On the Effectiveness of Electric Network Frequency (ENF) as a Source of Randomness

Girish Revadigar
Huawei International Pte. Ltd.
Singapore
girish.revadigar@huawei.com

Chitra Javali
*Institute for Infocomm Research (I2R), A*STAR*
Singapore
chitra_javali@i2r.a-star.edu.sg

Yang Li
College of Mechatronics and Control Engineering
Shenzhen University
China
yli@szu.edu.cn

Sreejaya Viswanathan
Advanced Digital Sciences Centre (ADSC)
Illinois at Singapore Pte. Ltd.
Singapore
sreejaya.v@adsc-create.edu.sg

***Abstract*—Random bit generator (RBG) is one of the most crucial part of cryptographic applications. In many systems, due to unavailability of true randomness source(s), deterministic/pseudo-random bit generators (DRBG) are used to generate pseudo-random bit-strings. However, recent studies have shown that, if the seed used to initialise DRBGs is obtained from a source producing low entropy bits, it can lead to serious security incidents. Hence, identifying reliable and pervasive source of (true) randomness has been an active area of research. In this work, we investigate the suitability of a ubiquitous source of randomness produced in a power grid's electric network, called Electric Network Frequency (ENF), for building a noise source for RBGs. ENF fluctuates around its mean value due to unpredictable power grid system loads. We build a low cost ENF recording device, and conduct extensive analysis of the captured dataset. Our results show that ENF exhibits good randomness properties and can be employed to build noise source for RBGs.**

***Index Terms*—noise source, electrical network frequency, randomness**

I. Introduction

Random Bit Generator (RBG) is the basic building block of cryptographic systems designed to provide security. The random bit-strings generated are used as nonces, passwords, unique strings for digital signatures, initialisation vectors, and hash functions etc. Generally, it is assumed by the system designers that the participating devices have access to a source of perfect randomness to construct the above security primitives. However, this assumption is not always true and/or most applications may not have access to the reliable sources possessing true randomness, e.g., thermal noise produced in electrical circuits [1], radio-active element's decay process, or timing information about the clicks of Geiger counter, etc [2]–[4]. Thus, Deterministic/Pseudo Random Bit Generators (DRBG/PRBG) initialised with a secret and truly random *seed*, are employed to generate the pseudo-random bit-strings. Since it is difficult to extract large amounts of entropy from the above sources, complex mathematical algorithms are used to mix the randomness and generate the output with sufficient entropy. As an alternative, researchers have proposed hardware-based random bit generators to draw higher entropy [5], [6]. However, these solutions increase the hardware complexity and can serve specific applications only. Moreover, the integration of specialized hardware will be cumbersome for existing and legacy devices. Many IoT devices extract randomness from various sensors such as cameras, accelerometers, microphones, etc., for random bit generation. However, these methods have limited scope and are subject to environmental contexts, and hence, are vulnerable to attacks [7].

In order to provide highest possible security for cryptographic applications ranging from small scale to high end systems, identification of a source of (true) randomness that can be easily accessed by almost all the devices of IoT ecosystem is a crucial challenge that must be addressed. Hence, identifying reliable and pervasive source of (true) randomness has been an active area of research from many decades till date. In this paper, we systematically address the above mentioned challenge. We investigate the suitability of a natural and ubiquitous source of randomness produced in the power grid's electric network, called as Electric Network Frequency (ENF), for designing a reliable noise (randomness) source suitable for RBGs. ENF fluctuates around its mean value, typically 50 Hz, or 60 Hz depending on the geographical region, and can be observed in the whole electric network [8]. This variation is caused due to varying power demand by the consumers, and is unpredictable. Thus, every device connected to power line can directly record this frequency variation. Figure 1 shows the architecture of a typical power generation and distribution network.

Our Contributions: In this work, we design and build a microcontroller-based simple hardware prototype to capture

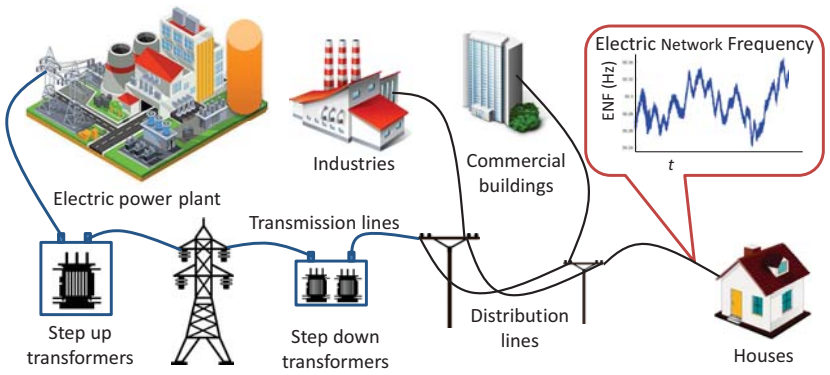


Fig. 1: The architecture of a typical electric power generation and distribution network. Due to varying demand for electric power by consumers, the plant controls generation and distribution process, which in turn leads to the fluctuation of electric frequency across the network, a.k.a., electric network frequency (ENF).

the ENF signal. We conduct extensive experimental analysis over a large ENF dataset and the observations reveal that the ENF signal exhibits good randomness properties and can be used to build a noise source for random bit generation. We propose the design and implementation of first RBG component/construction that exploits the randomness of ENF, a noise source called ENF-NS, that digitizes the ENF signal to produce random bit-strings.

ENF signal is ubiquitous, hence, our ENF-based noise source can be used to design public entropy sources widely used in many cryptographic applications, similar to other well known publicly available entropy/randomness sources e.g., NIST entropy beacon service [9] and Random.org [10] etc. All the above randomness sources e.g., [9] publish their random bit strings online, and any number of applications/systems can use these public random bit-strings for seeding/reseeding of the RBGs together with other mandatory secret personalization strings/inputs recommended by NIST [11], [12], during initialization of RBGs. To the best of our knowledge, we are the first to evaluate the randomness properties of ENF to design noise source. We believe that our findings will lead to many more research directions in the area of random bit generation using ENF or its derived signals.

The rest of this paper is organized as follows: Section II provides the details about related work, and Section III presents the preliminaries. Section IV explains about ENF signal collection and evaluation of its randomness properties. The design, implementation, and evaluation of ENF-based noise source are presented in Section V. The practical use cases of our ENF-NS are discussed in Section VI, and Section VII presents the conclusion and future work.

II. Related Work

True random bit generators require a source of true randomness whose output/behaviour is unpredictable. The sources of randomness can be either external or internal with respect to the generator. The random number generation service by NIST [9] uses two commercially available sources of randomness that employ quantum effect to generate truly random bits. The Internet service HotBits also provides random numbers generated using radio active decay process [2]. Another service by Random.org uses environmental noise as the source [10]. The above methods [2], [9], [10] use specialized equipment/hardware and software, which is non-trivial to set up by the end users/non-experts. However, the applications can only request and use a part of the random bits generated by these services as one time seed for DRBGs.

Harvesting randomness originated within the electrical circuits/material processes, in the form of burst noise, thermal and flicker [1], [4], [6], is another method of generating random bits. The researchers in [7] have investigated using on-board sensors of many personal devices like laptops, mobile phones etc., as entropy sources. The results show that accelerometer sensor is a reliable source, whereas, the other sensors e.g., microphone, camera, thermometer, photometer, proximity sensor, magnetometer, can be easily controlled by an adversary.

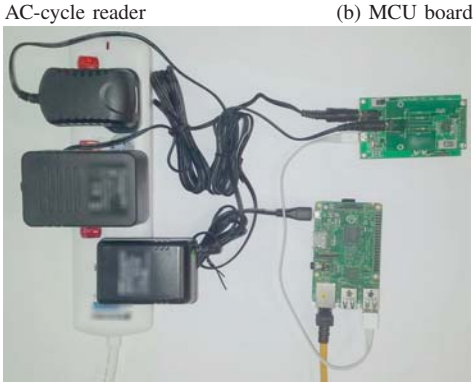
Prior work have employed ENF for time stamping, clock synchronization in distributed networks [8], and multimedia forensics [13]. Our work presented in this paper is entirely different from all the prior work. We advance the state of the art with our novel contribution – we systematically study the ENF signal and its randomness by collecting a large dataset and conclude that it can be used as a reliable and ubiquitous source of randomness. We propose the design, implementation and evaluation of ENF-based noise source ENF-NS. Our results demonstrate that ENF exhibits sufficient randomness to be employed as an input to the noise source.

III. Preliminaries

In this section, we present useful metrics to evaluate the quality of generated random bits and basic process of random bit generation.

(a) AC-cycle reader

(b) MCU board



(c) Experimental setup

Fig. 2: ENF measurement setup: Our prototype consists of (a) AC-cycle to square wave converter, and (b) MCU board. MCU board is connected to a Raspberry Pi via USB as shown in (c) to record real-time ENF data continuously.

A. Evaluation Metrics

- Entropy of a bit-string: It is the product of the number of bits and entropy per bit in a bit-string [11].
- Full entropy: A bit-string is said to have full entropy bits, if the entropy of the bit-string is equal to the number of bits, i.e., each bit will have highest entropy, similar to the bit-string generated by an ideal (true) random bit generator [11].
- Pearson correlation coefficient ($\mathcal{P}$): The correlation coefficient $\mathcal{P}$ of two sets $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ and $\mathcal{B} = \{b_1, b_2, \ldots, b_k\}$ is given by [14]:

$$\mathcal{P} = \frac{\sum_{i=1}^{k}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{k}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{k}(b_i - \bar{b})^2}} \quad (1)$$

where $\bar{a} = \frac{1}{k}\sum_{i=1}^{k} a_i$, and $\bar{b} = \frac{1}{k}\sum_{i=1}^{k} b_i$. $\mathcal{P}$ varies between -1 and 1, where 1 indicates perfect correlation, and -1 and 0 indicate anti-correlation and no correlation, respectively.

IV. Experimental Setup And Evaluation of Randomness Properties of ENF

Electric Network Frequency is an inherent feature of any electric power generation and distribution network, as shown in Figure 1. As explained earlier, the ENF signal fluctuates around its mean value e.g., 50 or 60 Hz, depending on the geographic location, due to varying load or demand by the consumers in the network [8]. The load variations will be always random and unpredictable, and hence, the measured

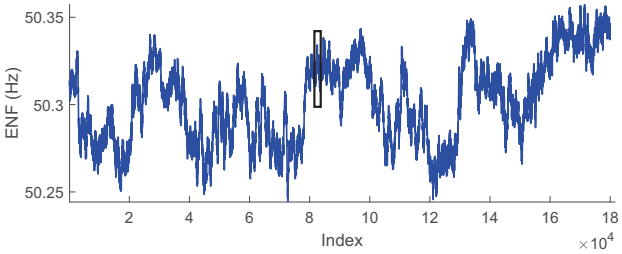
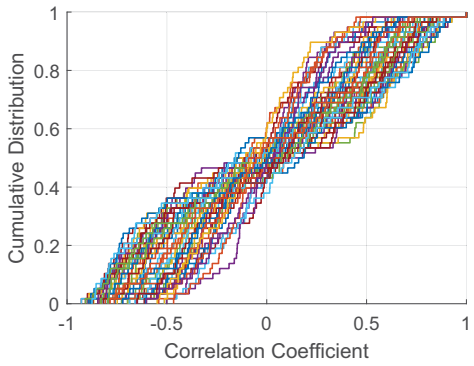
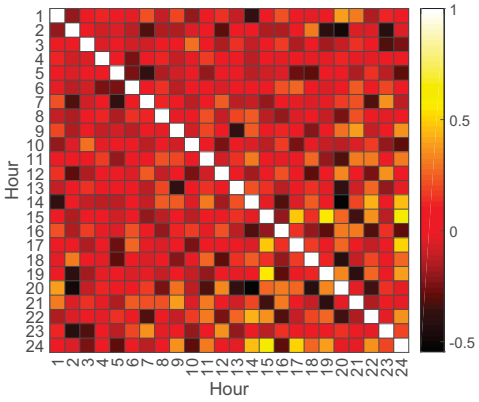


Fig. 3: Raw ENF captured for one hour at a location.



(a) 1 Hour



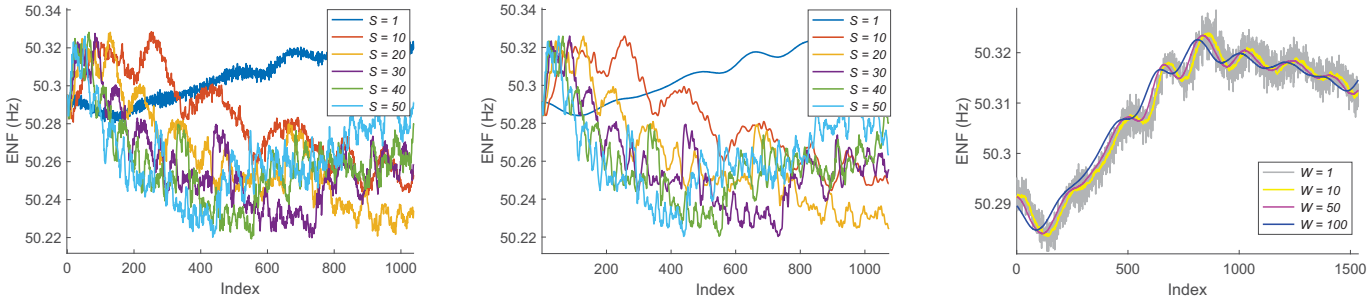
(b) 1 Day

Fig. 4: The correlation of (a) original ENF signal captured for each minute in an hour, and (b) per-second ENF captured for each hour in one day.

ENF signal will also be random. In the following sub-sections, we discuss ENF collection and randomness evaluation method.

A. Experimental Setup

We have designed a simple low cost ENF measurement hardware prototype having a small form factor. Figures 2a and 2b show our boards designed for capturing ENF. The circuit board shown in Figure 2a conditions and converts the AC voltage signal to square wave signal of same cycle length. The second board shown in Figure 2b is a microcontroller (MCU) based signal reader. The square wave signal is sensed by the MCU via interrupts, and using an internal timer module,



(a) $W = 1$ (b) $W = 50$ (c) $S = 1$

Fig. 5: Effect of different skip size S on (a) original, and (b) per-second ENF, and (c) effect of W for $S = 1$.

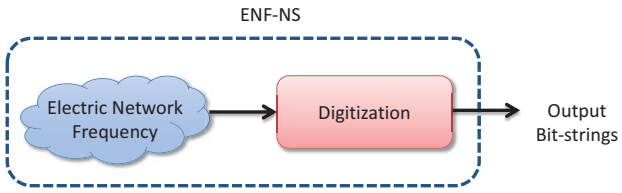


Fig. 6: ENF-based noise source model.

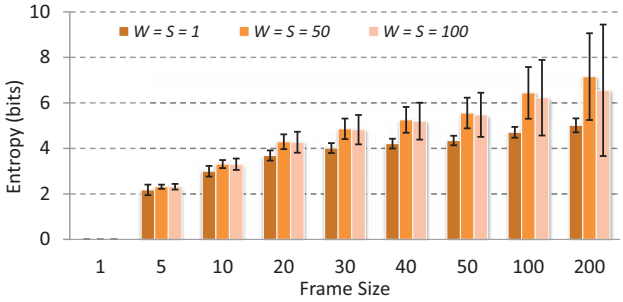


Fig. 7: Entropy of raw ENF samples showing the maximum, mean and minimum values.

the ENF cycle lengths are recorded locally or output via serial port/USB to an external device. Figure 2c shows the whole experimental setup. The square wave converter board and MCU board are stacked on each other using connector pins. The MCU reader board is connected to a Raspberry Pi via USB, that logs ENF data to a database server. We use off-the-shelf ac-to-ac adapter and a voltage divider to reduce the voltage and to provide the necessary electrical protection for our prototype. We use Matlab for processing the ENF-dataset and for implementing the proposed RBG component. Figure 3 shows the raw ENF captured for one hour.

B. Evaluation of the Randomness of ENF

We captured the ENF signal for 24 hours using our prototype and stored in a database. To evaluate the randomness properties of ENF signal, we analysed this dataset for the following different cases:

1) ENF Variation in an Hour: In order to check the variation of ENF in an hour, we processed the captured ENF for each hour separately. We sub-divided each 1-hour data into 60 sets of one-minute data, and calculated the correlation among them. Thus, for each hour, we obtained 3600 correlation values. Our results show that, a large portion i.e., 65–76% of the correlation values were < 0.5. About, 23–28% were between 0.5 and 0.75, and very few values, $\approx$2–4% were between 0.75–0.85. The highest correlation $\mathcal{P} = 1$ observed for each 1-hour data was only for 1.6667% of the values, i.e., only for auto-correlation of each 1-minute ENF data. It should be noted that, only $\mathcal{P} > 0.9$ is considered as good correlation among the signal slices [14]. This shows that ENF signal variation within an hour is quite random. Figure 4a shows the cumulative distribution of the correlation coefficient $\mathcal{P}$ calculated for one-hour ENF signal randomly selected from the experimental dataset.

2) ENF Variation in a Day: For analysing the per-day variation pattern, we grouped the data into 24 sets of one-hour data. Since the original ENF signal is highly fluctuating, in order to check the overall variation pattern in an hour, we processed the ENF by applying a moving window-based averaging method as follows. We considered a window size of W samples, and calculated the average ENF per window. Next, we shifted the window by a skip size of S samples, and repeated the process for complete ENF dataset. The sequence of per-window averages calculated gives the overall fluctuation of ENF. Figures 5a, 5b and 5c show the ENF processed for different W and S. Particularly, to compare ENF variation for different hours of the day, we set $W = S = 50$, which is same as the ENF mean value of 50 *Hz* in our region, to calculate (averaged) per-second ENF data. Our results show that the maximum correlation observed for different hours of a day (24 hour data) was < 0.75. Hence, it is evident that, the ENF variation is random and does not repeat in a day. Figure 4b shows the correlation of per-second ENF ($W = S = 50$) of each hour data in a day. The diagonal cubes indicate auto-correlation $\mathcal{P} = 1$.

V. ENF-Based Noise Source (ENF-NS)

In this section, we present the design of ENF-based noise source (ENF-NS). Figure 6 shows the ENF-NS model. We

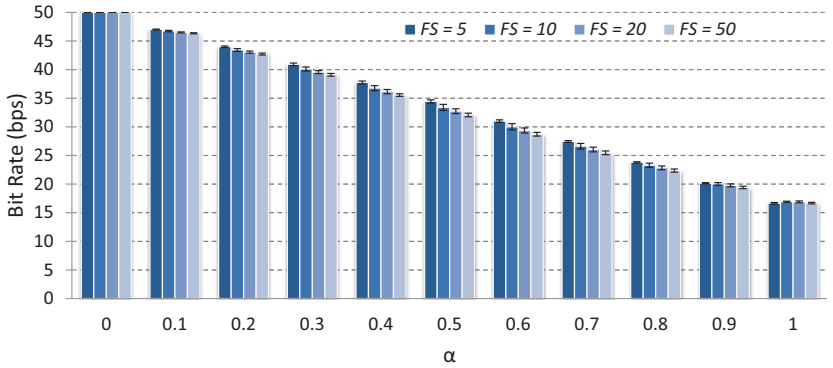


Fig. 8: Bit generation rate of raw ENF samples showing the mean and standard deviation. The maximum and minimum bit rates achievable are 50 and 18 bits/sec, respectively. This is also the rate at which the entropy can be collected from ENF-NS.

consider the raw ENF signal recorded using our prototype at 50 *Hz* sampling frequency as a noise-source signal. We process this noise source signal further using a *digitization* method to obtain binary bit-strings. Before applying a particular quantization method, it is essential to check the randomness/entropy supported by the signal being quantized. In other words, entropy determines the upper bound on the number of bits that can be extracted from each ENF-sample, such that the resulting bit-string exhibits sufficient randomness. We calculate the Shannon entropy H_s of raw ENF samples using a non-overlapping moving frame of size = *FS* samples. For a given ENF dataset (e.g., 1 hour data), we calculate H_s for different non-overlapping fixed frame lengths, i.e., *FS* = {1, 5, 10, 20, 30, 40, 50, 100, 200}. The procedure is repeated for all dataset captured. Figure 7 shows the H_s calculated for our dataset for different *W*, *S* and *FS*. As shown in Figure 7, for the original ENF signal (*W* = *S* = 1), $5 < FS < 20$ results in 2–4 bits/sample of entropy, and further increase in *FS* from 30 to 200 does not increase H_s drastically, and gives entropy in the range 4–5.5 bits/sample. For *W* = *S* = {50, 100}, the entropy increases by increasing *FS* from 20 to 200. The maximum entropy observed was ≈9.5 bits/sample in both the cases. The above analysis shows that H_s gives the maximum limit for the number of bits that can be extracted per ENF sample during quantization.

A. Design and Implementation

In the current work, we employ a threshold-based quantization mechanism [14], [15] for the *digitization* of ENF signal. We consider a non-overlapping moving frame = *FS* samples, and in each frame, we calculate two thresholds as follows:

$$Th^{+} = \tilde{m} + \alpha\tilde{s}; \qquad Th^{-} = \tilde{m} - \alpha\tilde{s} \tag{2}$$

where $\tilde{m}$ and $\tilde{s}$ are the mean and standard deviation of samples in the frame, and α is a control parameter that varies between 0 and 1. Each ENF sample is encoded as either bit 1 or 0 based on whether the sample value is greater than Th^{+}, or less than Th^{-}, respectively. The samples that lie between two thresholds, called as *guard space*, are discarded. The ENF-NS produces chunks of fixed length bit-strings e.g., 512 bits. The length of output is reconfigurable.

B. Evaluation

In this section, we evaluate the ENF-NS and randomness properties of its output. Figure 8 shows the bit generation rate of ENF-NS, for different α and frame size *FS*. Increasing α from 0 to 1 increases the guard space and also the number of samples discarded during quantization, hence, the bit rate decreases. However, this feature is desirable to improve the entropy of the bit-strings. It can be observed that, the bit rate is approximately the same for all *FS* for a given α, and the maximum and minimum bit rates obtained are 50 and 18 bits/sec, respectively. Depending on the application requirements, this bit rate can be increased by employing other multi-bit quantization methods [16]. It should be noted that, the bit rate of ENF-NS is not the bit rate of random bit generator, instead, this is the rate at which entropy can be collected from the output of ENF-NS, that can be used as an input to ENF-ES.

VI. Discussion

Since ENF is available to all the devices that are connected to the electric network of same power grid, the following question may arise: "*Is it secure to employ ENF as a source of randomness for random bit generation? Or, whether an adversary capturing ENF simultaneously with a legitimate device can produce same random bits using ENF-NS as noise source for his RBG?*" ENF is safe to use as a source of randomness despite its ubiquitous nature. However, it should be noted that, *the output of ENF-NS should never be used as secret keys for encryption/cryptographic applications*, instead, should be used only as input to the next block *entropy source* of RBGs, or for seeding/reseeding the RBGs together with other proper mandatory secret initialization strings as recommended by NIST [11], [12]. To be specific, the ENF-NS output can be considered as public noise/entropy source, similar to other publicly available sources e.g., NIST entropy

beacon service [9]. The NIST website also strongly recommends not to use the entropy beacon strings published online as secret keys, instead, to use them for the initialization of RNGs only. As per NIST standards [11], [12], the random bit-strings from public sources together with other mandatory secret inputs/strings are used to first generate the internal *seed*. These secret user inputs/strings need not have high entropy. The internal *seed* generated will be secret and highly random, which is later used to instantiate the RBG. As a strict rule, all the RNGs must be instantiated with different entropy inputs (e.g., bits from public entropy sources) and mandatory secret initialization strings/inputs. If the RBGs implemented are as per NIST specifications, the output bits of these RBGs cannot be differentiated from true random bit string using statistical analysis [11], [12].

Thus, even if the adversary captures same ENF signal as a legitimate device, the bit strings generated by adversary's RNGs and legitimate devices' RBGs will be totally different since they are instantiated separately using different secret user inputs. It is worth noting that, the RNGs properly instantiated and running on the same device will also produce different random bit-strings.

VII. Conclusion and Future Work

In this paper, we have thoroughly investigated the randomness properties of ENF signal using a large dataset collected by our low-cost and simple hardware prototype. As per our observations, ENF is highly fluctuating and exhibits very good randomness properties, and thus, can be used as a reliable source of randomness. We have presented the novel design, implementation, and extensive evaluation of a noise source ENF-NS exploiting ENF signal as a source of randomness. Our analysis shows that the bit-strings generated by ENF-NS exhibit high randomness and hence, can be safely employed in RBGs.

In our future work, we would like to propose the construction of other blocks of RBGs i.e., entropy source, DRBGs and true random bit generators (NRBGs) based on proposed ENF-NS and other derived signals of ENF viz., electro-magnetic radiation (EMR) etc., for battery operated devices, evaluate the effectiveness, and compare with other commercial/public RBGs/NRBGs.

Acknowledgement

The authors would like to thank Prof. David You, SUTD, Singapore, and Prof. Rui Tan, NTU, Singapore for their valuable suggestions.

References

[1] K. Lundberg, "Noise Sources in Bulk CMOS," http://web.mit.edu/klund/, Accessed: 15-Jan-2019.

[2] "HotBits: Genuine random numbers, generated by radioactive decay," https://www.fourmilab.ch/hotbits/, Accessed: 15-Jan-2019.

[3] G. B. Agnew, "Random Sources for Cryptographic Systems," in *Proc. International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT)*, 1987.

[4] R. C. Fairfield, R. L. Mortenson, and K. B. Coulthart, "An LSI Random Number Generator (RNG)," in *Proc. Advances in Cryptology (CRYPTO)*, 1985.

[5] Apple, "iOS Security," Apple Inc., Technical Report, Feb 2014.

[6] B. Jun and P. Kocher, "THE INTEL RANDOM NUMBER GENERATOR," https://www.rambus.com/intel-random-number-generator/, Accessed: 15-Jan-2019.

[7] J. Voris, N. Saxena, and T. Halevi, "Accelerometers and Randomness: Perfect Together," in *Proc. ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2011.

[8] S. Viswanathan, R. Tan, and D. K. Y. Yau, "Exploiting Electrical Grid for Accurate and Secure Clock Synchronization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 14, no. 2, pp. 12:1–12:32, May 2018.

[9] "NIST Randomness Beacon," https://www.nist.gov/programs-projects/nist-randomness-beacon, Accessed: 15-Jan-2019.

[10] M. Haahr, "RANDOM.ORG: True Random Number Service," https://www.random.org, Accessed: 15-Jan-2019.

[11] M. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, and M. Boyle, *SP 800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation*. National Institute of Standards & Technology, 2018.

[12] E. Barker and J. Kelsey, *SP 800-90C: Recommendation for Random Bit Generator (RBG) constructions*. National Institute of Standards & Technology, 2016.

[13] R. Garg, A. L. Varna, A. Hajj-Ahmad, and M. Wu, ""Seeing" ENF: Power-Signature-Based Timestamp for Digital Multimedia via Optical Sensing and Signal Processing," *IEEE Trans. Information Forensics and Security (TIFS)*, vol. 8, no. 9, pp. 1417–1432, 2013.

[14] G. Revadigar, C. Javali, W. Hu, and S. Jha, "DLINK: Dual Link Based Radio Frequency Fingerprinting for Wearable Devices," in *Proc. IEEE Local Computer Networks (LCN)*, 2015.

[15] C. Javali, G. Revadigar, M. Ding, and S. Jha, "Secret Key Generation by Virtual Link Estimation," in *Proc. EAI International Conference on Body Area Networks (BodyNets)*, 2015.

[16] 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 (TIFS)*, vol. 12, no. 10, pp. 2467–2482, 2017.