IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON EMERGING APPROACHES TO CYBER SECURITY

Received March 22, 2020, accepted April 4, 2020, date of publication April 9, 2020, date of current version April 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986822

A True Random Number Generator Based on Gait Data for the Internet of You

CARMEN CAMARA[1], HONORIO MARTÍN[2], (Member, IEEE), PEDRO PERIS-LOPEZ[1], AND LUIS ENTRENA[2], (Member, IEEE)

[1]Department of Computer Science, University Carlos III of Madrid, 28903 Getafe, Spain
[2]Department of Electronic Technology, University Carlos III of Madrid, 28903 Getafe, Spain

Corresponding author: Carmen Camara (macamara@pa.uc3m.es)

This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Contract ESP-2015-68245-C4-1-P, in part by the Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation under Grant P2019-CARDIOSEC, and in part by the Comunidad de Madrid, Spain, under Project CYNAMON (P2018/TCS-4566), co-financed by the European Structural Funds (ESF and FEDER).

ABSTRACT The Internet of Things (IoT) is more and more a reality, and every day the number of connected objects increases. The growth is practically exponential -there are currently about 8 billion and expected to reach 21 billion in 2025. The applications of these devices are very diverse and range from home automation, through traffic monitoring or pollution, to sensors to monitor our health or improve our performance. While the potential of their applications seems to be unlimited, the cyber-security of these devices and their communications is critical for a flourishing deployment. Random Number Generators (RNGs) are essential to many security tasks such as seeds for key-generation or nonces used in authentication protocols. Till now, True Random Number Generators (TRNGs) are mainly based on physical phenomena, but there is a new trend that uses signals from our body (e.g., electrocardiograms) as an entropy source. Inspired by the last wave, we propose a new TRNG based on gait data (six 3-axis gyroscopes and accelerometers sensors over the subjects). We test both the quality of the entropic source (NIST SP800-90B) and the quality of the random bits generated (ENT, DIEHARDER and NIST 800-22). From this in-depth analysis, we can conclude that: 1) the gait data is a good source of entropy for random bit generation; 2) our proposed TRNG outputs bits that behave like a random variable. All this confirms the feasibility and the excellent properties of the proposed generator.

INDEX TERMS True random number generator, Internet of Things, gait data, entropy, randomness.

I. INTRODUCTION

The ever-growing number of IoT devices paves the way to become IoT technology an integral part of our lives soon [1], [2]. Multiple applications are benefiting from this technology ranging from industrial applications (e.g., Industry 4.0 [3]) to consumer/commercial applications such as smart homes or healthcare. Special attention deserves wearable devices that can be an essential part of which is known as the Internet of You [4], [5]. The Internet of You (IoY) is a new paradigm where sensor-based devices allow us to collect environmental data (e.g. temperature, location, etc.) and also biometric data such as heart rate or oxygen saturation level. These data can be combined to create more personal experiences, so the technology works for us, not the other way around. Nonetheless, the concerns about cybersecurity are a significant barrier and are hindering the adoption of IoT technology in the domains where sensitive data is at stake [6].

Securing this sensitive information in the IoT context is facing today, several challenges stemming from the wireless nature of the communication (insecure radio channel) and the scarcity of resources (e.g., memory and computation) in IoT devices. Hence, these constraints severely limit solutions offered by conventional cryptographic primitives, which turn too expensive to achieve in these resource-limited devices. Because of this, a new generation of cyphers, hash functions or true random number generators (TRNGs) devoted to the IoT environment is flourishing [7], [8] . In this article, we focus on the design of a TRNG that is a critical cryptographic primitive typically used to generate session keys, nonces or padding plain-texts. A TRNG usually is composed of (i) an entropy source from which we extract randomness,

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

(ii) a digitizer that transforms the entropy derived into a digital value and (iii) a post-processing module that usually unbiased the data [9]. The entropy source is the cornerstone of a TRNG because it limits the amount of randomness of the system and also affects the selection of the other TRNG's parts. Among the most used entropy sources for TRNGs stand out the solutions that include different kinds of electrical noise [10], [11], clock jitter and coherent sampling [12], metastability [13], [14] and chaos [15], [16]. A new trend, which exploits biosignals (vital information) as a source of entropy, is emerging in the IoT context [17], [18].

Using wearable body sensors approaches for key generation helps to overcome the stringent power and area constraints of the IoT environment. Deriving cryptographic keys from Electrocardiogram signals (ECG) is the greatest exponent of this trend [17], [19], [20]. Other alternatives exploit different sensor information such as the Galvanic Skin Response (GSR) [21], [22], the electroencephalogram (EEG) signal [23], the blood volume pulse or respiration [24]. Nevertheless, some sensor sources of information remain unexplored for key generation such as human gait data. Human gait data has some interesting features to become the noise source of a biometric random number generator. According to [22], it offers the following features: (i) fast sampling rate, (ii) simple data acquisition, (iii) high accuracy of measurement and (iv) variability of biological data.

In this work, we propose a novel TRNG based on human gait data that can be used for generating cryptographic keys for an Internet of You application. To that end, we will make use of a public dataset containing human gait information [25]. The contributions of this paper are innovative in several ways. For the first time in the literature (as far as we know), we propose the use of human gait data to generate random numbers. Secondly, we work in a transform domain via the Walsh-Hadamard for the randomness extraction and also offer lightweight post-processing. Thirdly, we have experimentally analyzed the quality of the generated random numbers by using the most exigent randomness test batteries, and even we have executed some other additional tests (e.g., NPCR and UACI coefficients).

We organize the rest of this paper as follows. In Section II, we introduce the dataset used in our experiments and describe the algorithm used to generate the random numbers. Section III presents the experimental results, including the results related to the quality of the entropy source and the quality of the random numbers generated. In Section IV some interesting implementation features of our proposal are described. Finally, some conclusions and recommendations are drawn in Section V.

II. METHODS AND MATERIALS

In this section we describe the Gait database used to conduct the experiments and the algorithm elaborated to extract the randomness from the gait information.

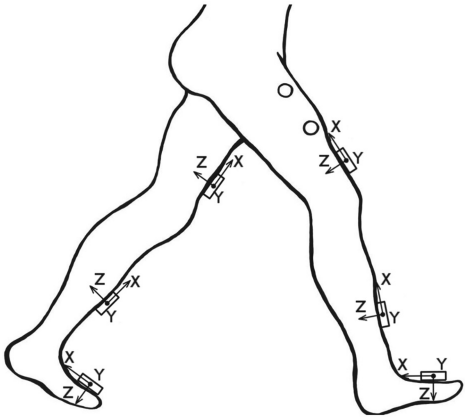

FIGURE 1. Loaction of sensors. Boxes represent inertial sensors while circles correspond to EMG sensors [25].

A. HUMAN GAIT DATABASE

Many gait databases are publicly accessible. These range from databases collected from healthy people [25], [26] to databases that contain gait data of patients affected with Parkinsons's disease [27], [28] or gait data collected after surgeries [29]. For this work, we have selected the HuGaDB because it provides human gait data in great detail compared to other published datasets [25].

More specifically, this database was initially intended for analysis and activity recognition. The information on 18 healthy participants (aged 23.67 ± 3.69) was recorded while performing a combination of activities. The dataset includes continuous recordings (around 10 hours in total) of different activities such as sitting, walking, bicycling, etc. The data provided has been obtained from six MPU9250 inertial sensors and two electromyography (EMG) sensors whose location is shown in Fig.1. Each inertial sensor provides the information of three-axis accelerometers and three-axis gyroscopes. In total, 38 signals per subject are accessible.

All the information provided by the sensors mentioned above is necessary for the activity recognition pursued in the original work. Note that the dependency between some sensors can be high because of the kinematics of the human motion itself. A high reliance can be a useful feature in some security applications like the identification of sensors belonging to the same host. On the contrary, entropy extraction can be negatively affected by this dependency since it reduces the freshness and non-predictability of the data.

About the quality of data input, on the one hand, gyroscope sensors are precise, but often a drift appears in the measurements. We can rid of this drift using a high-pass filter. On the other hand, accelerometers do not present drift but are a bit unstable. We need to pass these signals through a low-pass filter to smooth the signal. We urge the reader to consult [25] for details about pre-processing and characteristics of the signals recorded. In our experiments, we do not deem necessary do any extra pre-processing on the signals from HuDB dataset since our proposal aims to extract randomness from the captured data. Note that we could tolerate even small

levels of noise, which would not be the case if the purpose of our application was to determine the absolute orientation (a complementary filter might be useful in this case).

Finally and concerning the type of subjects, the patients do not have any pathology. Patients with pathologies such as Parkinson's (or any other movement disorder diseases) could be engaging because the uncontrolled movements caused by the disease can be useful (e.g., highly entropic) for randomness extraction. However, in our study, we prefer to use healthy patients since we can assume a similar behaviour (no-bias) among all subjects.

B. METHODS

The purpose of the proposed method is to extract randomness of the inertial sensors. In Algorithm 1, we summarise our proposed procedure. The algorithm is split into two main procedures 1) *GetEntropy*$(\cdot)$ and 2) *GetRandomness*$(\cdot)$, which are explain below.[1]

Algorithm 1 GAIT-TRNG

procedure GETENTROPY(Gait^{cleaned})
  Delete bad channels of $Gait^{cleaned}$
  for each Gait-observation **do**
    Error magnification between consecutive channels:
    $Gait^{mag}_{channel_i} = |chanel_i - chanel_{i+1}|$
    Concatenation : $Gait^{mag} = \left[Gait^{mag}_{channel_1}, \cdots, Gait_{channel_K}\right]$
  Split $Gait^{mag}$ into windows consisting on N samples
  for each Gyro-window $\left(x^{(j)}(t)\right)$ **do**
    Hadamard Transform:
    $\gamma^{(j)}_W(n) = abs\left(FWHT\left(x^{(j)}(t)\right)\right) (n = 1, \cdots N)$
    Quantization algorithm:
    $G^{(j)}_{(0,\ldots,7)}(n) = uint8 \times \left(\left(uint32\left(abs\left(Y^{(j)(n)}_W * \frac{10^5}{\pi}\right)\right)\right) >> 24\right)$
  Output the random bit stream G
procedure GETRANDOMNESS(G$_{rand}$)
  Segment the long G_{rand} input into vectors: $W^{(j)}$
  for each $W^{(j)}$ data chunk: **do**
    Split $W^{(j)}$ into two matrices: W_1 and W_2
    Extract Entropy:
    $R(i,j) = (W_1(i,j) \oplus ((W_1(i,j) < (j \bmod 8)) \oplus W_2(i,j)$
    Split R into two matrices: R_1 and R_2
    Final output:
    $O(i,j) = (R_1(i,j) \oplus ((R_1(i,j) < (j \bmod 8)) \oplus R_2(i,j)$
    Convert the matrix (O) into a vector and output
    the values

[1]The source code is available at: https://lightweightcryptography.com/?p=712

In our proposal, we focus on the 3-axis accelerometer and 3-axis gyroscope sensors; each pair constitutes an inertial sensor. We have six pairs of those sensors, that is, a total of 36 data channels. Note that in our experiments (TRNG analysis), we employ all the available channels since we need to generate moderately large files for the analysis. Nevertheless, the proposal is also feasible with a reduced set of sensors. In the *GenEntropy* procedure, the first step is the elimination of the bad channels that are those who present a high correlation between them. We have performed an intercorrelation analysis and eliminated those channels whose average correlation between channels is less than a threshold γ (i.e., $\gamma = 10^{-1}$ in our experiments). From this, we have eliminated the following list of channels: {5, 6, 11, 12, 17, 18, 29, 30}. After this, we have magnified the non-deterministic noise by computing differences (absolute value) between channels. At this point, we have concatenated all the good channels, and split the resulting data stream into windows $(x^{(j)}(t))$ of N-samples ($N = 100$ in our proposal). For the entropy extraction, we work in the Hadamard domain due to its compression capabilities and its low computational requirements. Using the fast Walsh-Hadamard transform, we only need to compute additions and subtractions, and its complexity is $\mathcal{O}(n \log n)$, being n the length of the data input [30]. Besides, it is a well-known approach for dealing with physiological signals [31], [32]. In our case, we truncate the output of the Hadamard transform to the first hundred coefficients. Then, for pulling out highly entropic bytes, we have used a quantifier. In particular, we use the eight Least Significant Bits (LBSs) of each Hadamard coefficient. Previous works, with other physiological signals [33], have inspired us for the usage of the LSBs, and we have confirmed their viability by experimentation. Mathematically, the procedure followed with each data window is described as below:

$$Y^{(j)}_W(n) = abs(FWHT(x^{(j)}(t)) \quad (n = \{1, \cdots N\}) \tag{1}$$

$$G^{(j)}_{(0,\ldots,7)}(n) = uint8((uint32(abs(Y^{(j)(n)}_W * \frac{10^5}{\pi}))) \gg 24) \tag{2}$$

Besides, it is frequent to use a post-processing algorithm to eliminate or reduce the statistical deficiencies of random bitstreams. In the case of stationary TRNGs that produce statistically independent bits with a constant bias, one of the most extended post-processing mechanisms consists of using an XOR compressor [34]. However, in [34] Ditch presented the H function, which is also a lightweight post-processing algorithm (16 XOR gates with two inputs) than in comparison to the XOR offers better performance (extracting higher entropy) for cases with high bias (see Fig.2). The H function uses XOR operations and circular left rotations, as shown below:

$$H(x_1, x_2) = (x_1 \oplus (x1 < 1) \oplus x2 \tag{3}$$

where "$\oplus$" symbolizes the bitwise XOR operation and "$<$" represents left circular shift rotation.

The *GetRandomness* procedure represents our proposed post-processing algorithm, and the output (G) of the

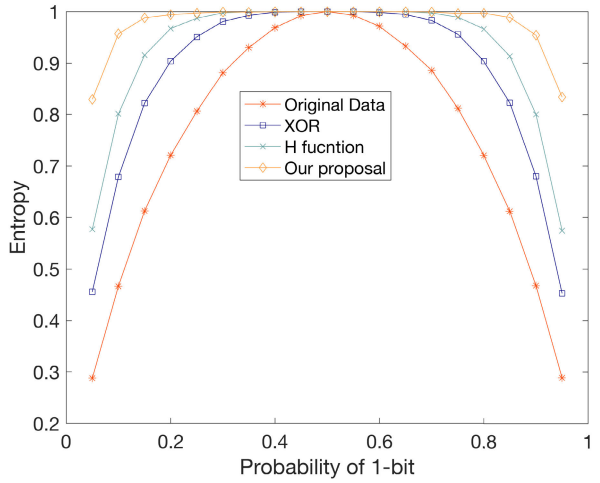

FIGURE 2. The entropy of post-processing functions.

GetEntropy procedure is its random input. For speeding the calculations, we split this long bit stream input into vectors (1×10^5), which are then reshaped into matrices $W^{(j)}$ of $P \times Q$ ($10^3 \times 10^2$ in our experiments) bytes. We divide each matrix into two halves (W_1 and W_2 with $P/2$ rows and Q columns respectively), and thereupon compute the H function over the elements of both matrices in the same position (row-i and column-j). The particularity in our proposal is that the positions rotate in the circular shift are set by the column j at play (i.e., $j \mod 8$). Mathematically,

$$R(i,j) = (W_1(i,j) \oplus ((W_1(i,j) < (j \mod 8)) \oplus W_2(i,j) \tag{4}$$

Then, and being somewhat conservative, and to guarantee an utterly random output without bias, we have repeated the previous process with the R matrix. Finally, this matrix is converted into a vector that represents the random bytes generated by our proposed TRNG. For testing the quality of our proposal, we have produced a file of 3.75 MB and analyzed the occurrence of each value. In detail, the probability of ones and zeros is 0.499924 and 0.500076, and the entropy is 1,0 (per bit)and 7.9999985 (per byte). Therefore, post-processing works appropriately. That is, the bits generated by our proposal are indistinguishable from those that would produce a perfect RNG. For completeness in Figure 2, we show the entropy of a file (16 MB in our experiments) for a set of probabilities of ones (and zeros) and under different post-processing algorithms (XOR, H function, our proposal). It is clear how our approach slightly outperforms the H function, which in turn surpasses the XOR.

III. RESULTS

We have subjected the True Random Number Generator to a thorough analysis. On the one hand, we assess the quality of the Gait data as a good source of entropy using the NIST SP 800-90B recommendation [35], [36]. On the other hand, and after post-processing, we analyze the randomness quality

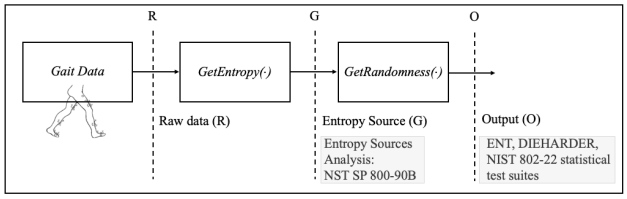

FIGURE 3. Entropy source and randomness output analysis.

of the proposed TRNG using well-established randomness battery of tests like DIEHARDER [37] or NIST 800-22 [38], [39]. Figure 3 sum ups the process and outlines the test suites used and each step. The tests mentioned above are very data demanding, so for testing purposes and verifying the feasibility of our design, we have created several large files from the HugaDB to perform an in-depth analysis of the proposed TRNG.

A. NIST SP 800-90B RECOMMENDATION: SOURCE ENTROPY ANALYSIS

Traditionally, the entropy source consisted of an analogue source of noise, for instance, the thermal noise of a diode Zenner. In our case, our analogue signal comes from six gyroscopes and six accelerometers (raw data "R" in Figure 3), and the "non-deterministic noise" has been magnified via computing differences between channels. Then, we compress the signal via Hadamard Transform. Finally, we extract bits using a quantization algorithm. The three above mentioned steps are part of the *GetEntropy* procedure, and we analysis its output "G" using the NIST SP 800-90B recommendation.

The probability that an adversary disclose a secret value at the first trial is linked with the min-entropy. Mathematically, assume X a discrete random variable that takes values defined in the set $\{x_1, x_2, \ldots, x_k\}$ with probability $P(X = x_i) = p_i$ $(i = 1, 2, \ldots, k)$, the min-entropy is defined as:

$$H = min_{1 \le i \le k}(-log_2(p_i)) \tag{5}$$

The NIST SP 800-90B recommendation computes ten entropy estimators:1) The Most Common Value Estimate; 2) The Collision Estimate; 3) The Markov Estimate; 4) The Compression Estimate; 5) The t-Tuple Estimate; 6) The Longest Repeated Substring (LRS) Estimate; 7) The Multi Most Common in Window Prediction Estimate; 8) The Lag Prediction Estimate; 9) The MultiMMC Prediction Estimate; and 10) The LZ78Y Prediction Estimate. Finally, the min-entropy represents the minimum value of these ten values.

In our experiments, we have generated a file of around 30 Mbit to conduct the analysis. In the Table 1 we summarize the obtained results. We can observe how the entropy for the majority of the tests is over 0.95 and in the worst case (LZ78Y prediction in which a dictionary with a maximum capacity of 65,536 words is built) is still a high entropy value (0.913). Therefore, the bits generated using the proposed *GetEntropy* procedure are suitable for cryptography.

TABLE 1. Min-entropy results (NIST SP 800-90B suite).

Method	Min-Entropy
Most Common Value Estimate	0.9967
Collision Estimate	0.9339
Markov Estimate	0.9951
Compression Estimate	0.9508
t-Tuple Estimate	0.9233
LRS Estimate	0.9127
MultiMCW Prediction Estimate	0.9966
Lag Prediction Estimate	0.9965
MultiMMC Prediction Estimate	0.9973
LZ78Y Prediction Estimate	0.9970
Overall estimation	0.9127

The entropy (H_I) calculated over a single and long sequence of 1s and 0s can produce an overestimated value. If the adversary has the chance to observe data sequences after restarts conditions, it could be beneficial to predict sequences after a new restart condition. The NITS recommendation defines a restart test to assess this issue. It requires 1000 restart conditions and 1000 values are stored each time. Then, all these values are concatenated, and the test checks if the estimated entropy is less than half of the min-entropy H_I obtained previously with a long sequence (0.9127 in our case). If so, the test is successful. If not, the validation fails. In our experiments, we simulate each reset condition by exposing the subject to an activity condition different from that of the current moment (e.g., sitting → walking). As shown, in Table 2, to have more certainty about the restart analysis, we have repeated the experiment five times. The results indicate that an adversary has no advantage by forcing a restart condition in the system.

TABLE 2. Restart tests (NIST SP 800-90B suite).

File ID	Result
File-1	Pass
File-2	Pass
File-3	Pass
File-4	Pass
File-5	Pass
Final min-entropy estimation	**0.913**

B. RANDOMNESS BATTERY OF TESTS: OUTPUT RANDOM ANALYSIS

Once we have verified that the proposed *GetEntropy* procedure produces a highly entropic output, we need to evaluate the randomness quality of the bits ("*O*" output in Figure 3) generated by the *GetRandomness* procedure. In a nutshell, this procedure is based on 1) bitwise XOR operations between matrices; and 2) bitwise circular shift operations over each element (byte) of a matrix (as described in Equation 4). As explained in Section II-B, the design of the post-processing procedure is inspired by the use of the H function due to its excellent properties to correct biased outputs.

As a very preliminary analysis, as display in Figure 4, we generate some bytes using our proposed TRNG and display these bytes using a scatter plot image (128 × 128 bytes). From visual inspection, we do not detect any anomaly and the picture looks like the one from a random variable.

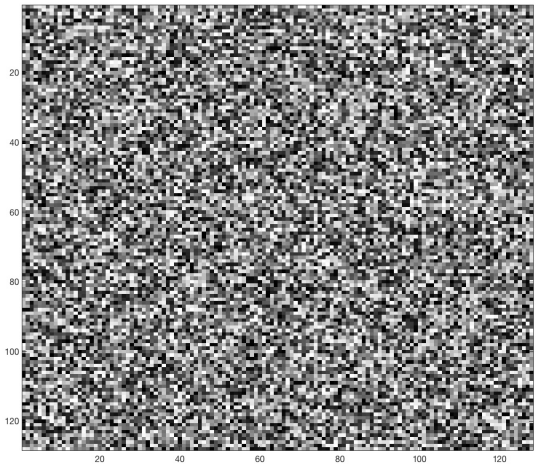

FIGURE 4. Random numbers generated by the proposed Gait-TRNG.

TABLE 3. ENT results.

Entropy	7.999985
Optimum compression	0 %
Chi square	244.55 (66.98 %)
Arithmetic mean value	127.5467
Monte Carlo π value	3.141160791 (error 0.01 %)
Serial correlation coefficient	0.000331

Then, we have scrutinized the proposed TRNG using well-known batteries of tests to assess randomness. For this, we have generated a large file of around 15 MBytes. We started using the ENT suite [40] since although it is not very exacting, it allows to discard bad/weak designs that commonly fails the Chi-square test. We summarise the results in Table 3 and all of them are almost perfect. For instance, the entropy is optimal, the correlation between values is minimal, there is no bias, and the chi-square test is successful (there is no suspicion of not being random).

As far as bias is concerned, we have performed an additional verification. In this test, we have evaluated the behaviour of the random numbers generated individually by each subject in the dataset. For this purpose, for each individual and each of her session recordings, we have generated a binary file. We have tested each file with the ENT suite. In Figure 5, we display the values obtained for the Chi-square test. Fortunately, there are no rare conditions, and the vast majority of the tests are between the mean value (255) and one (65% of the values) or two (resting 45%) standard deviations. It means that the quality of the random numbers generated is independent of the subject(s) used for the generation. That is, the subjects, and more particularly, the signals acquired from each subject behave similarly –in the sense that all of them looks like a random variable.

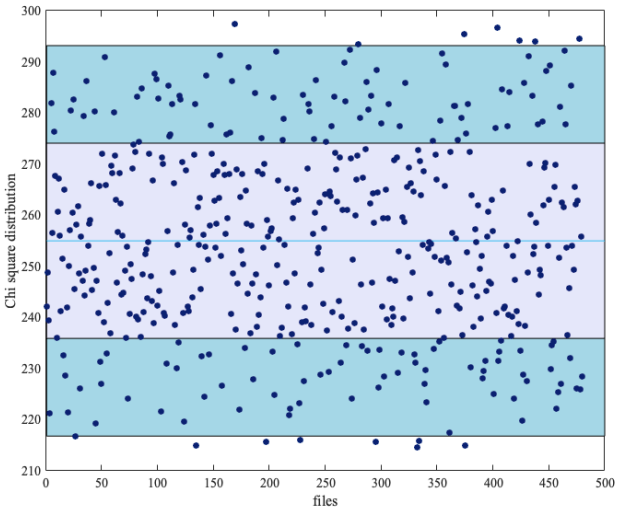

FIGURE 5. Bias analysis.

In 1995, Marsaglia proposed a battery of statistical test called DIEHARD to assess the quality of a random number generator. Later, Brown revised and extended the suite (named DIEHARDER [37]). The battery is composed of seventeen tests, and a p-value summarises each one of the tests. In detail, the software uses a Kolmogorov-Smirnov test to verify the uniformity in the interval [0,1] when several p-values are obtained in one of the tests. Assuming an extremely conservative criterion and in line with [37], a test is marked as ''pass'' if the p-value is within the interval [0.0050 - 0.9950], being 1% the significance level. Since many p-values are computed is not uncommon for some p-values to be out of this range: in our case, only two tests are marked as ''weak'' and the others are within the confidence interval (see Table 4a).

Finally, and to clear up any doubt about the randomness of the values generated, we have tested the binary file with the NIST 800-22 suite. Note that the NIST tests are very demanding and we can check the randomness quality of cryptography RNGs. The suite consists of fifteen tests, and several sequences are tested for each one. As shown in Table 4b, an overall p-value summarises each test. It also includes the number of sequences that successfully pass the test. In Figure 6, we display the minimum proportion of tests that should be got for various significance levels. For instance, 96 is the threshold for 100 sequences assuming a 1% of the significance level. We can verify that all the tests are over the required limit, and p-values are between 0.0050 and 0.9950 ($\alpha = 0.01$).

As a final check of all the tests, we have verified using a Kolmogorov-Smirnov test that all p-values (DIEHARDER and NIST results) are uniformly distributed (the overall p-value is 0.10446). From all this, we can conclude that the analyzed file looks like binary data generated by a random variable.

TABLE 4. DIEHARD and NIST tests. (a) Diehard results. (b) NIST results.

(a) Diehard results

Birthdays	0.98470 (PASSED)
OPERM5	0.01235 (PASSED)
32x32 Binary Rank	0.20338 (PASSED)
6x8 Binary Rank	0.03818 (PASSED)
Bitstream	0.95394 (PASSED)
OPSO	0.02755 (PASSED)
OQSO	0.01134 (PASSED)
DNA	0.01398 (PASSED)
Count the 1s (stream)	0.03893 (PASSED)
Count the 1s Test (byte)	0.68049 (PASSED)
Parking Lot	0.83992 (PASSED)
Minimum Distance (2d Circle)	0.99766 (WEAK)
3d Sphere (Minimum Distance)	0.79454 (PASSED)
Squeeze Test	0.21229 (PASSED)
Sum Test	0.04621 (PASSED)
Runs	0.89567 (PASSED) 0.38002 (PASSED)
Craps	0.00016 (WEAK) 0.01184 (PASSED)

(b) NIST results

Frequency	0.69931 (98/100) (PASSED)
Block Frequency	0.89776 (99/100) (PASSED)
Cumulative Sums	0.65753 (2/2) (98/100) (PASSED)
Runs	0.55442 (98/100) (PASSED)
Longest Run	0.43727 (99/100) (PASSED)
Rank	0.15376 (99/100)(PASSED)
FFT	0.88317 (98/100) (PASSED)
Non-Overlapping Template	0.90514 (148/148) ($>$ 98/100) (PASSED)
Overlapping Template Universal	0.36692 (98/100) (PASSED) 0.59555 (100/100) (PASSED)
Approximate Entropy	0.79814 (99/100) (PASSED)
Random Excursions	0.01943 ($>$ 52/53) (PASSED)
Random Excursions Variant	0.02686 (18/18) ($>$ 52/53) (PASSED)
Serial	0.9505 (2/2)) ($>$ 100/100) (PASSED)
Linear Complexity	0.07572 (98/100) (PASSED)

Overall Kolmogorov-Smirnov value	**0.10446**

C. ADDITIONAL TESTS

Although there is no uncertainty about the excellent behaviour of our proposed RNG, we have conducted two extra tests inspired by previous works [20], [41]. Firstly, we have studied whether there exits some correlation between the random numbers generated by each subject. Note that the goal is that the inter-correlation between individuals is zero (or close to zero). If not, the attacker could exploit the knowledge of a user-X to predict the random numbers produced by another user-Y. We have grouped the binary

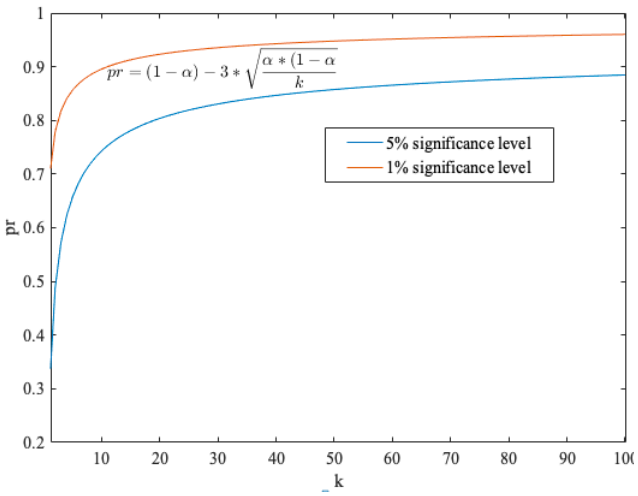

FIGURE 6. Proportion of sequences passing a test: NIST 800-22.

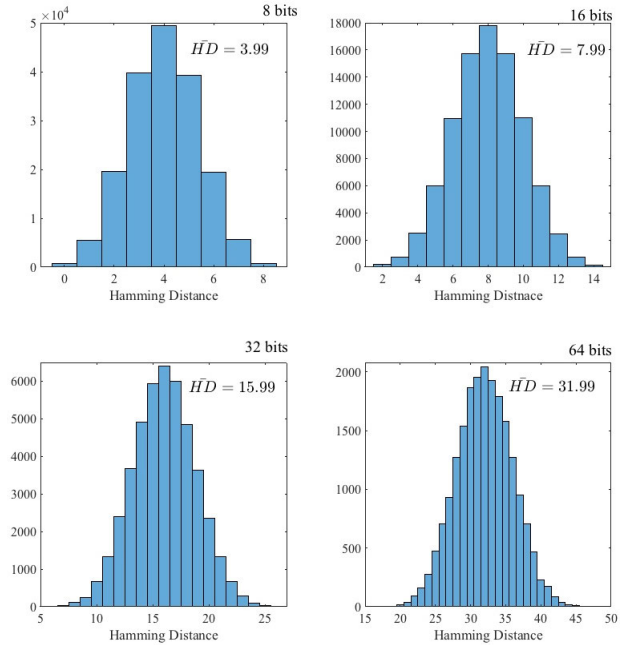

FIGURE 7. Hamming distance distribution.

data of each file (subject) into words, and then we compare the words between files via the hamming distance. Ideally, if the adversary has a zero advantage the hamming-distance between users has to follow a binomial distribution ($B(n, p)$), with mean $n \times p$ and variance $n \times p \times (1-p)$, where n represents the size word and $p = 1/2$ assuming the same probability for 1s and 0s. In Figure 7, we display the histogram for 8, 16, 32 and 64 bits word lengths. We can conclude that the advantage for an adversary is zero since the probability distributions are almost perfect with mean values of 3.99, 7.99, 15.99 and 31.99 respectively.

Finally, we have checked of the generated random bits as a keystream (K). Image a *one-time pad* cypher in which the cypher text C is computed by XORing the message M and the key K. Empirically, we display this procedure in

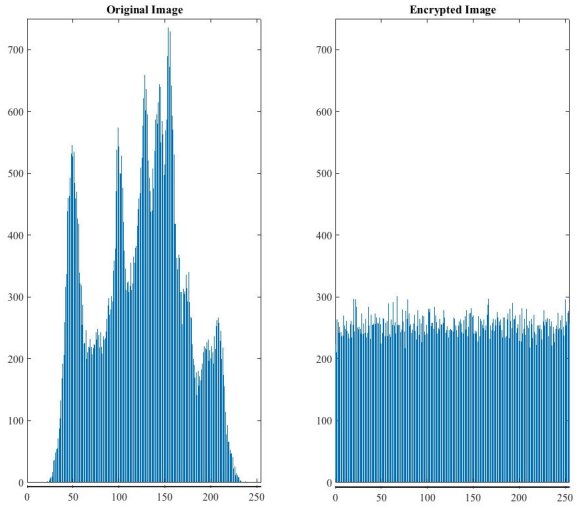

FIGURE 8. Original and encrypted statistical histograms.

Figure 8 in which the message M represents an image (256 x 256 greyscale image) randomly picked from the Internet. Particular, we display the histograms of the original image and the ciphered image (i.e., $M \oplus K$). As expected, the encryption makes the histogram uniform since K follows a uniform distribution (independent and random variable) and C is computed by XORing M and K. That is, an adversary does not stand a chance to extract information from the encrypted image. Formally, we regularly use NPCR (number of pixels change rate), and UACI (uniform average change intensity) tests to evaluate the proper behaviour of image encryption cyphers regarding differential attacks [42]. For computing these two tests, we encrypt two images that in plain-text only differs by one-pixel. The resulting encrypted images, C^1 and C^2 are used to computed both coefficients. Shortly, NPCR represents an average value of the pixels that change, and UACI is an average value of intensity changes (in both cases between C^1 and C^2). We have randomly taken five images from the Internet and calculated NPCR and UACI coefficients for each one as displayed in Table 5. We can conclude that in all the cases, the two tests pass successfully at the 0.01 significance level (taking into consideration the thresholds suggested in [43]).

IV. OUR PROPOSAL IN THE IoY CONTEXT

In the previous sections, we study the suitability of human gait data and that of the proposed algorithm for generating random numbers. In this section, we describe the features of our proposal that make it suitable for the IoY environment.

First of all, the scarcity of resources in the IoY context is one of the critical aspects of the technology. In that sense, our proposal offers a lightweight solution in terms of computational resources. As we stated before, we intend to use the sensors integrated on wearable devices so that these sensors are already present in the system, and no extra-hardware is necessary for them. Likewise, the *GetEntropy* procedure involving the use of the Hadamard transformation is also a

TABLE 5. NPCR and UACI randomness tests. (a) NPCR tests. (b) UACI randomness tests.

(a) NPCR Tests

	NPCR
File-1	99.6139 %
File-2	99.5758 %
File-3	99.6032 %
File-4	99.5895 %
File-5	99.6353 %
Optimal value (256 x 256) [43]	$\mathbf{NPCR_{0.05} \geq 99.5693}$
	$\mathbf{NPCR_{0.01} \geq 99.5527}$
	$\mathbf{NPCR_{0.001} \geq 99.5341}$

(b) UACI Randomness Tests

	UACI
File-1	33.2277 %
File-2	33.5180 %
File-3	33.5165 %
File-4	33.3346 %
File-5	33.6247 %
Optimal value (256 x 256) [43]	$\mathbf{33.2824\% \leq UACI_{0.005} \leq 33.6447\%}$
	$\mathbf{33.2255\% \leq UACI_{0.01} \leq 33.7016\%}$
	$\mathbf{33.1594\% \leq UACI_{0.001} \leq 33.7677\%}$

lightweight solution that consumes very few resources. Note that this transformation has been used widely as an efficient alternative to the Fast-Fourier-Transform in several fields such as communications or spectral analysis [44]. Notably, in the context of IoT, we can find implementations optimised explicitly for these environments. [45]. Concerning the *GetRandomness* procedure, we use a modification of the H-function. As stated in section II.B, bitwise XOR operations between matrices and bitwise circular shift operations over each element of a matrix are the operations computed for its calculation. These two operations are among the most lightweight computations commonly used in lightweight cryptography [13], [46].

Power consumption is also one of the critical features of IoY solutions. To obtain an accurate estimation of the power consumption used by our proposal, we have implemented it in an Artix-7 C7A35T FPGA. This board is a low-cost low-power FPGA. We can program it using the Vivado Tool that integrates a very accurate power consumption estimator. At a nominal temperature of 25^{o}C and a clock frequency of 100 MHz, the algorithm consumes 2,3 μJ.

Finally, another critical aspect of any TRNG is its throughput. The application in which we utilise the TRNG determines how demanding is the bit rate required. Note that our proposal is very efficient since we generate 1 byte (without post-processing) and 2 bits (final output of *GetRandomness* procedure) per each captured sample. Under the settings of HuGaDB in which the same sampling frequency is 60 Hz, it implies that we can generate 120 bits/sec. In Table 6, we provide a throughput comparison between some existing TRNGs based on biometric data and our proposal. Our proposal offers a slightly moderate-high rate. It is two orders of magnitude higher than the bitrate offered by ECG based TRNGs [19], [20]. Even it is more than six times the throughput provided by the novel TRNG proposed by Tuncer and Kaya, and that applies to a wide variety of biological signals including EEG, blood pressure or GSR [24].

TABLE 6. Throughput comparison of RNG based on biometric data.

Reference	Noise Source	Throughput
This Work	Gait	$fs \cdot 2$ bits/sec
[19]	ECG	2 bits/sec
[20]	ECG	14 bits/sec
[24]	Various	$\frac{fs}{3}$ bits/sec

V. CONCLUSION

There is a vast amalgam of applications in which IoT devices are useful or will use shortly. The incorporation (preferably by-design) of security services is mandatory to prune security incidents and to avoid continuous and frequent use of security patches. The random numbers used in security applications must comply with exigent batteries of tests. In the past, even well-know cryptographic protocols like OpenSSL or successful commercial products like the PlayStation 3 put at risk their security due to the usage of weak random-numbers. Motivated by this, we propose a new and robust TRNG based on Gait data. In the past, TRNGs were mainly built on a physical phenomenon (e.g. thermal noise or decay of a nuclear source). Still, recently solutions based on data acquired by sensors (especially when those are in or over our body) have gained fervour. In our case, we use gait data, and more particular data captured from six accelerometers and six gyroscopes (both capturing data in the standard three axes: x, y and z).

Finally, it is worth mentioning the novelty of our proposal. Instead of using a classical approach based on a physical phenomenon of nature, we build our TRNG on sensors that are over our body. It means that we are a fruitful source of entropy while going through our daily activities. For this, we use gyroscope and accelerometer data which are included in a wide variety of IoT devices such as smartwatches or even smart socks for runners. We have tested the generated random number with several batteries of tests. For this extensive analysis, we can conclude that the output bits look like the ones produced by a random variable. We hope this contribution helps to make more secure IoT devices since random numbers are critical in security services and mechanisms.

REFERENCES

[1] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019, doi: 10.1109/ACCESS.2019.2924045.

[2] M. B. Mohamad Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019.

[3] E. Manavalan and K. Jayakrishna, "A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements," *Comput. Ind. Eng.*, vol. 127, pp. 925–953, Jan. 2019.

[4] R. Metz, *Wearable Computers and the Internet of You*. MIT Technology Review, 2014. [Online]. Available: https://www.technologyreview.com/2014/05/20/172797/the-internet-of-you/

[5] D. Estrin and C. W. Thompson, "Internet of you: Data big and small [guest editors' introduction]," *IEEE Internet Comput.*, vol. 19, no. 06, pp. 8–10, Nov. 2015.
[6] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.
[7] A. B. O. Lopez, L. H. Encinas, A. M. Munoz, and F. M. Vitini, "A lightweight pseudorandom number generator for securing the Internet of Things," *IEEE Access*, vol. 5, pp. 27800–27806, 2017.
[8] R. Ijaz and M. A. Pasha, "Area-efficient and high-throughput hardware implementations of TAV-128 hash function for resource-constrained IoT devices," in *Proc. 9th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Systems: Technol. Appl. (IDAACS)*, vol. 2, Sep. 2017, pp. 832–835.
[9] X. Chen, B. Li, Y. Wang, Y. Liu, and H. Yang, "A unified methodology for designing hardware random number generators based on any probability distribution," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 8, pp. 783–787, Aug. 2016.
[10] L. Gong, J. Zhang, H. Liu, L. Sang, and Y. Wang, "True random number generators using electrical noise," *IEEE Access*, vol. 7, pp. 125796–125805, 2019.
[11] M. Huang, A. Wang, P. Li, H. Xu, and Y. Wang, "Real-time 3 Gbit/s true random bit generator based on a super-luminescent diode," *Opt. Commun.*, vol. 325, pp. 165–169, Aug. 2014.
[12] H. Martin, P. Peris-Lopez, J. Tapiador, and E. Millan, "A new TRNG based on coherent sampling with self-timed rings," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 91–100, Feb. 2016.
[13] V. B. Suresh and W. P. Burleson, "Entropy and energy bounds for metastability based TRNG with lightweight post-processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 7, pp. 1785–1793, Jul. 2015.
[14] S. Mathew, S. Hsu, R. Krishnamurthy, D. Johnston, P. Newman, S. Satpathy, V. Suresh, M. Anders, H. Kaul, G. Chen, and A. Agarwal, "RNG: A 300–950 mV 323Gbps/W all-digital full-entropy true random number generator in 14 nm FinFET CMOS," in *Proc. ESSCIRC Conf. 41st Eur. Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2015, pp. 116–119.
[15] P. Z. Wieczorek and K. Golofit, "True random number generator based on flip-flop resolve time instability boosted by random chaotic source," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 4, pp. 1279–1292, Apr. 2018.
[16] E. Farcot, S. Best, R. Edwards, I. Belgacem, X. Xu, and P. Gill, "Chaos in a ring circuit," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 29, no. 4, Apr. 2019, Art. no. 043103.
[17] 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, Art. no. 2747, [Online]. Available: http://www.mdpi.com/1424-8220/18/9/2747, doi: 10.3390/s18092747.
[18] G. Chen, "Are electroencephalogram (EEG) signals pseudo-random number generators?" *J. Comput. Appl. Math.*, vol. 268, pp. 1–4, Oct. 2014.
[19] D. Karaoğlan Altop, A. Levi, and V. Tuzcu, "Deriving cryptographic keys from physiological signals," *Pervas. Mobile Comput.*, vol. 39, pp. 65–79, Aug. 2017.
[20] S. Pirbhulal, H. Zhang, W. Wu, S. C. Mukhopadhyay, and Y.-T. Zhang, "Heartbeats based biometric random binary sequences generation to secure wireless body sensor networks," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 12, pp. 2751–2759, Dec. 2018.
[21] C. Camara, H. Martín, P. Peris-Lopez, and M. Aldalaien, "Design and analysis of a true random number generator based on GSR signals for body sensor networks," *Sensors*, vol. 19, no. 9, p. 2033, 2019.
[22] J. Szczepanski, E. Wajnryb, J. M. Amigó, M. V. Sanchez-Vives, and M. Slater, "Biometric random number generators," *Comput. Secur.*, vol. 23, no. 1, pp. 77–84, Feb. 2004.
[23] D. Nguyen, D. Tran, W. Ma, and K. Nguyen, "EEG-based random number generators," in *Network and System Security*, Z. Yan, R. Molva, W. Mazurczyk, and R. Kantola, Eds. Cham, Switzerland: Springer, 2017, pp. 248–256.
[24] S. Arslan Tuncer and T. Kaya, "True random number generation from bioelectrical and physical signals," *Comput. Math. Methods Med.*, vol. 2018, pp. 1–11, Jul. 2018.
[25] R. Chereshnev and A. Kertész-Farkas, "Hugadb: Human gait database for activity recognition from wearable inertial sensor networks," in *Analysis of Images, Social Networks and Texts*. Springer, 2018, pp. 131–141.
[26] G. Bovi, M. Rabuffetti, P. Mazzoleni, and M. Ferrarin, "A multiple-task gait analysis approach: Kinematic, kinetic and EMG reference data for healthy young and adult subjects," *Gait Posture*, vol. 33, no. 1, pp. 6–13, Jan. 2011.
[27] A. Sant'Anna, A. Salarian, and N. Wickstrom, "A new measure of movement symmetry in early Parkinson's disease patients using symbolic processing of inertial sensor data," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 7, pp. 2127–2135, Jul. 2011.
[28] M. Bachlin, D. Roggen, G. Troster, M. Plotnik, N. Inbar, I. Meidan, T. Herman, M. Brozgol, E. Shaviv, N. Giladi, and J. M. Hausdorff, "Potentials of enhanced context awareness in wearable assistants for Parkinson's disease patients with the freezing of gait syndrome," in *Proc. Int. Symp. Wearable Comput.*, Sep. 2009, pp. 123–130.
[29] M. Giuberti and G. Ferrari, "Simple and robust BSN-based activity classification: Winning the first BSN contest," in *Proc. 4th Int. Symp. Appl. Sci. Biomed. Commun. Technol. (ISABEL)*, 2011, pp. 1–5.
[30] B. J. Fino and R. Algazi, "Unified matrix treatment of the fast Walsh-Hadamard transform," *IEEE Trans. Comput.*, vol. C-25, no. 11, pp. 1142–1146, Nov. 1976.
[31] C. Camara, P. Peris-Lopez, and J. E. Tapiador, "Human identification using compressed ECG signals," *J. Med. Syst.*, vol. 39, no. 11, pp. 148:1–148:10, Nov. 2015.
[32] H. Hosseini-Nejad, A. Jannesari, and A. M. Sodagar, "Data compression in brain-Machine/Computer interfaces based on the Walsh–Hadamard transform," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 1, pp. 129–137, Feb. 2014.
[33] S. Peter, B. Pratap Reddy, F. Momtaz, and T. Givargis, "Design of secure ECG-based biometric authentication in body area sensor networks," *Sensors*, vol. 16, no. 4, p. 570, 2016.
[34] M. Dichtl, "Bad and good ways of post-processing biased physical random numbers," in *Fast Software Encryption*, A. Biryukov, Ed. Berlin, Germany: Springer, 2007, pp. 115–137.
[35] M. S. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, and M. Boyle, *Recommendation for the Entropy Sources Used for Random Bit Generation*, document NIST Special Publication 800-90B, 2018.
[36] S. Zhu, Y. Ma, X. Li, J. Yang, J. Lin, and J. Jing, "On the analysis and improvement of min-entropy estimation on time-varying data," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1696–1708, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8871148
[37] R. G. Brown. (2011). *Dieharder: A Random Number Test Suite V3.31.1*. [Online]. Available: https://webhome.phy.duke.edu/~rgb/General/dieharder.php
[38] B. K. Park, H. Park, Y.-S. Kim, J.-S. Kang, Y. Yeom, C. Ye, S. Moon, and S.-W. Han, "Practical true random number generator using CMOS image sensor dark noise," *IEEE Access*, vol. 7, pp. 91407–91413, 2019.
[39] F. Yu, Q. Wan, J. Jin, L. Li, B. He, L. Liu, S. Qian, Y. Huang, S. Cai, Y. Song, and Q. Tang, "Design and FPGA implementation of a pseudo-random number generator based on a four-wing memristive hyperchaotic system and Bernoulli map," *IEEE Access*, vol. 7, pp. 181884–181898, 2019.
[40] J. Walker. (1998). *Randomness Battery*. [Online]. Available: http://www.fourmilab.ch/random/
[41] D. Hurley-Smith and J. Hernandez-Castro, "Certifiably biased: An in-depth analysis of a common criteria EAL4+ certified TRNG," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 1031–1041, Apr. 2018.
[42] M. Z. Yildiz, O. F. Boyraz, E. Guleryuz, A. Akgul, and I. Hussain, "A novel encryption method for dorsal hand vein images on a microcomputer," *IEEE Access*, vol. 7, pp. 60850–60867, 2019.
[43] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J., Multidisciplinary Jo. Sci. Technol.*, vol. 1, no. 2, pp. 31–38, 2011.
[44] S. S. U. Qadri, C. F. Azim, D. Hazry, S. F. Ahmed, M. K. Joyo, M. H. Taveer, and F. A. Warsi, "Hardware implementation of fast-sequency ordered complex Hadamard transform," in *Proc. IEEE 10th Int. Colloq. Signal Process. Appl.*, Mar. 2014, pp. 106–110.
[45] A. A. Rahimi, H. Hu, K. Sivakumar, and S. Gupta, "Energy-efficient serialized Walsh-Hadamard transform based feature-extraction for information-aware compressive sensing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.
[46] H.-Y. Chien, "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," *IEEE Trans. Dependable Secure Comput.*, vol. 4, no. 4, pp. 337–340, Oct. 2007.

CARMEN CAMARA received the Ph.D. degree in computer science and the Ph.D. in biomedical engineering. She is currently an Assistant Professor with the Computer Security Laboratory, Carlos III University of Madrid, Spain. Her research interests are in the fields of cybersecurity in e-Health, bioengineering, and data science.

HONORIO MARTÍN (Member, IEEE) received the Ph.D. degree in electronics engineering from the Universidad Carlos III de Madrid, Spain, in 2015. He is currently a Postdoctoral Researcher with the Department of Electronic Technology, Universidad Carlos III de Madrid. His current research interests include the study of lightweight cryptography, hardware implementations, radio-frequency identification systems, and low-power design.

PEDRO PERIS-LOPEZ received the M.Sc. degree in telecommunications engineering and the Ph.D. degree in computer science from the Carlos III University of Madrid, Spain, in 2004 and 2008, respectively. He is currently an Associate Professor with the Department of Computer Science, Carlos III University of Madrid. His research interests are in the field of cybersecurity and e-health, digital forensics and hardware security. In these fields, he has published a large number of articles in specialized journals (57) and conference proceedings (44). His works have more than 4250 citations, and his H-index is 29. For additional information see: https://www.lightweightcryptography.com/.

LUIS ENTRENA (Member, IEEE) received the Industrial Engineer degree from the Universidad de Valladolid, Spain, in 1998, and the Ph.D. degree in electronic engineering from the Universidad Politécnica de Madrid, Spain, in 1995. From 1990 to 1993, he was with AT&T Microelectronics, Bell Labs, USA. From 1993 to 1996, he was a Technical Project Leader with TGI, Spain. Since 1996, he has been an Associate Professor with Universidad Carlos III de Madrid, Spain, where he has served as the Head of the Electronic Technology Department and the Director of the Postgraduate Program in electrical engineering. He has coauthored over 150 articles and one patent. His current research interests include on-line testing, fault tolerance, soft error sensitivity evaluation and mitigation, hardware security, and hardware acceleration.