EXTENDED ANALYSIS USING NIST METHODOLOGY OF SENSOR DATA ENTROPY

Florin RĂSTOCEANU[1], Bogdan-Iulian CIUBOTARU[2], Ionuț RĂDOI[3], Constantin Viorel MARIAN[4]

IoT security is an unresolved issue at this time, which is not easy to achieve due to the multitude of types of sensors used. As almost any security solution involves cryptography elements, which by default requires random numbers generators, we aimed to analyze the possibility to extract entropy from different sensors data. The method proposed in this paper minimizes the effects of the diversity of different devices used and optimizes the entropy level according to consumed energy. The proposed solution was validated using the NIST methodology, which involves 10 different estimators and a large dataset.

Keywords: entropy, cryptography, random numbers, IoT security, sensors.

1. Introduction

Internet of Things (IoT) technologies are present in our lives in an increasing percentage. In case of applications aimed at human consumers, Internet connected electronic devices share personal data and confidential information. In case of applications aimed at automating cars, industrial processes or interconnected cities, the Internet connected electronic devices are mission critical resources that share sensitive data. The market has a high growth rate. According to statistics presented in [1], the number of connected IoT devices in 2020 is 30.73 trillion and is estimated to increase by about 2.5 times in the next five years. This rapid growth creates a diversified and heterogeneous market in which security issues are left in the background. The lack of standardization in the field accentuates these problems. The NIST report [2] summarizes the status of international cyber security standards for IoT applications and states that not one of the security areas taking into consideration is fully standardized.

In this context, the identification of security solutions, independent of the type of device used, represents an interest for the entire research community. The paper aims to present a solution, which meets the security requirements for an IoT

[1] Scientific researcher gr. III, Military Equipment and Technology Research Agency & PhD student, University POLITEHNICA of Bucharest, Romania, e-mail frastoceanu@acttm.ro

[2] Assistant researcher, Military Equipment and Technology Research Agency & PhD student, University POLITEHNICA of Bucharest, Romania, e-mail bciubotaru@acttm.ro

[3] Scientific researcher gr. III, Military Equipment and Technology Research Agency & PhD student, University POLITEHNICA of Bucharest, Romania, e-mail iradoi@acttm.ro

[4] Associate Professor, Department of Engineering in Foreign Languages, University POLITEHNICA of Bucharest, Romania, e-mail: constantinvmarian@gmail.com

system, in order to ensure the CIA tirade: confidentiality, authentication and integrity for any data. This is achieved using cryptographic protocols, methods and algorithms. Most of cryptographic techniques use randomly generated numbers, used in different forms: cryptographic keys, initialization vectors, challenge response nonce, random seeds, passwords, etc. Modern cryptography uses number random bit generators (RBG) which can be of two types: deterministic random bit generator (DRBG) or a non-deterministic random bit generator (NRBG). Both types use entropy sources that are based on noise sources.

The solution proposed in this paper is a suitable noise source that can be used for IoT applications that are characterized by limited resources. It is well known that most IoT solutions do not have sufficient resources to ensure security through cryptographic functions. Using data already collected from sensors to be used as a noise source can be a successful approach. To demonstrate this we aimed to identify the types of sensors suitable for the construction of noise sources. In our experiments, we used a number of six different types of sensors: temperature, humidity, air pressure, accelerometer, gyroscope and magnetometer. In order to highlight the applicability of the proposed solution, the data were acquired from sensors during their use in specific and common applications. Thus, the temperature, humidity and air pressure sensors were evaluated in parameters monitoring applications inside the buildings but also in the open air, and the motion sensors (the accelerometer, the magnetometer and the gyroscope) in a movement case (data collected from a moving car), but also in the case of no movement.

The remainder of this paper is structured as follows. Section 2 presents theoretical background that includes solutions for entropy extraction and NIST methodology used to validate an entropy source. Section 3 presents the experiments performed and results interpretation and in section the conclusions are summarized.

2. Theoretical background

There are currently different methods to extract entropy. Some of them are based on random events such as inter-keyboard timings, inter-interrupt timings from some interrupts. These are specific to environments running operating systems such as Linux [3] or Windows [4]. These solutions are often not suitable for IoT devices due to the resources required by operating systems. On the other hand, there are reports of various vulnerabilities, most of which come from improper operation [5][6]. Other methods of entropy extraction are based on the randomness of physical phenomena [7][8]. Although they generate enough entropy for cryptographic applications, these solutions require additional resources or even additional hardware. The solution proposed in this paper has the

advantage of using what already exists on IoT devices that collect data from sensors.

The idea of extracting entropy from data collected from sensors has been discontinued by researchers. There are several published papers in which sensors of various types have been used to extract entropy. In most cases, the conclusions were that the data extracted from the sensors contained sufficient entropy, but there were also conclusions that contradicted this. These contradictions arise for several reasons. One is that different entropy estimators have been used. Another would be that the data are extracted under different conditions and often in insufficient quantity for a statistical analysis.

In [9] the authors extracted data from on-board of temperature, humidity and light sensors. Min-entropy and Shannon entropy were used as entropy estimators. To extract the data, they used four methods in which they selected the least significant bits, starting from the idea that they can provide the higher amount of entropy. The conclusion was that the most entropy could be harvested using the method of extracting only the least-significant bit of the sensors data. J Voris, N. Saxena, T. Halevi in [10] analyzed the entropy generated by the accelerometer in different types of movement: stationary, hand, drop, triangle, key twist, circle, alpha and arc swipe. They used min-entropy estimator in all their experiments. For experiments that involved movement they used accelerometer samples collected within 10 minutes. In the stationary case alone, they used a number of samples over 1000000 to estimate entropy. The conclusion of this paper was that accelerometers are suitable for generating entropy and, more than that, are resistant to a variety of environmental variations and even to adversarial manipulation.

On the other hand, the conclusions from [11] contradict these results. In this paper, the authors evaluated several types of sensors such as vibration, magnetic, temperature, humidity, motion, gas pressure in three modes of operation: stability, saturation and dynamic mode. Min-entropy, Shannon entropy and four other estimators were used to calculate the entropy: frequency tests, collision test, compression test and Markov test. A first important conclusion is that the values obtained with these estimators are very different. For example, the value of the entropy calculated with the Shannon formula can be five times higher than that of the estimator calculated after the compression test. Another conclusion is that only vibration, motion and magnetic sensors can provide sufficient entropy. As the authors admit, because only for the stability mode they managed to collect enough samples, the confidence level for this mode is higher than in the saturation and dynamic mode.

From the studies performed so far, the following question cannot be answered: can the data extracted from the sensors be used for the construction of entropy sources? In this paper, for the first time, a NIST-accredited methodology

was used, which includes a number of 10 estimators and datasets of at least 1000000 samples. The use of a large and diverse number of estimators as well as sufficiently large datasets minimizes the probability that entropy values to be greatly overestimated. Since the goal is for this source of entropies to be used by IoT devices with limited resources, we performed tests to identify the maximum level of entropy that can be extracted from a single sample by making a bitmap based on the entropy generated by the bits on each sample position.

Recently, NIST released final version of the recommendations for the construction of entropy sources used in random bit generators [12]. This publication specifies the design principles and requirements for the entropy sources and the tests for the validation of entropy sources. An entropy source contains several components: a noise source, health tests and an optional conditional component. The root element of an entropy source is the noise source. This component must contain a non-deterministic process that provides the uncertainty associated with the output values of the entropy source. Therefore, the first step in analyzing an entropy source is to estimate the value of the raw data entropy that represents the output from the noise source. Depending on the analyzed characteristics, the NIST methodology approaches the entropy estimation process differently. Thus, the analyzed data can be independently and identically distributed (IID) or non-IID. In both cases, the tests are performed on a dataset of 1000000 samples extracted directly from the noise source, named raw data. This dataset can be composed of several subsets of length at least 1000 samples. Min-entropy is used to estimate the entropy value in these recommendations. Min-entropy of a random variable X (in bits) is the largest value m (in bits) that has the property that each observation of X provides at least m bits of information. In [12] and in our experiments the min-entropy of an independent discrete random variable X that takes values from the set A={x1,x2,…,xk} with probability Pr(X=xi) = pi for i =1,…,k is defined as:

$$H = \min_{1<i<k}(-\log_2 p_i) = -\log_2 \max_{1<i<k} p_i \quad (1)$$

The samples extracted from a noise source are considered IID if each sample has the same probability distribution as every other sample, and all samples are mutually independent. This is tested by applying statistical tests that have the role of highlighting the fact that samples are not IID. If no evidence is identified that these samples are non-IID, then it is assumed that the samples are IID. The statistical tests used for IID data are of two types: permutation tests and chi-square tests. Permutation testing is a method of testing a statistical hypothesis in which the result of the statistical test applied to a permutation of the original data is compared with the result of the statistical test applied to the original data and not with a standard statistical distribution. In this case, 10000 permutations of the dataset are generated. Permutations are obtained using Fisher-Yates shuffle

algorithm. The individual statistical tests used to verify the IID characteristic of the data are the following [12]: Excursion Test Statistic, Number of Directional Runs Test, Length of Directional Runs Test, Number of Increases and Decreases Test, Number of Runs Based on the Median Test, Average Collision Test Statistic, Maximum Collision Test Statistic, Periodicity Test Statistic, Covariance Test Statistic, Compression Test Statistic. The choice of this testing process started from the premise that the values obtained by the statistical tests applied to permutations of the original data do not differ much in the case of IID data. After attesting that the source is IDD, the entropy of the min-entropy estimation is determined using the most common estimated value. Chi-square statistical procedures includes the following tests [12]: Independence for Non-Binary Data, Goodness-of-fit for Non-Binary Data, Independence for Binary Data, Goodness-of-fit for Binary Data, Length of the Longest Repeated Substring Test.

Because the vast majority of entropy sources are not IID and to exploit these sources as well, NIST has developed a more complex methodology for testing these sources using a diverse and conservative set of entropy tests. Thus, a number of ten different estimators are used for which the min-entropy value is calculated. The minimum of all the estimates is taken as the entropy assessment. The estimators used are as follows [12]:

Most Common Value Estimate finds the proportion of the most common value in the input dataset and constructs a confidence interval for this proportion. The upper bound of the confidence interval is used to estimate the min-entropy per sample of the source.

Collision Estimate measures the mean number of samples to the first collision in a dataset, where a collision is any repeated value. The goal of the method is to estimate the probability of the most-likely output value, based on the collision times. This entropy estimation method is only applied to binary inputs.

Markov Estimate provides a min-entropy estimate by measuring the dependencies between consecutive values from the input dataset. The min-entropy estimate is based on the entropy present in any subsequence (i.e., chain) of outputs, instead of an estimate of the min-entropy per output.

Compression Estimate computes the entropy rate of a dataset, based on how much the dataset can be compressed. This estimator is based on the Maurer Universal Statistic [13].

t-Tuple Estimate examines the frequency of t-tuples (pairs, triples, etc.) that appears in the input dataset and produces an estimate of the entropy per sample, based on the frequency of those t-tuples.

Longest Repeated Substring Estimate computes the collision entropy (sampling without replacement) of the source, based on the number of repeated substrings (tuples) within the input dataset.

Multi Most Common in Window Prediction Estimate contains several subpredictors, each of which aims to guess the next output, based on the last w outputs.

Lag Prediction Estimate contains several subpredictors, each of which predicts the next output, based on a specified lag. The lag predictor keeps a scoreboard that records the number of times that each subpredictor was correct and uses the subpredictor with the most correct predictions to predict the next value.

MultiMMC Prediction Estimate is composed of multiple Markov Model with Counting (MMC) subpredictors. Each MMC predictor records the observed frequencies for transitions from one output to a subsequent output (rather than the probability of a transition, as in a typical Markov model), and makes a prediction, based on the most frequently observed transition from the current output.

LZ78Y Prediction Estimate is based on LZ78 encoding with Bernstein's Yabba scheme [14] for adding strings to the dictionary.

3. Experiments and results

The purpose of the experiments was to validate a noise source based on unpredictable phenomena captured from different types of sensors. For this purpose, the NIST methodology described in chapter 3 was used. As the data collected from the sensors are not IID, the corresponding was approached.

3.1. Platform description

For all the experiments performed, on-board sensors were used on the B-L475E-IOT01A1 board developed by STMicroelectronics. This board is specially developed for use in IoT nodes and is equipped with an ultra-low-power STM32L4 Series MCUs based on ARM Cortex-M4 core.
The board is equipped with the following sensors:

- HTS221 a capacitive digital sensor for relative humidity and temperature;
- LIS3MDL a high-performance 3-axis magnetometer;
- LSM6DSL a 3D accelerometer and 3D gyroscope;
- LPS22HB an absolute digital output barometer that includes temperature and air pressure sensors.

3.2. Experiments description

The experiments aimed to identify the possibility of using sensors evaluated for entropy extraction in common use cases. Thus, the analyzed sensors were divided into two categories: sensors that perceive movement (accelerometer, gyroscope and magnetometer) and sensors that perceive environment properties (temperature, humidity and pressure). For each category, we analyzed two use-

cases. For the first category, the sensors were placed in an indoor environment thus simulating air quality monitoring applications in homes or offices and in an outdoor environment simulating air quality monitoring application in smart cities or in agricultural applications. Accelerometer, gyroscope and magnetometer sensors were analyzed in no-moving case and in car-driving use case.

The entropy analysis was performed in several stages. In the first stage, 1000000 samples were extracted for each experiment, representing the least significant 8 bits of the values of the data acquired from the sensors. The values of 8 bits were chosen in accordance to NIST entropy evaluation methodology [12], where is specified that for sizes larger than 8 the results are not guaranteed.

In the second stage, for the cases where reasonable values of entropy were obtained, bitstrings composed of bits from a single position were analyzed. In addition, in the last stage, the entropy of the samples containing only bits of the positions for which good values of the entropy were obtained was re-evaluated.

In all experiments performed for data extraction, the B-L475E-IOT01A1 board sensors was used. The data were collected by a laptop on the serial port emulated by the Teraterm application. The recommended NIST SP800-90B_EntropyAssessment C ++ package application, downloaded from the Github web site [15], was used to evaluate the entropy.

- ***Experiment 1***

- Purpose: evaluation of the entropy of motion sensors in two use-cases: no-moving and car-driving.
- Sensors used:
    - LSM6DSL accelerometer using the following parameters: measurement range: ±2G for no-moving use case and ±4G for car-driving use case and ODR: 416 Hz for both uses-cases;
    - LSM6DSL gyroscope using the following parameters: measurement range: 245mdps for no-moving use case and 1000 mdps for car-driving use case and ODR: 416 Hz for both use-cases;
    - LIS3MDL magnetometer: measurement range: ±4 Gauss for no-moving use case and ±8 Gauss for car-driving use case and ODR: 80 Hz for both use- cases;
- Data acquisition method: 1000000 samples composed of the least significant 8 bits extracted from sensors data. For the tests in the no-moving use case, the board was placed on the table without moving at all during the experiment. For car-driving tests, the data were acquired while the car was in motion. In order not to influence the results, the car was

driven on roads outside the city, thus excluding as much as possible the waiting times at traffic lights or pedestrian crossings;
- Results: Table I shows the entropy values per bit (min-entropy) for the three sensors for each axis (x-axis, y-axis and z-axis) in the two use-cases.

Table 1

Accelerometer, gyroscope and magnetometer entropy values per bit for no-moving and /car driving use cases

Sensor type	No moving case			Car driving case		
	x-axis	y-axis	z-axis	x-axis	y-axis	z-axis
Accelerometer	0.46	0.41	0.30	0.82	0.82	0.82
Gyroscope	0.25	0.36	0.26	0.53	0.54	0.39
Magnetometer	0.47	0.46	0.62	0.56	0.48	0.59

- ***Experiment 2***

- Purpose: evaluation of the entropy of air properties sensors in two use-cases: indoor and outdoor.
- Sensors used: HTS221 temperature extracted using 1Hz ODR and 16 bits resolution, HTS221 humidity extracted using 1Hz ODR and 16 bits resolution, LPS22HB air pressure extracted using 25 Hz ODR and 24 bits resolution, LPS22HB temperature extracted using 25Hz ODR and 16 bits resolution:
- Data acquisition method: 1000000 samples composed of the least significant 8 bits extracted from sensors data;
- Results: Table II shows the entropy values (min-entropy) for the four sensors in the two use-cases.

Table 2

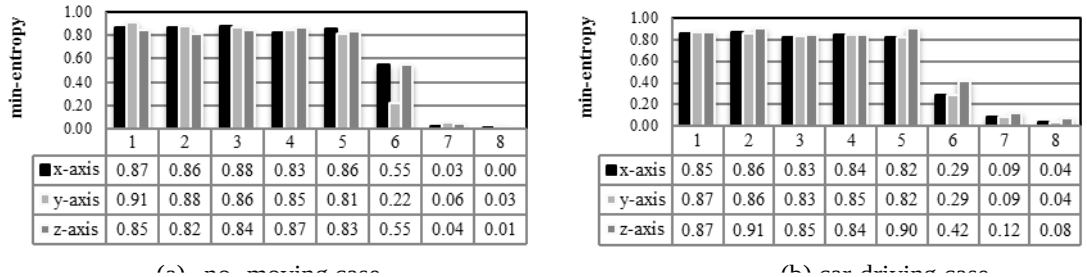
Temperature, humidity and air pressure entropy values per bit for indoor/outdoor use cases

Use case	HTS221		LPS22HB	
	Humidity	Temperature	Air pressure	Temperature
Indoor	0.0321	0.0041	0.3067	0.0004
Outdoor	0.0405	0.0078	0.3028	0.0019

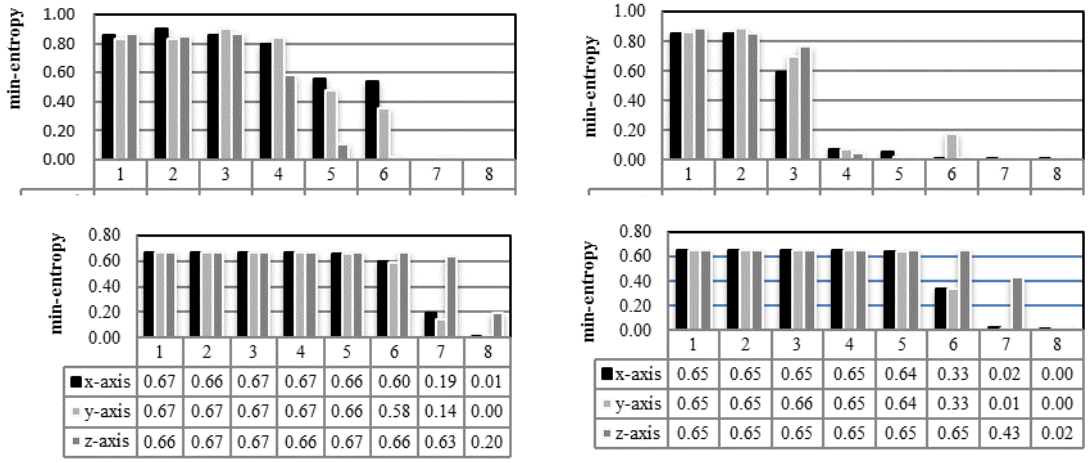
- ***Experiment 3***

- Purpose: identifying the bit positions that contribute the most to the entropy of the extracted data;
- Sensors used: LSM6DSL accelerometer, LSM6DSL gyroscope, LIS3MDL magnetometer and LPS22HB air pressure
- Data acquisition method: 8 bitstrings of 1000000 bits extracted from the same position from the data accumulated from the sensors (the same data from experiment 1 were used);

- Results: Figs. 1-3- show the entropy values for accelerometer, gyroscope, magnetometer and air pressure sensors for each axis in the two use-case and Fig. 4 shows the entropy values (min-entropy) for air pressure in the two use-case.

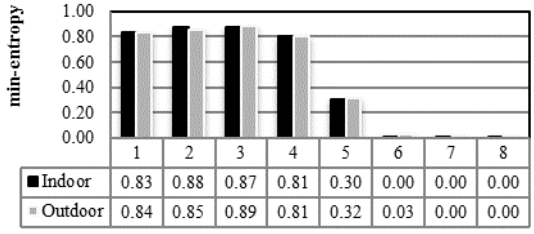


(a) no- moving case (b) car-driving case



(a) no- moving case (b) car-driving case

Fig. 3. Magnetometer entropy values for each sample bit



- ***Experiment 4***

- Purpose: identifying the sample size according to the bit entropy value per sample;

- Sensors used: LSM6DSL accelerometer, LSM6DSL gyroscope, LIS3MDL magnetometer and LPS22HB air pressure
- Data acquisition method: 1000000 samples composed of the bits that generate entropy (taking into account the results obtained in experiment 3, we analyzed samples with different sizes);
- Results: Table III shows the entropy values (bits per sample) for the four sensors in each use-case.

3.3. Results analysis

Not all types of sensors can generate entropy under any conditions. From the results of experiment 2 (Tables I and II) it can be deduced that the values of temperature and humidity cannot change fast enough to generate entropy. Also in the results of the same experiment, it is observed that the air pressure changes enough to ensure an acceptable level of entropy. On the other hand, it must be considered that this sensor has a much higher sensitivity of 4094 LSB/hPa than temperature sensors with a sensitivity of 64 LSB/°C or the humidity sensor with a sensitivity of 256 LSB/% rH. In the case of sensors that monitor the environment properties, no large differences in entropy values were observed for the two use cases. In the case of motion sensors, the problem is different. According to the results of Experiment 1, they can produce entropy even if they are not moved and theoretically, there should be no changes in the displayed values. This is due to the sensitivity of the sensors, which is large enough to detect very small values of the measured phenomena. Thus, these sensors can detect vibrations of very small amplitude or minor changes in the magnetic field. In the case of car driving due to the applied motion, the entropy values increase significantly for accelerometer and gyroscope. Because magnetometer is not influenced very much by this type of movement, its values are at the same level.

From the results of experiment 3 (Fig. 1÷4) it can be seen that not all bits produce entropy. In all the analyzed cases, less significant bits produce more entropy, which validates from an experimental point of view the theoretical analysis. The number of bits that produce entropy is different from sensor to sensor and is different even for different axes of the same sensor. From what can be observed the bits that produce entropy do not depend much on the type of motion applied to the sensors. Considering this aspect, in experiment 4 we tried to identify how many least significant bits should select from the sample to obtain a maximum value of entropy per sample. In Table III the maximum values obtained for entropy are bolded. The highest values are obtained by the accelerometer in the case of car-driving (approximately 6 bits of entropy per sample), but not much lower values are obtained by the other sensors.

Table 3

Entropy values in bits/sample for different sample sizes

	Accelerometer					
	No moving case			Car driving case		
	x-axis	y-axis	z-axis	x-axis	y-axis	z-axis
LSB4	3.4452	3.418	3.3514	3.2722	3.2722	3.4289
LSB5	**4.5005**	**4.1806**	**4.089**	4.3818	4.3818	4.2337
LSB6	3.0933	2.6726	3.3552	4.8049	4.8049	4.9146
LSB8	3.7377	3.2904	3.6296	**6.6008**	**6.6008**	**6.6344**
	Gyroscope					
	No moving case			Car driving case		
	z-axis	x-axis	z-axis	x-axis	z-axis	x-axis
LSB3	2.5392	2.485	2.5301	2.5249	2.4407	2.4157
LSB4	**3.1611**	3.2561	2.8752	3.0367	2.9862	2.7979
LSB5	2.8757	**3.5868**	**2.9282**	3.8585	3.5642	**3.4328**
LSB8	2.063	2.8812	2.1167	**4.296**	**4.3472**	3.1736
	Magnetometer					
	No moving case			Car driving case		
	z-axis	x-axis	z-axis	x-axis	z-axis	x-axis
LSB4	-	-	-	3.1551	3.1587	3.1451
LSB5	4.1465	4.0954	4.0954	4.0125	**4.0732**	4.1665
LSB6	2.7094	2.6811	2.8065	2.5144	2.3736	2.7029
LSB7	**4.9631**	**5.1138**	**5.4314**	-	-	-
LSB8	3.8132	3.7446	4.9727	**4.512**	3.8448	**4.776**
	Air pressure					
	Indoor			Outdoor		
LSB3	2.4921			2.4921		
LSB4	**3.2984**			**3.2984**		
LSB5	2.5577			2.5577		
LSB8	2.4225			2.4225		

4. Conclusions and future work

In this paper, we analyzed the possibility of extracting entropy from data sensors. For this purpose, we analyzed a number of six sensors used in common use cases. For the evaluation of the entropy, we used the min-entropy estimator according to the NIST methodology [12], which uses a number of 10 different estimators and datasets with minimum 1000000 samples. The lowest value of the entropy values obtained by the estimators was taken into account. From the obtained results, it can be concluded that all motion sensors (accelerometer, magnetometer and gyroscope) and air pressure sensor could be used for entropy extraction. In addition, from the performed experiments, it can be observed that higher values of the entropy per extracted sample can be obtained if only the bits that generate the highest entropy are used.

Motion sensors generate entropy even when no motion is applied to them. This may be due to their sensitivity, the intrinsic noise of the sensor and the components with which the data is collected. These aspects were not analyzed in this paper and will be addressed in future work. Other aspects that were not also analyzed in this paper and that can be addressed in future work are related to the stability of the entropy values generated by the sensors. In conclusion, using for the first time the new NIST-approved methodology, we demonstrated in this paper that it is possible to use data generated by certain sensors as a source of entropy.

R E F E R E N C E S

[1] Statista - size of the IoT market worldwide - source: https://www.statista.com/statistics/764051/iot-market-sizeworldwide/, accessed on 10.09.2020

[2] NISTIR 8200 - Interagency Report on the Status of International Cybersecurity Standardization for the Internet of Things (IoT), November 2018, https://doi.org/10.6028/NIST.IR.8200

[3] *François Goichon, Cédric Lauradoux, Guillaume Salagnac, Thibaut Vuillemin* Entropy transfers in the Linux Random Number Generator. [Research Report] RR-8060, INRIA. 2012, pp.26. ffhal00738638f

[4] *Niels Ferguson*, The Windows 10 random number generation infrastructure, October 2019

[5] *David Kaplan, Sagi Kedmi, Avi Dayan, Roee Hay,* Attacking the Linux PRNG on android: weaknesses in seeding of entropic pools and low boot-time entropy, WOOT'14: Proceedings of the 8th USENIX conference on Offensive TechnologiesAugust 2014 Pages 14\

[6] *Yevgeniy Dodis , David Pointcheval , Sylvain Ruhault , Damien Vergnaud , Daniel Wichs*, Security Analysis of Pseudo-Random Number Generators with Input: /dev/random is not Robust,Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, November 2013

[7] *Ben Lampert, Riad S. Wahby, Shane Leonard, Philip Levis*, Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems, November 2016

[8] *Lakshmi Sreekumara, Dr.Ramesh P*, Selection of an Optimum Entropy Source Design for a True Random Number Generator, Procedia Technology 25 (2016) 598 – 605

[9] *M. P. Pawlowski, A. Jara, M. Ogorzalek*, Harvesting Entropy for Random Number Generation for Internet of Things Constrained Devices Using On-Board Sensors, Sensors 2015, 15, 26838-26865

[10] *J. Voris, N. Saxena, T. Halevi*, Accelerometers and Randomness: Perfect Together, Proceedings of the fourth ACM conference on Wireless network security June 2011 Pages 115–126

[11] *C. Hennebert, H. Hossayni, C. Lauradoux,* Entropy harvesting from physical sensors, Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC'13),Budapest, Hungary, 17–19 April 2013; ACM: New York, NY, USA, 2013; pp. 149–154.

[12] *M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, M. Boyle*, NIST Special Publication 800-90B - Recommendation for the Entropy Sources Used for Random Bit Generation, January 2018

[13] *U. Maurer,* A Universal Statistical Test for Random Bit Generators, Journal of Cryptology, Vol. 5, No. 2, 1992, pp. 89-105

[14] *D. Salomon*, Data Compression: The Complete Reference, Chapter 3, Springer, 2007

[15] SP800-90B_EntropyAssessment C++ package https://github.com/usnistgov/SP800-90B_EntropyAssessment, accessed on 10.09.2020