thereby, we hypothesized that voltage across a photoresistor sensor sampled by microcontroller ADC can result in variations across some of the least significant bits (LSBs). The variation in LSBs due to slow response time, non-linear characteristic and ADC noise serve as a practical proof of randomness.

To evaluate randomness, we setup (as shown in figure 3) a voltage divider circuit using a photoresistor sensor and 10 kΩ resistor connected between supply and ground voltage of a microcontroller(Tiva™ C series TM4C123GH6PM in our

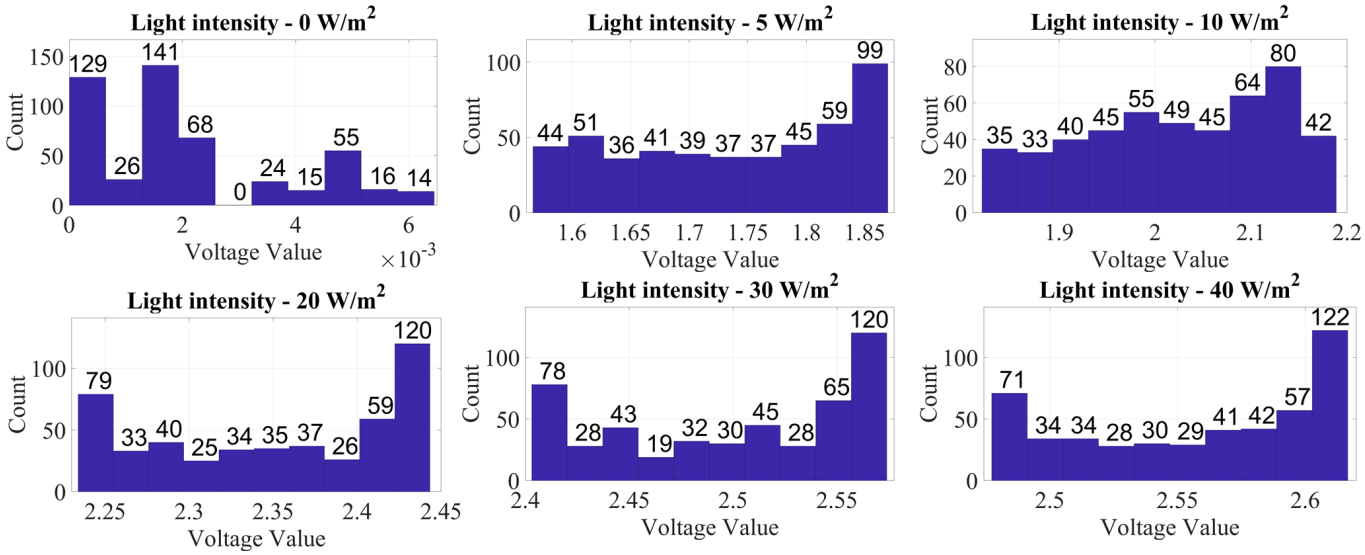

Fig. 4: Histogram of photoresistor sensor voltage at different light intensity

setup). The ADC of the microcontroller was configured to sample the change in voltage across photoresistor sensor and supply voltage. The setup was placed inside a light chamber that facilitates change in light intensity from 0 W/m^2 (Extreme Dark) to 40 W/m^2 (Normal Surrounding light). Figure 4 shows the histogram of 488 voltage readings across a photoresistor sensor. It can be observed from Figure 4 that the histogram has near uniform voltage distribution (over $\approx$ 200 mv range) at every light except for a slightly skewed distribution at 0 W/m^2(extreme dark). Hence, the photoresistor sensor has a prospect to work as a source of randomness. The technique to mitigate close distribution of photoresistor voltage at 0 W/m^2 is explained later in Section III-B.

B. Electronic Hardware

The electronic hardware of proposed TRNG is shown in Figure 5. It has eight photoresistor sensors connected with the ADC pins of an ARM Cortex-M4 microcontroller (we implemented prototype using PDV-P8104 photoresistor and Tiva^{TM} C series TM4C123GH6PM microcontroller). The photoresistor sensors in IoT nodes are usually configured in arrays and manufacturing process variation causes each sensor to respond differently, even at same ambient light condition. This results in more uncertainty and can help to achieve faster random bit generation rate. The interfacing to PC is done via Universal Asynchronous Receiver-Transmitter (UART) to transmit random bits for NIST STS testing. The UART interfacing can be omitted in a standalone TRNG system.

C. Software Framework

The software framework for the proposed TRNG basically implements post-processing algorithms to extract 128-random bits. The goal of the software framework was to make it as simple as possible to get adapted in resource constrained embedded computing nodes in IoT. The following major operations are performed in software framework:

1) Sampling: The 12 bit ADC has 0.8 mv resolution, which is sufficient to detect typical variations of 200 mv in photoresistor voltage at given light intensity. The sampled voltage of same sensor is compared with previous sampled voltage of sensor to generate bit. The bit is '0', if the sampled ADC voltage is the same or less, otherwise it is '1'.

2) Debiasing: The first bits generated from sensor voltages are usually biased and debiasing is required in order to produce the random bits. There exist several debiasing techniques, for example, cryptographic hash functions, deterministic extractor functions, resilient functions and correcting functions [4] [12]. In this research work, the von Neumann correcting function is chosen due to its lower computing and memory requirements. These properties are highly desirable in order for the proposed TRNG to get adapted for implementation in lighter IoT nodes. The von Neumann correcting function rejects any successive occurrence of bit "00" and bit "11". The bit sequence "01" and "10" is accepted as bit '0' and bit '1' respectively.

3) Ex-OR with LFSR output bits: This part is the key and core of the TRNG framework. When sensors response is changing slowly and sampled at high frequency, we observed that in worst case they may produce raw bit strings of repeated "10" or "01" of some bit length. It would result into uninterrupted sequence of identical bits (i.e. long runs) at the output of von Neumann correcting function. The true random source is expected to have runs of 0s and 1s of different length with expected frequency. However, too many or too small length of runs and with high frequency results into poor randomness. One possible solution to reduce long biased bits is to sample the sensors at low rate, however it could result into lower random bit generation rate.

In our proposed prototype, we do Ex-OR operation of two 128 bit chunks: First from the output of the von Neumann debiasing function and other from software implemented 32 bit maximum length Linear feedback Shift Register (LFSR). We

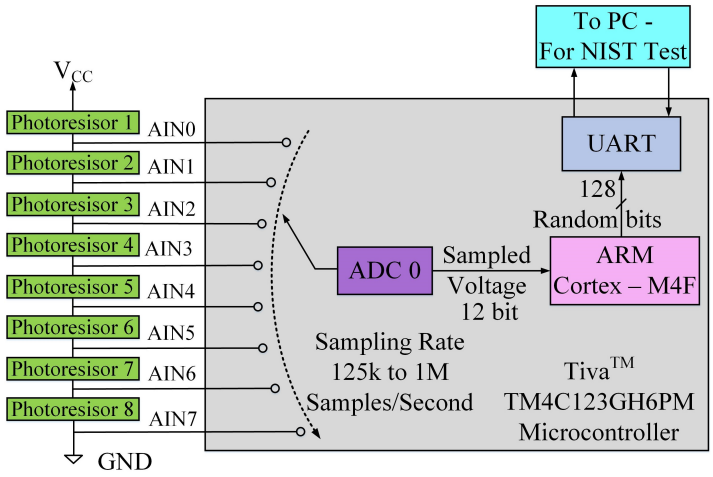

Fig. 5: Hardware setup of proposed TRNG

chose 32 bit LFSR because it has large period of 4294367295 bits and can provide different 128 bit chunks at each Ex-Or operation. We will show in next section that this method fixes the lower entropy at dark light condition and removes the problem of long runs of 1s and 0s.

IV. Experimental Setup and Results

The ideal TRNG should work independently from changes in the quantity being sampled. The proposed TRNG was subjected to varying light condition inside a light chamber, as shown in Figure 6. The NIST STS has 15 tests to validate randomness [15]. We collected 1 million bits (one sequence) at light intensity 0 W/m^2 (extreme dark) to 40 W/m^2 (normal sunlight). A test with p-value ≥ 0.01 indicates that sequence is random with 99 % confidence and test is passed.

Fig. 6: Experiment Setup

The efficacy of the additive scrambling approach can be checked in two ways, first by entropy measurement and second by passing NIST tests. Figure 7 uses Shanon's entropy equation to plot entropy for random bit sequence and validate the claim of improved entropy at dark light. The entropy after additive scrambling is ≈ 1, that indicates high uncertainty. Further, it is evident from the first two test results in Table I that additive scrambling helps to pass all 15 NIST STS test.

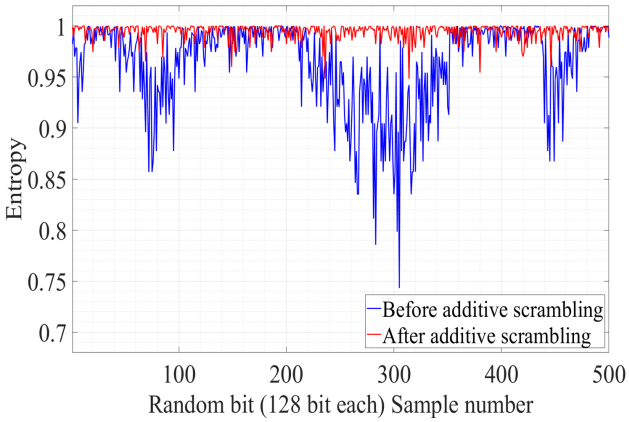

Fig. 7: Effect of additive scrambling on Entropy at light intensity 0 W/m^2

Each NIST STS test checks random bit sequence for unique purpose, which is listed in NIST guideline [16]. Due to abrasive nature of the random bit sequence, the criteria for each test is different. The non-overlapping template, random excursion and random excursion variant test has 148, 8 and 18 sub-tests respectively. The p-value for above tests is an average value of all sub-test in Table I. Further, to perform random excursion and random excursion variant test, the random bit sequence should pass frequency test and has number of cycles greater than 500 [16]. The term not applicable (n/a) in Table I point out that relevant test is skipped due to insufficient number of cycles by test suite.

The exhaustive test for randomness verification was performed to find anomalous behavior of the proposed TRNG. We collected 100 sequences to subject to NIST STS testing. The natural environment was simulated by randomly varying light condition from extreme dark to full at random interval.

TABLE I: NIST Statistical Test Suite results. Result is 'pass', if p-value > 0.01

Test name	p-value Ex-OR LFSR		p-value at different light intensity					Exhaustive Test		
	without	with	0 W/m^2	10 W/m^2	20 W/m^2	30 W/m^2	40 W/m^2	p-value	Proportion	Result
Frequency	***0.000000***	0.156099	0.227307	0.923738	0.083471	0.939268	0.548669	0.816537	0.9900	**Pass**
Block frequency	***0.000000***	0.927501	0.954847	0.739690	0.828306	0.241850	0.343236	0.851383	1.0000	**Pass**
Cumulative sums (forward)	***0.000000***	0.126651	0.279898	0.762967	0.058918	0.989132	0.535035	0.096578	0.9900	**Pass**
Cumulative sums (reverse)	***0.000000***	0.249778	0.248341	0.673262	0.075123	0.965337	0.489175	0.514124	0.9800	**Pass**
Runs	***0.000000***	0.866566	0.870771	0.105751	0.246265	0.446113	0.559371	0.455937	1.0000	**Pass**
Longest run	***0.000000***	0.864463	0.687594	0.873904	0.096038	0.435841	0.842273	0.289667	1.0000	**Pass**
Rank	***0.004280***	0.250500	0.669781	0.903934	0.608650	0.281791	0.964394	0.759756	0.9900	**Pass**
FFT	***0.000000***	0.453635	0.264045	0.556518	0.217423	0.812934	0.756450	0.978072	0.9800	**Pass**
Non-overlapping template (148)	***0.000000***	0.469966	0.488747	0.481182	0.501738	0.511771	0.551411	0.501522	0.9879	**Pass**
Overlapping template	***0.000000***	0.567448	0.898972	0.332220	0.266956	0.883922	0.557932	0.494392	0.9800	**Pass**
Universal	***0.000000***	0.762748	0.101559	0.273205	0.096837	0.883723	0.189590	0.383827	0.9900	**Pass**
Approximate entropy	***0.000000***	0.753883	0.630872	0.045073	0.016413	0.529153	0.720651	0.534146	1.0000	**Pass**
Random excursions (8)	***n/a***	*n/a*	0.720551	0.559315	*n/a*	0.507115	*n/a*	0.421557	0.9927	**Pass**
Random excursions variant (18)	***n/a***	*n/a*	0.482817	0.425737	*n/a*	0.537126	*n/a*	0.225245	0.9943	**Pass**
Serial-1	***0.000000***	0.888561	0.464811	0.041445	0.983962	0.448933	0.342303	0.637119	0.9800	**Pass**
Serial-2	***0.000000***	0.584569	0.214004	0.311381	0.882605	0.545856	0.189946	0.616305	1.0000	**Pass**
Linear complexity	0.381449	0.968967	0.956401	0.767418	0.835116	0.327585	0.011104	0.171867	1.0000	**Pass**

The additive scrambling and exhaustive test was performed at varying light intensity from extreme dark to normal sunlight.
The minimum pass rate for each statistical test with the exception of the random excursion and random excursion variant test is 96 for 100 binary sequences.
The minimum pass rate for the random excursion and random excursion variant test is 65 for 69 binary sequences.

Further, we obstructed light falling on any randomly chosen sensors at random time during operation. Table I lists the p-value for exhaustive test and is practical proof that the prototype can tolerate change in ambient light. The data rate for random bit generation was speculated using SysTick Timer in microcontroller for 100k samples. The average random bit generation rate for the proposed TRNG is 8 kbps, which is higher than the current maximum reported (250 bps) in [7] among sensor based TRNG, to the best of our knowledge.

V. Conclusion

The proposed TRNG has a simple electronic hardware and software framework that is suitable for integration in existing photoresistor based IoT node. The Ex-OR operation of row sensor signal and LFSR is a novel method that helps to achieve high entropy ($\approx$1) and pass all mandatory NIST STS tests to examine randomness. Additionally, this method helps to achieve faster random bit generation rate. The proposed photoresistor based TRNG works satisfactorily at light conditions varying from extreme dark to normal sunlight and can tolerate random changes in light intensity. To the best of our knowledge, the average random bit generation rate of 8 kbps of the proposed prototype is better in current sensor based TRNG research. The proposed TRNG can be used in IoT cryptographic operation such as key generation in symmetric and asymmetric encryption, creation of Digital signature and random vector in stream cipher.

Acknowledgment

The research in this paper was partially supported by National Science Foundation (NSF) under grant no: 1738662.

References

[1] H. Thapliyal, "Internet of things-based consumer electronics: Reviewing existing consumer electronic devices, systems, and platforms and exploring new research paradigms," *IEEE Consumer Electronics Magazine*, vol. 7, no. 1, pp. 66–67, Jan 2018.

[2] 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, 2018.

[3] C. Labrado, H. Thapliyal, S. Prowell, and T. Kuruganti, "Use of thermistor temperature sensors for cyber-physical system security," *Sensors*, vol. 19, no. 18, p. 3905, 2019.

[4] M. Stipčević and Ç. K. Koç, *True Random Number Generators*. Cham: Springer International Publishing, 2014, pp. 275–315. [Online]. Available: https://doi.org/10.1007/978-3-319-10683-0_12

[5] J. Voris, N. Saxena, and T. Halevi, "Accelerometers and randomness: Perfect together," in *Proceedings of the Fourth ACM Conference on Wireless Network Security*, ser. WiSec '11. New York, NY, USA: ACM, 2011, pp. 115–126. [Online]. Available: http://doi.acm.org/10.1145/1998412.1998433

[6] S. L. Hong and C. Liu, "Sensor-based random number generator seeding," *IEEE Access*, vol. 3, pp. 562–568, 2015.

[7] G. Revadigar, C. Javali, W. Xu, A. V. Vasilakos, W. Hu, and S. Jha, "Accelerometer and fuzzy vault-based secure group key generation and sharing protocol for smart wearables," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2467–2482, Oct 2017.

[8] C. Erbay and S. Ergün, "Random number generator based on fuel cells," in *2018 New Generation of CAS (NGCAS)*, Nov 2018, pp. 98–101.

[9] C. Erbay and S. Ergïn, "Random number generator based on hydrogen gas sensor for security applications," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug 2018, pp. 709–712.

[10] Y. Sun and B. Lo, "Random number generation using inertial measurement unit signals for on-body iot devices," in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, March 2018, pp. 1–9.

[11] C. Camara, P. Peris-Lopez, H. Martín, and M. Aldalaien, "Ecg-rng: A random number generator based on ecg signals and suitable for securing wireless sensor networks," *Sensors*, vol. 18, no. 9, 2018.

[12] C. Kösemen and G. Dalkiliç, "Designing a random number generator for secure communication with wisp," in *Proceedings of the International Conference on Compute and Data Analysis*, ser. ICCDA '17. New York, NY, USA: ACM, 2017, pp. 289–292.

[13] B. Haraoubia, *Nonlinear Electronics 1: Nonlinear Dipoles, Harmonic Oscillators and Switching Circuits*. Elsevier, 2018.

[14] "Pdv-p8104 datasheet."

[15] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Booz-Allen and Hamilton Inc Mclean Va, Tech. Rep., 2001.

[16] "Random bit generation - guide to the statistical tests," 2019. [Online]. Available: https://csrc.nist.gov/Projects/Random-Bit-Generation/Documentation-and-Software/Guide-to-the-Statistical-Tests