Received 30 May 2022, accepted 15 June 2022. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.3185057

A High-Throughput Random Binary Sequence Generator Based on ECG

CHRISTINE ZENIEH, MOHAMED MAZEN AL-MAHAIRI, AND MOUFID HADDAD
Computer and Automation Engineering Department, Faculty of Mechanical and Electrical Engineering, Damascus University, Damascus, Syria
Corresponding author: Christine Zenieh (eng.c.zenieh@hotmail.com)

ABSTRACT A Wireless Body Area Network (WBAN) is a network that expands over the human body, consisting of multiple nodes that are connected through wireless channels. It offers many applications in the area of remote health care. Maintaining the security of health information in WBAN is an essential requirement. One aspect of ensuring WBAN security is the generation of random binary sequences (RBSs), e.g., encryption keys generation. Due to the very limited resources of WBAN sensors, traditional pseudorandom number generators cannot be used. To reduce resource consumption, some researchers suggested using biometrics in generating RBSs, specifically the electrocardiogram (ECG) signal. However, their methods suffer from low throughput, so they are not suitable for real-time healthcare applications. In this paper, we present a new random sequence generator based on the ECG signal. Our contribution is to build a random sequence generator that generates different length RBSs and has throughput tens or hundreds of times higher than previous methods. Our generator reduces resource consumption due to its very simple processing operations. To evaluate the proposed generator, RBSs of different lengths (128, 256, 512, 1024, 2048 bits) were generated from two ECG datasets, the first is for healthy people, and the second is for people who suffer from arrhythmia. The randomness and distinctiveness of the generated RBSs are evaluated using the National Institute of Standards and Technology (NIST) statistical tests and the Hamming distance. Thus, we have proved that the resulting RBSs are appropriate for information security applications.

INDEX TERMS Electrocardiogram, pseudorandom number generator, random bit sequence, random number generator, wireless body area network.

I. INTRODUCTION

Wireless Body Area Network (WBAN) is a network that provides a mechanism for collecting patient health data using sensors [1]–[3]. It consists of various sensors that can be placed on, around, or in the human body to monitor various biometrics such as body temperature, blood pressure, pulse oximetry, electrocardiogram (ECG), etc. [3]–[5]. WBANs can be applied in multiple medical applications. For example, in healthcare systems, the data collected by WBAN's sensors are used to alert medical personnel when a life-threatening event occurs [4], [6]. Securing the communication between WBAN sensors is essential for preserving the privacy of health data and for ensuring the safety of healthcare delivery [1]–[8]. The tampered biometric data could cause serious medical accidents and even threaten the patient's life [8].

In cryptographic applications, the need for random numbers arises, e.g., common cryptosystems employ keys that must be randomly generated. Many cryptographic protocols require random input at various points [9], e.g., in [10] we developed a secret key exchange scheme for WBAN that uses random numbers to hide secret exchanged values. Two basic types of generators are used to generate random sequences: random number generators (RNG) and pseudorandom number generators (PRNG). RNG uses an entropy source, along with processing functions, to produce randomness. Those functions are needed to overcome any weakness in the entropy source. The entropy source typically consists of some physical quantity. Using RNGs, the production of high-quality random numbers may be too time-consuming. To produce a large number of random numbers, PRNG may be preferable [9]. PRNG uses one or more inputs, called seeds. These seeds themselves must be random and unpredictable. Hence, they should be obtained from the output of an RNG [9]. In general, wireless sensor networks (WSNs) use PRNGs for generating random binary sequences (RBSs) [11], [12]. However, PRNGs require heavy computations to obtain randomness, and their seeds must be carefully chosen

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Ali.

and protected, which in turn leads to the consumption of resources [11].

In this context, one can take advantage of the fact that WBAN sensors record biometric signals, thus randomness can be extracted from them. Some biometric-based RNGs have been developed to avoid the use of PRNGs in WBAN's sensors to save energy and processing capacity [11], [13]–[16]. Some researchers have studied this topic in the context of neuronal signals [17], [18]. The main limitation of these studies is the length of the recordings used and the fact that EEG sensors have limited portability capabilities [13]. Some other researchers have focused on cardiac signals, specifically the ECG signal. There are several characteristic points in the ECG signal that can be used to generate random values [16], [19].

In this paper, we review the most important previous work on ECG-based RNG in the second section. In the third section, we provide a detailed explanation of the proposed scheme. The fourth section presents the implementation results of the proposed generator. Finally, we conclude our work in the fifth section.

II. RELATED WORKS

For securing WBAN, the time interval between two consecutive heartbeats, commonly referred to as the Inter-Pulse Interval (IPI), has gained the attention of many researchers [11], [13], [14], [20]. Each IPI value differs from the others with little deviation. Therefore, some existing approaches extract only the last four entropic bits from each IPI value to produce 128-bit RBSs [14]. The main downside of these approaches is that they are substantially time-consuming. Medical sensors have to acquire an ECG signal for approximately 25-30 seconds to generate 128-bit RBSs. Each 128-bit RBS is generated from 32 IPIs that are obtained from at least 33 successive heartbeats. For an adult, the normal heart rate is 60-100 beats per minute (bpm) [11].

Zheng *et al.,* [16] presented a method that can extract 16 random bits from each heartbeat; they used the periods RR, RQ, RS, RP, and RT to generate RBS. The limitation of their method is the long encoding time required to convert the five different heartbeat intervals into RBSs.

To improve time efficiency, Pirbhulal *et al.,* [11] concatenate only eight consecutive IPIs to produce 128-bit RBSs. A cyclic block encoding technique is applied for decreasing the measurement errors and generating random binary sequences from heartbeats. This technique can be up to four times faster than other IPI-based methods but still suffers from poor time efficiency.

From the above, it can be seen that generating RBSs based on IPI values provides very low performance. In addition to the low performance, according to some recent studies, the time interval between two heartbeats can be determined using a camera and skin color analysis, making these methods less secure [21]. Consequently, Camara *et al.,* [13] did not use IPI values in constructing their RNG, but they use the entire ECG signal. First, the ECG record should be cleaned using filters. Then it is divided into windows that contain an R-peak (one heartbeat). Secondly, the approximation coefficients of each ECG window are obtained by wavelet analysis. The signal is then subsampled by 2, and the process is repeated to increase the level of decomposition. This method is better than its previous in terms of throughput (number of random bits generated per second), but it needs to perform wavelet analysis that may lead to consuming the sensor's power.

To improve time efficiency and power consumption we develop in [22] an ECG-based random binary sequence generator that generates RBSs of 128 bits. In this paper, we improve the proposed generator to generate RBSs of 128, 256, 512, 1024, and 2048 bits. We chose those lengths because they are appropriate for encryption keys and other random values used to secure communication between WBAN sensors. The proposed generator uses the ECG samples' values instead of the IPI values, to profit from the rich entropy that they have. Our generator depends on very simple operations which in turn reduce the energy consumption. It has a very high throughput that outperforms previous works.

III. PROPOSED SCHEME

In the proposed random binary sequence generator scheme, we assume that any entity that is not in contact with the patient's body cannot measure its ECG signal. We rely on the fact that the values of ECG samples when they are selected nonconsecutively, they have better randomness than consecutive ones. To generate random sequences, simple arithmetic operations including addition, subtraction, multiplication, and modulus are applied to some nonconsecutively chosen ECG samples. Those samples are chosen based on the value of another selected sample.

To generate 32-bit random sequences (*RS32*), the ECG signal is acquired and sampled for a specific duration with

TABLE 1. Notations and their descriptions.

Notation	Description
ecg_i	An ECG sample from the acquired ECG signal with index i.
len	The number of the acquired ECG samples.
X100, *X10*, *X1*	The tenths, hundredths, and thousandths of ecg_i respectively, given that ecg_i values range from -1 to 1.
a, b, c, d, e	Secret values used in calculating f_1, f_2, f_3, f_4, and f_5, with a minimum length of 10 bits for each.
f_1, f_2, f_3, f_4, f_5	Five values used for selecting the five ECG samples from which RS32 will be generated.
RS23	A 32-bit random sequence generated from the proposed generator.
RS32_SET	The set of the generated *RS32*.
n	The number of the generated *RS32* (the number of *RS32_SET* elements).
RSL	An L bit random sequence formed by the concatenation of *RS32*.
L	The length of the random sequence *RS* in bits. It must be equal to one of the values: 128, 256, 512, 1024, 2048.
l	L/32.
RS128, *RS256*, *RS512*, *RS1024*, *RS2048*	A 128-bit, 256-bit, 512-bit, 1024-bit, and 2048-bit random sequence respectively, that is formed by the concatenation of 4, 8, 16, 32, and 64 *RS32* respectively.

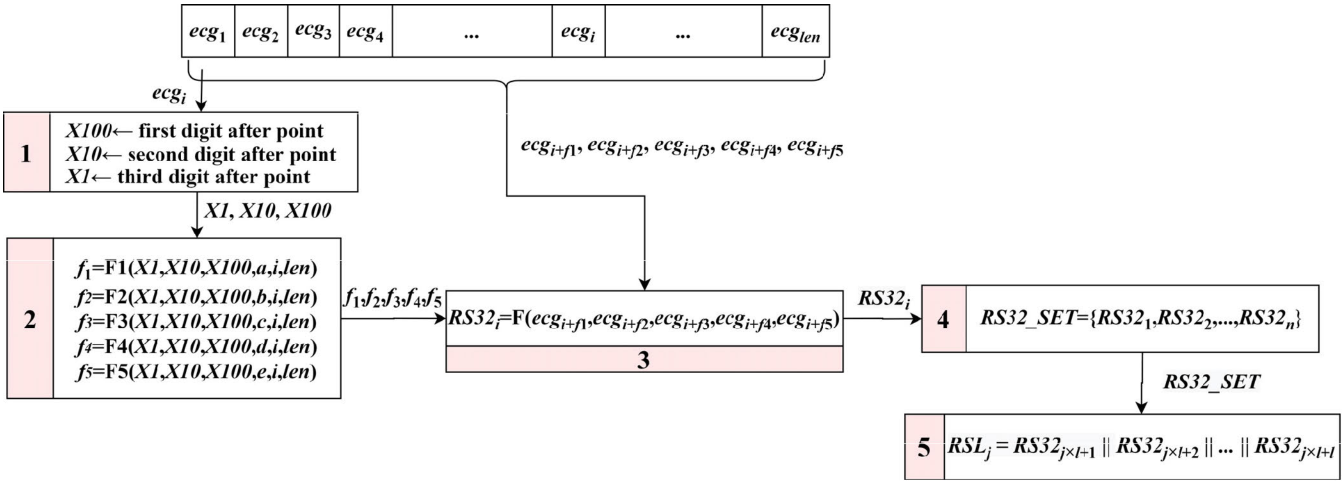
ecg_1 ecg_2 ecg_3 ecg_4 ... ecg_i ... ecg_{len}
ecg_i
1
X100← first digit after point
X10← second digit after point
X1← third digit after point
ecg_{i+f1}, ecg_{i+f2}, ecg_{i+f3}, ecg_{i+f4}, ecg_{i+f5}
X1, X10, X100
2
f_1=F1(X1,X10,X100,a,i,len)
f_2=F2(X1,X10,X100,b,i,len)
f_3=F3(X1,X10,X100,c,i,len)
f_4=F4(X1,X10,X100,d,i,len)
f_5=F5(X1,X10,X100,e,i,len)
f_1,f_2,f_3,f_4,f_5
$RS32_i$=F(ecg_{i+f1},ecg_{i+f2},ecg_{i+f3},ecg_{i+f4},ecg_{i+f5})
3
$RS32_i$
4
RS32_SET={$RS32_1$,$RS32_2$,...,$RS32_n$}
RS32_SET
5
$RSL_j = RS32_{j\times l+1}$ || $RS32_{j\times l+2}$ || ... || $RS32_{j\times l+l}$

FIGURE 1. The proposed ECG-based random binary sequence generator scheme where ecg_i is an ECG sample with the index i, len is the number of the acquired ECG samples, X100, X10,and X1 are digits equal to the tenths, hundredths, and thousandths of ecg_i respectively given that ecg_i values range from −1 to 1, a, b, c, d, e are five secret values used to calculate f_1, f_2, f_3, f_4, f_5, RS32 is a 32-bit random sequence generated from the proposed generator, RS32_SET is the set of the generated $RS32_i$, n is the number of generated RS32, RSL is an L-bit random sequence formed by the concatenation of l string RS32.

a minimum sampling frequency of 128 Hz, and a minimum analog to digital convergence resolution of 11 bits. For each *RS32*, one sample is chosen, and some calculations are applied to it to generate five values. The resulting five values are then used to calculate five samples' indexes. Next, the samples' values with the calculated indexes are used to generate *RS32*. Briefly, it can be said that in the proposed generator, each *RS32* is generated from the values of five nonconsecutive ECG samples after some calculations. These samples are selected based on the value of another selected ECG sample. Then an RBS of the required length is created by concatenating multiple *RS32*.

Fig. 1 shows the proposed ECG-based random binary sequence generator scheme and Table 1 summarizes the notations used in this paper. To form n random sequences of *RS32*, which can be represented by the set *RS32_SET*={$RS32_1$, $RS32_2$, …, $RS32_n$}, the ECG signal is acquired and sampled for a specific duration. Let the number of acquired ECG samples be *len*, thus the set of samples should be {ecg_1, ecg_2,…, ecg_{len}}. Each $RS32_i$ is generated based on the values of five samples of the samples set, where i (ranges from 1 to n) represents the index of the generated *RS32*. The five samples are selected based on: i, ecg_i, *len*, and a set of predefined secret values a, b, c, d, e.

To generate the set *RS32_SET* and then use its elements ($RS32_i$) in the generation of different-lengths random sequences, e.g., 128-bit RBS (*RS128*), the following steps can be performed:

1) Select ecg_i value from the samples set. Then extract *X1, X10,* and *X100*, where *X100*, *X10*, and *X1* are three digits equal to the tenths, hundredths, and thousandths of *ecgi,* respectively, given that ecg_i value ranges from -1 to 1. If ECG samples values have a different range, they must be scaled to be within the range {-1,1}.
2) Calculate f_1, f_2, f_3, f_4, f_5 based on the values of *X1, X10, X100*, i, *len*, and the secret values a, b, c, d, e as shown in (1)-(5):

$$\begin{aligned} f_1 &= F1\left(X1, X10, X100, a, i, len\right)\\ &= (i \times X100 + X10 \times a \times i + X1)mod(len\text{-}i) \end{aligned} \tag{1}$$

$$\begin{aligned} f_2 &= F2\left(X1, X10, X100, b, i, len\right)\\ &= (i \times X10 + X1 \times b \times i + X100)mod(len\text{-}i) \end{aligned} \tag{2}$$

$$\begin{aligned} f_3 &= F3\left(X1, X10, X100, c, i, len\right)\\ &= (i \times X10 + X100 \times c \times i + X1)mod(len\text{-}i) \end{aligned} \tag{3}$$

$$\begin{aligned} f_4 &= F4\left(X1, X10, X100, d, i, len\right)\\ &= (i \times X1 \times X100 + X10 \times d\\ &\quad \times i + X100)mod(len\text{-}i) \end{aligned} \tag{4}$$

$$\begin{aligned} f_5 &= F5\left(X1, X10, X100, e, i, len\right)\\ &= (i \times X1 \times X10 + X100\\ &\quad \times e \times i + X100)mod(len\text{-}i) \end{aligned} \tag{5}$$

where a, b, c, d, and e are positive integers greater than 0.
3) Calculate $RS32_i$ based on ECG samples set {ecg_1, ecg_2, …, ecg_{len}}, i, and f_1, f_2, f_3, f_4, f_5as shown in (6):

$$\begin{aligned} RS32_i = (uint32)\Big(\Big(&ecg_{i+f1} \times 10^9 + ecg_{i+f2} \times 10^8\\ &+ ecg_{i+f3} \times 10^6 + ecg_{i+f4} \times 10^5\\ &+ ecg_{i+f5} \times 10^3\Big) \times (i) + i\Big) \end{aligned} \tag{6}$$

4) Repeat the previous steps for $i = 1$ to $i = n$, thus generating n *RS32*, that is, *RS32_SET*.
5) Generate RBSs of L-bit length (*RSL*), where L can have one of the values 128, 256, 512, 1024, 2048, by concatenating l sequences from the resulting *RS32_SET*,

where $l = L/32$, as shown in (7):

$$RSL_j \leftarrow RS32_{j\times l+1}||RS32_{j\times l+2}||\ldots||RS32_{j\times l+l} \quad (7)$$

where j ranges from 1 to n/l

For example, 128-bit random sequences (*RS128*) can be generated by concatenating every four $RS32_i$, where $L = 128$ and $l = L/32{=}4$, as shown in (8):

$$RS128_j \leftarrow RS32_{j\times 4+1}||RS32_{j\times 4+2}|| RS32_{j\times 4+3}||RS32_{j\times 4+4} \quad (8)$$

where j ranges from 1 to $n/4$.

Using the proposed scheme, random streams longer than 128 bits can be generated by concatenating more than four *RS32*. For example, to generate 256-bit random sequences (*RS256*), every eight *RS32* are concatenated as shown in (9):

$$RS256_j \leftarrow RS32_{j\times 8+1}||RS32_{j\times 8+2}||\ldots ||RS32_{j\times 8+8} \quad (9)$$

where j ranges from 1 to $n/8$.

Similarly, 512-bit, 1024-bit, and 2048-bit random streams can be generated by concatenating the adequate number of *RS32*.

Algorithm 1 The Proposed ECG Based Random Binary Sequence Generator

```
Input: $ECG{=}\{ecg_1, ecg_2, \ldots, ecg_{len}\}$, $len$, $n$, $L$
Output: $RS32\_SET{=}\{RS32_1, RS32_2, \ldots, RS32_n\}$, $RSL$
For $i \leftarrow 1$ to $n$ do
    $X100 \leftarrow$ first digit after point.
    $X10 \leftarrow$ second digit after point.
    $X1 \leftarrow$ third digit after point.
    $f_1 \leftarrow (i\times X100 + X10\times a \times i + X1)\mathrm{mod}(len–i)$
    $f_2 \leftarrow (i\times X10 + X1\times b \times i + X100)\mathrm{mod}(len–i)$
    $f_3 \leftarrow (i\times X10 + X100\times c \times i + X1)\mathrm{mod}(len–i)$
    $f_4 \leftarrow (i\times X1\times X100 + X10\times d \times i + X100)\mathrm{mod}(len–i)$
    $f_5 \leftarrow (i\times X1\times X10 + X100\times e \times i + X100)\mathrm{mod}(len–i)$
    $RS32_i \leftarrow (\mathrm{uint32})((ecg_{i+f1}\times 10^9 + ecg_{i+f2}\times 10^8 + ecg_{i+f3}\times 10^6 + ecg_{i+f4}\times 10^5 + ecg_{i+f5}\times 10^3) \times i + i;$
    add $RS32_i$ to $RS32\_SET$
end for
select $j$, where $j$ ranges from 1 to $n/l$ and $l = L/32$
$RSL_j = RS32_{j\times l+1}||\ RS32_{j\times l+2}||\ldots||RS32_{j\times l+l}$
```

The pseudocode for the proposed generator is shown in Algorithm 1.

The value of n can be chosen depending on the number of random strings to be generated and their length L. For example, if only one 128-bit random sequence has to be generated, then n can be set to 4. In general, the value of n is chosen to be less than or equal to $len/2$ because f_1, f_2, f_3, f_4, f_5values which determine the samples contributing to the generation of *RS32*, are modulus *len-i* as shown in (1)-(5). Assuming that $n{=}len$, when i becomes close to n, then the resulting *mod*(*len-i*) will have a very limited range of values, making the choice of ECG samples that contribute to the generation of *RS32* also limited. This may reduce the randomness of the resulting strings and make it easier for an opponent to determine ecg_i values from *RS32*. It is clear that for each *RS32_SET*, n/l sequences of *RSL* can be generated.

In the proposed generator, arithmetic operations such as addition, subtraction, multiplication, and modulus are applied to obtain the desired randomness. *a, b, c, d,* and *e* are used to protect the ECG signal from exposure. These values are integer numbers selected by the user. The greater these values are, the more secure the generator is against ECG signal exposure. The minimum length of each of the secret values should be 10 bits.

IV. RESULTS AND DISCUSSIONS

A. PROPOSED SCHEME SECURITY EVALUATION

In RNG, the entropy source must be resistant to any attack that could decrease the level of entropy [23]. In our case, the entropy source is the ECG signal. Attacks against entropy can be of two types: Active and Passive [24]. In RNGs, active attacks mean that the attacker can control the entropy source [23]. In our case, controlling the subject's heartbeats is impossible. Therefore, these types of attacks are impractical against the developed generator. Passive attacks are those in which the attacker, using an identical signal acquisition platform, tries to deduce the random sequence generated by the entropy source [23]. These types of attacks are also impractical because the ECG signal differs from one person to another and is a time-variant signal. The distinctiveness of the generated RBSs is proven in section IV. B.2 which emphasizes that the attacker cannot use data from one subject to predict the RBSs generated by another.

In biometric-based RNG, supposing that the generated RBSs are public values, an opponent should not be able to retrieve the biometric values from the generated RBSs. This condition is fulfilled in our proposed scheme. Below, we demonstrate the inability of the opponent to disclose the ECG signal and the secret values *a, b, c, d,* and *e* from the resulting RBSs.

Assuming that the length of each of the secret values *a, b, c, d,* and *e* is x bits and the opponent could determine the index i for each of the generated $RS32_i$. To expose the values f_1, f_2, f_3, f_4, f_5, and then the corresponding indexes of the ECG samples that contribute to the generation of $RS32_i$, the opponent needs to accomplish $(2^3)^3 \times (2^x)^5 = 2^{9+5x}$ tries. $(2^3)^3$ represents the approximate number of tries needed to get *X1, X10,* and *X100* values, where each of them is a digit with value ranges between 0 and 9. $(2^x)^5$ represents the number of tries needed to expose the five secret values of *a, b, c, d,* and *e*. After finding *ecg* indexes, the opponent needs to find ECG samples values. If the opponent wants to disclose the values of *len* samples from the ECG signal, which is the worst case, he needs to find f_1, f_2, f_3, f_4, and f_5 for all of the resulting $RS32_i$, thus, he needs to accomplish $(2^{9+5x})^n$ tries. For example, if the opponent wants to determine f_1, f_2, f_3, f_4, and f_5 values for all of the resulting $RS32_i$ from ECG signal acquired within one second with a sampling rate of 360 sample-per-second, and if $n=len/2$ and $x = 10$, where 10 is the minimum length of each secret value, then the opponent needs to accomplish $(2^{59})^{360/2} = 2^{10620}$ tries. We used the expression "worst case" because some ECG samples might be used in generating multiple $RS32_i$, on the other hand, other samples might never be used. Therefore, the opponent may expose the same sample multiple times, and some other samples can never be exposed. Consequently, knowing all ECG samples is the worst case. Since the opponent cannot determine the index i of $RS32_i$, it becomes more difficult to expose the values of the ECG samples.

It is noticeable that the more *len* is greater than *n*, the greater the security against ECG exposure.

B. RANDOMNESS AND DISTINCTIVENESS EVALUATION

To ensure the capability of using the proposed generator in securing medical data, it is necessary to evaluate the randomness and distinctiveness of the resulting RBSs. It is insufficient to ensure randomness but also distinctiveness, which indicates that different individuals create different RBSs. The distinctiveness ensures that opponents would not be able to impose security threats on WBSNs using medical information, i.e., ECG signals from other patients [11].

To analyze the randomness and distinctiveness of the proposed generator, RBSs generated from different subjects with two different cases are tested. The first case is the normal one, where the heartbeats of the subjects are regular. In this case, the randomness generated from the entropy source (ECG signal) is at its minimum. The second case is where the subjects suffer from Arrhythmia. In this case, the randomness generated from the entropy source is better than that in the first case. RBSs from the ECG signal of 50 subjects were generated. The subjects' data (ECG signals) are divided into two datasets. The first dataset consists of the ECG signal for 25 subjects in good health (healthy subjects), which is the normal case, retrieved from the MIT-BIH Normal Sinus Rhythm Database [25] (all 18 records), and from the MIT-BIH Long-Term ECG Database [26] (all 7 records), where the sampling rate is 128 samples-per-second. The second dataset consists of the ECG signal for 25 subjects with arrhythmia (arrhythmia subjects), retrieved from the MIT-BIH Arrhythmia Database [27] (the first 25 records, i.e., from record 100 to record 201), where the sampling rate is 360 sample-per-second.

Randomness and distinctiveness evaluations were performed for random sequences of 128-bit, 256-bit, 512-bit, 1024-bit, and 2048-bit. The RBSs were generated for *len* (the number of ECG samples acquired) corresponding to a signal acquisition duration of 1, 2, or 4 seconds. The minimum acceptable signal acquisition duration is 1 second because it is preferable that the samples contain at least one heartbeat to provide more randomness.

It should be noted that the signal acquisition duration has nothing to do with the length of the resulting RS. For example, if the signal is acquired for 1 second and the sampling rate is 360 sample-per-second, at most $len/2=(1 \times 360)/2=180$ *RS32* can be generated, where *len* equals the acquisition time in seconds multiplied by the sampling rate. Every 4 *RS32* can be concatenated together to get 45 *RS128*, or every 8 *RS32* can be concatenated together to get 22 *RS256*, etc.

1) RANDOMNESS EVALUATION

We evaluated the randomness of the generated RBSs through entropy analysis and by applying the National Institute of Standards and Technology (NIST) statistical tests. In addition, we apply the health test to evaluate the behavior of the noise source (ECG signal).

a) Randomness Evaluation through Entropy Analysis

RBSs' entropy is defined as the unpredictability of the generated RBSs. Before analyzing the entropy of the resulting RBSs, the entropy source (ECG signal) was evaluated. The entropy level of the ECG signal was estimated for healthy subjects. As mentioned before, the entropy of the ECG signal of healthy subjects is less than that of Arrhythmia due to the regularity of heartbeats. The minimum entropy of the ECG samples was analyzed using the NIST 800-90B standard [28]. Entropy was estimated for the ECG signal of 25 healthy subjects according to [28]. The ECG signal is classified with a non-IID assumption. For each subject, a bitstream of 1,000,000 samples was tested using the NIST recommended software for entropy estimation.

The min-entropy of the Arrhythmia subjects cannot be estimated because each record in the MIT-BIH Arrhythmia

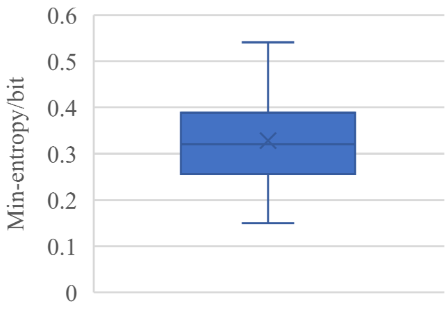

FIGURE 2. Box-and-whisker plot of the minimum entropy per bit for ECG signal samples of healthy subjects.

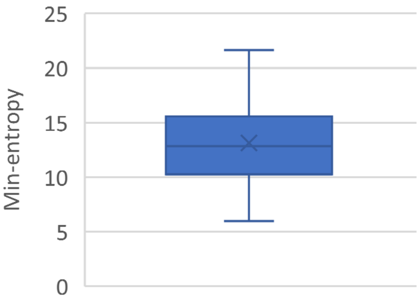

FIGURE 3. Box-and-whisker plot of the minimum entropy for the generated RS32s from the ECG signal of healthy subjects.

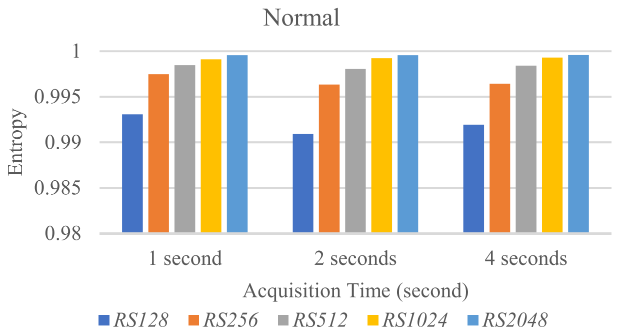

FIGURE 4. The average entropy of RBSs generated from healthy subjects.

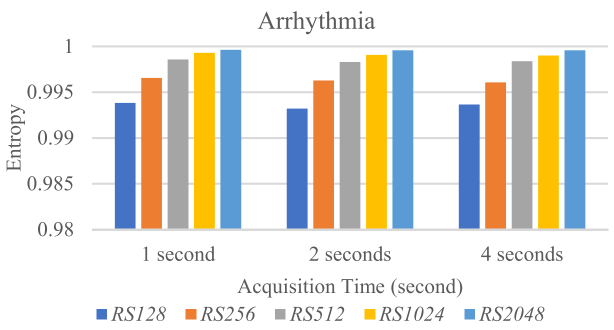

FIGURE 5. The average entropy of RBSs generated from Arrhythmia subjects.

Database contains only 650,000 samples, whereas the NIST 800-90B tests require at least 1,000,000 samples as input. Furthermore, the restart test cannot be performed because the entropy source (the heart) works continuously.

As shown in Fig. 2, the mean of the min-entropy per bit for the ECG signal samples of the 25 health subjects is 0.32823156. It is clear that the min-entropy of the noise source in our case is not high because of some periodic aspects of the ECG signal.

The min-entropy of the RBSs generated after the conditioning component is also estimated. In our generator, the conditioning component consists of all the operations applied to ECG samples that are described in Algorithm 1. From each record of the 25 healthy subject records, 1,000,000 *RS32*s were generated, and their min-entropy was calculated as described in 3.1.5 in NIST 800-90B for Non-vetted Conditioning Components [28]. Fig. 3 shows the min-entropy of *RS32*s where the mean equals 13.1291 bits. The min-entropy is low due to the formula for calculating h_{out} (the entropy of the conditioned output) shown in (10):

$$h_{out} = min(Output\_Entropy, 0.999n_{out}, h' \times n_{out}) \quad (10)$$

where *Output_Entropy* is a function of several parameters (see 3.1.5 from [28] for more details). Its output depends on the min-entropy of the noise source, which is low in our case. Therefore, the output of *Output_Entropy* has a low value, which in turn makes the min-entropy h_{out} low.

The entropy of the RBSs can be evaluated using Shannon entropy, as shown in (11):

$$H(X) = -\sum_{j=1}^{n} P(x_j) \times log_2 P(x_j) \quad (11)$$

where X is an information source with n mutually exclusive events, $x_1, x_2, \ldots, x_n$, and $p(x_j)$ is the probability of the j^{th} event. The entropy can have a maximum value of 1 if it fulfills a uniform distribution [11].

To test the randomness of the resulting RBSs, their entropy was evaluated and analyzed. The entropy was calculated for the resulting RBSs from the two datasets, the dataset of healthy subjects and the dataset of subjects with arrhythmia.

Fig. 4 shows the average entropy of RBSs generated from the dataset of healthy subjects (25 subjects). The average entropy of 75 RBSs (3 per subject) is calculated for each acquisition duration (1, 2, 4 seconds) and each RBS length (*RS128*, *RS256*, *RS512*, *RS1024*, *RS2048*). Fig. 5 shows the average entropy of RBSs generated from the dataset of subjects with arrhythmia (25 subjects). The average entropy of 75 RBS (3 per subject) is also calculated for each acquisition duration and each RBS length. As shown in Figs. 4 and 5, all of the average entropies (for each acquisition duration and each RBS length) are greater than 0.99, so they are very close to 1. It is noticed that all of the resulting averages entropies are nearly equal. The entropy of *RS2048* is the closest to 1. Sometimes it can also be noticed that the entropy of RBSs generated from arrhythmia subjects is better than that of healthy subjects. It can be justified by noting that the ECG data of arrhythmia subjects are more random than that of the healthy subjects.

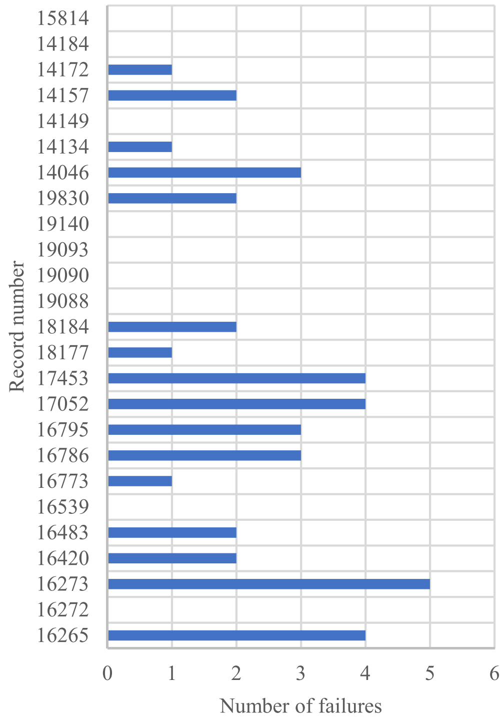

FIGURE 6. Number of repetition count test failures out of 1,000,000 tested samples from each record.

b) Health test

Health tests are tests that aim to catch the failures of the entropy source quickly and with a high probability [28]. Another aspect of the health testing strategy is determining the likely failure modes for the noise source [28].

Health tests are applied to the outputs of a noise source before any conditioning is done. NIST provides two health tests: the Repetition Count test, and the Adaptive Proportion test. The health tests were performed offline using ECG samples. The calculations were made to 1,000,000 samples obtained from healthy subjects. The tests were applied to healthy subjects for the same reasons mentioned in the previous section. α is set to 2^{-20}, and the cutoff value C is calculated for each subject depending on its min-entropy,

TABLE 2. The NIST tests results for RBSs with different lengths generated from ECG signal acquired in one second from healthy and arrhythmia subjects.

NIST Test	*RS128*		*RS256*		*RS512*		*RS1024*		*RS2048*	
	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia
Frequency test	0.4539371 (145/150)	0.466891 (149/150)	0.445182 (146/150)	0.491286 (147/150)	0.425492 (145/150)	0.488742 (148/150)	0.403181 (145/150)	0.478143 (147/150)	0.449535 (146/150)	0.506926 (149/150)
Block Frequency test	0.4861063 (150/150)	0.476131 (149/150)	0.502289 (147/150)	0.470235 (150/150)	0.465151 (149/150)	0.48316 (149/150)	0.453126 (148/150)	0.467849 (150/150)	0.466188 (149/150)	0.444678 (149/150)
Runs test	0.4894513 (149/150)	0.477067 (147/150)	0.467138 (147/150)	0.532988 (147/150)	0.453703 (150/150)	0.456062 (146/150)	0.44296 (150/150)	0.480238 (148/150)	0.478791 (149/150)	0.497732 (149/150)
Longest Run test	0.4933910 (149/150)	0.455496 (149/150)	0.471647 (150/150)	0.491179 (150/150)	0.469515 (149/150)	0.487499 (148/150)	0.503016 (150/150)	0.48583 (150/150)	0.507175 (150/150)	0.50758 (148/150)
FFT test	0.4786736 (148/150)	0.507419 (150/150)	0.445765 (148/150)	0.455913 (147/150)	0.460525 (148/150)	0.516048 (150/150)	0.471097 (148/150)	0.534245 (149/150)	0.492745 (145/150)	0.47219 (149/150)
Non-overlapping (148/148)	0.8806404 (150/150)	0.876812 (150/150)	0.693465 (150/150)	0.691438 (150/150)	0.609397 (150/150)	0.611467 (150/150)	0.595678 (150/150)	0.608489 (150/150)	0.59885 (150/150)	0.565482 (150/150)
Linear Complexity test	0.5330263 (146/150)	0.574781 (146/150)	0.501857 (148/150)	0.53163 (147/150)	0.523807 (148/150)	0.511964 (146/150)	0.517081 (149/150)	0.491068 (148/150)	0.538196 (145/150)	0.442865 (148/150)
Serial test (2/2)	0.487359 (147/150)	0.499778 (149/150)	0.514288 (148/150)	0.497818 (149/150)	0.505249 (148/150)	0.512397 (149/150)	0.508009 (147/150)	0.50701 (149/150)	0.516703 (148/150)	0.49346 (149/150)
Approximate Entropy test	0.4642176 (149/150)	0.452294 (149/150)	0.465356 (148/150)	0.504686 (149/150)	0.450087 (147/150)	0.494426 (148/150)	0.477133 (145/150)	0.48493 (148/150)	0.495937 (145/150)	0.457565 (145/150)
Cumulative Sums test (2/2)	0.4776528 (146/150)	0.484026 (149/150)	0.471015 (147/150)	0.485231 (147/150)	0.449389 (147/150)	0.475344 (149/150)	0.430528 (146/150)	0.489653 (149/150)	0.466256 (146/150)	0.506654 (149/150)

TABLE 3. The NIST tests results for RBSs with different lengths generated from ECG signal acquired in two seconds from healthy and arrhythmia subjects.

NIST Test	*RS128*		*RS256*		*RS512*		*RS1024*		*RS2048*	
	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia
Frequency test	0.418237 (146/150)	0.444315 (150/150)	0.449109 (148/150)	0.474032 (148/150)	0.395032 (145/150)	0.487659 (150/150)	0.394635 (145/150)	0.502462 (147/150)	0.430471 (150/150)	0.489063 (148/150)
Block Frequency test	0.453203 (147/150)	0.518383 (149/150)	0.472918 (145/150)	0.459901 (149/150)	0.427422 (147/150)	0.470684 (150/150)	0.432275 (147/150)	0.499396 (149/150)	0.435216 (148/150)	0.476052 (148/150)
Runs test	0.461375 (147/150)	0.541177 (150/150)	0.452958 (145/150)	0.518318 (147/150)	0.467974 (146/150)	0.537438 (150/150)	0.492123 (145/150)	0.499039 (148/150)	0.517553 (146/150)	0.550426 (148/150)
Longest Run test	0.496489 (147/150)	0.484528 (149/150)	0.49682 (150/150)	0.497948 (147/150)	0.494257 (150/150)	0.477641 (150/150)	0.510431 (148/150)	0.504717 (148/150)	0.488336 (148/150)	0.493951 (147/150)
FFT test	0.484264 (147/150)	0.491878 (148/150)	0.529008 (145/150)	0.480064 (149/150)	0.478057 (147/150)	0.497503 (149/150)	0.428912 (146/150)	0.507568 (146/150)	0.460306 (145/150)	0.50516 (146/150)
Non-overlapping (148/148)	0.880055 (150/150)	0.881541 (150/150)	0.698853 (150/150)	0.70016 (150/150)	0.614562 (150/150)	0.611163 (150/150)	0.607865 (150/150)	0.604192 (150/150)	0.569852 (150/150)	0.562963 (150/150)
Linear Complexity test	0.557688 (145/150)	0.542994 (148/150)	0.51928 (146/150)	0.510974 (146/150)	0.478893 (145/150)	0.52219 (148/150)	0.495107 (148/150)	0.500315 (148/150)	0.463939 (145/150)	0.510728 (146/150)
Serial test (2/2)	0.483262 (147/150)	0.484225 (149/150)	0.509993 (147/150)	0.542076 (150/150)	0.46917 (146/150)	0.532033 (149/150)	0.46351 (150/150)	0.504178 (148/150)	0.46152 (149/150)	0.487694 (150/150)
Approximate Entropy test	0.434853 (149/150)	0.506904 (149/150)	0.428811 (146/150)	0.515209 (146/150)	0.436112 (145/150)	0.506208 (149/150)	0.472628 (145/150)	0.535284 (149/150)	0.454319 (147/150)	0.554778 (150/150)
Cumulative Sums test (2/2)	0.463168 (148/150)	0.500383 (149/150)	0.4675 (149/150)	0.467258 (148/150)	0.416593 (146/150)	0.489002 (150/150)	0.411216 (147/150)	0.507198 (147/150)	0.453467 (150/150)	0.504684 (148/150)

TABLE 4. The NIST tests results for RBSs with different lengths generated from ECG signal acquired in four seconds from healthy and arrhythmia subjects.

NIST Test	*RS128*		*RS256*		*RS512*		*RS1024*		*RS2048*	
	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia	Normal	Arrhythmia
Frequency test	0.423602 (145/150)	0.496572 (148/150)	0.46244 (146/150)	0.505411 (150/150)	0.427315 (145/150)	0.489915 (149/150)	0.467806 (145/150)	0.493505 (150/150)	0.359222 (145/150)	0.498964 (149/150)
Block Frequency test	0.491905 (150/150)	0.45626 (149/150)	0.512393 (150/150)	0.500857 (147/150)	0.491971 (149/150)	0.504304 (147/150)	0.482289 (150/150)	0.485361 (148/150)	0.369335 (145/150)	0.538561 (150/150)
Runs test	0.489282 (146/150)	0.475512 (149/150)	0.445517 (149/150)	0.498767 (149/150)	0.4418 (147/150)	0.461012 (150/150)	0.397959 (149/150)	0.482889 (148/150)	0.424549 (148/150)	0.481619 (148/150)
Longest Run test	0.492095 (150/150)	0.486322 (149/150)	0.506059 (149/150)	0.556481 (150/150)	0.521922 (148/150)	0.524108 (148/150)	0.493468 (148/150)	0.469066 (150/150)	0.437641 (147/150)	0.465465 (150/150)
FFT test	0.473154 (149/150)	0.496539 (148/150)	0.492549 (149/150)	0.451194 (145/150)	0.484266 (149/150)	0.527761 (149/150)	0.467082 (150/150)	0.47951 (147/150)	0.517437 (148/150)	0.51086 (147/150)
Non-overlapping (148/148)	0.881451 (150/150)	0.883568 (150/150)	0.697464 (150/150)	0.698377 (150/150)	0.611055 (150/150)	0.61583 (150/150)	0.606604 (150/150)	0.604696 (150/150)	0.564882 (150/150)	0.564889 (150/150)
Linear Complexity test	0.532023 (145/150)	0.525388 (146/150)	0.487119 (147/150)	0.548289 (149/150)	0.513783 (146/150)	0.517785 (148/150)	0.504859 (147/150)	0.491369 (147/150)	0.540259 (145/150)	0.490268 (147/150)
Serial test (2/2)	0.494628 (150/150)	0.532958 (150/150)	0.523817 (150/150)	0.537888 (150/150)	0.481081 (147/150)	0.483329 (146/150)	0.469921 (147/150)	0.49085 (147/150)	0.478425 (148/150)	0.496623 (150/150)
Approximate Entropy test	0.455321 (146/150)	0.503039 (150/150)	0.437514 (147/150)	0.504106 (150/150)	0.451565 (147/150)	0.517057 (148/150)	0.413003 (149/150)	0.48722 (148/150)	0.44148 (148/150)	0.454336 (149/150)
Cumulative Sums test (2/2)	0.463227 (146/150)	0.511769 (147/150)	0.475655 (147/150)	0.518753 (150/150)	0.460115 (147/150)	0.518051 (149/150)	0.49223 (146/150)	0.495791 (150/150)	0.364521 (147/150)	0.515032 (149/150)

using the formulas declared in 4.4. from [28]. W is set to 512 because the noise source is non-binary. The ECG samples passed the Adaptive Proportion tests for all records and failed very few times in the Repetition Count test, as illustrated in Fig. 6. The maximum number of failures was 5 and the mean of the failures is 1.6.

c) Randomness Evaluation Based on NIST tests

Various statistical tests can be applied to evaluate the randomness of bit sequences and compare them with truly random sequences. The most popular test suites are the NIST Statistical Test Suite (NIST) [9], ENT [29], and Dieharder [30].

The NIST Test Suite is a statistical package consisting of 15 tests that were developed to test the randomness of binary sequences [9]. These tests focus on a variety of different types of non-randomness that could exist in a sequence. We chose the NIST tests because they focus on those applications where randomness is required for cryptographic purposes.

In our study, short RBSs are generated (ranging from 128 bits to 2048 bits), so 10 out of 15 NIST tests have been performed, which are appropriate for evaluating short random sequences. The randomness of the tested RBSs can be evaluated based on the value of α, which is called the level of significance of the test. Each test results in a P-value. We assume that RBS is random if the resulting P-value is greater than α, where $\alpha = 0.01$ [9], [11].

Several NIST tests were performed on the RBSs generated using the proposed generator to verify their randomness. Those tests were performed on RBSs with different lengths (*RS128*, *RS256*, *RS512*, *RS1024*, *RS2048*) resulting from ECG signals acquired from the prementioned two datasets (healthy and Arrhythmia subjects) for different durations (1, 2, or 4 seconds). 150 RBSs are tested from each group (6 per subject). We mean by RBSs group, the group of RBSs that have the same length and are generated from ECG signals acquired from the same dataset for a specific duration.

Equation (12) gives the minimum number of tests that must be passed for each NIST test [31]:

$$mpr = (1 - \alpha) - 3 \times sqrt(\frac{\alpha \times (1 - \alpha)}{k}) \qquad (12)$$

being α the level of significance of the test and k the number of RBSs tested. In our particular case, $\alpha = 0.01$ and k = 150, thus the minimum pass rate was 0.9656. Therefore, 145 test or more must be passed for each NIST test.

Table 2 shows the average P-value and the proportion of tests that pass each of the 10 NIST tests. These results are for RBSs generated using ECG signals of 25 subjects from each dataset (6 per subject) where the ECG is acquired in one second. Thus, 150 RBSs are generated and tested for each RBS length and each dataset. Table 2 shows that the generated RBSs passed the 10 tests because P-value was greater than 0.01 in 145 tests or more for each of the NIST tests. Therefore, the binary sequences generated from ECG signals acquired in one second using the proposed scheme can be considered random sequences.

Tables 3 and 4 are similar to Table 2, but for different acquisition durations. Tables 2-4 show that the generated RBSs passed the 10 NIST tests where P-value was greater than 0.01 in more than 145 tests of each NIST test. As a result, the RBSs with different lengths generated using the proposed scheme from ECG signals acquired in different acquisition durations from the two datasets, are random enough to be used for cryptographic purposes.

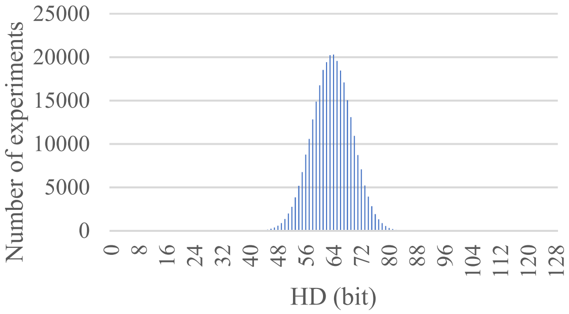

FIGURE 7. Hamming distance distribution of RS128 for different healthy subjects, where RS128 are generated from the signal acquired in 1 second.

Furthermore, Tables 2-4 show that the average P-values for almost all tests are close to 0.5. This can be justified by the uniform distribution of the average P-values. According to [9], the uniform distribution of P-values resulting from NIST tests indicates the randomness of the tested sequences. Thus, this is another confirmation of the randomness of the generated RBSs.

From Tables 2-4, it is observed that the normal and the arrhythmia cases do not differ much in the average P values. Returning to the proposed scheme, each *RS32* is generated from the values of five non-consecutive ECG samples after some calculations. Due to the non-consecutive selection of the ECG samples and the different time intervals between them, the results did not differ significantly between healthy and arrhythmia subjects, noting that the ECG signal of subjects with Arrhythmias has the same waveform as the ECG signal of healthy subjects (P, QRS, T waves), but it is characterized by different time intervals between every two successive pulses.

2) DISTINCTIVENESS EVALUATION

The distinctiveness evaluation is used to measure if RBSs generated from different subjects are sufficiently distinct. If this holds, an adversary cannot use data from another subject to predict the values generated by the target [13]. The Hamming distance (HD) is applied to measure the dissimilarity between two RBSs of the same length. It is the number of places at which the corresponding bits are different. Hence, for true random binary sequences, the average HD distribution can be nearly equal to 50% of the RBS length [11].

TABLE 5. The average HDs between RBSs from different subjects.

Signal acquisition duration	dataset	*RS128*	*RS256*	*RS512*	*RS1024*	*RS2048*
1 second	Healthy	63.52	127.09	254.32	508.69	1016.86
	Arrhythmia	63.94	127.95	256.06	511.86	1023.09
2 seconds	Healthy	63.75	127.54	255.09	510.26	1020.96
	Arrhythmia	63.95	127.87	255.81	511.31	1024.27
4 seconds	Healthy	63.84	127.64	255.35	510.75	1021.48
	Arrhythmia	63.97	127.96	255.70	511.09	1021.30

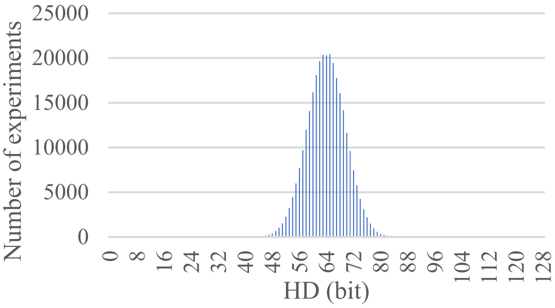

FIGURE 8. Hamming distance distribution of RS128 for different Arrhythmia subjects, where RS128 generated from the signal acquired in 1 second.

To evaluate the distinctiveness, a file of 7680 *RS32* is generated for each subject and each acquisition duration (1, 2, and 4 seconds). The *RS32* are concatenated, generating *RS128*, *RS256*, *RS512*, *RS1024,* and *RS2048*. The Hamming distance is calculated between the RBSs generated from each subject and their corresponding RBSs generated from the other subjects of the same dataset (healthy or arrhythmia subjects) and have the same acquisition duration. For *RS128*, HD is calculated between RBS pairs, where 1920 RBSs are generated from each subject. For *RS256, RS512, RS1024,* and *RS2048*, HD is calculated between RBS pairs, where 960, 480, 240, and 120 RBSs are generated from each subject, respectively.

Fig. 7 elaborates the normalized distribution of HDs of 128-bit RBSs (*RS128*). The average HDs for RBSs generated from healthy subjects equals 63.518, which is nearly equal to 50% of RBS length as revealed in Fig. 7. Fig. 8 elaborates the normalized distribution of HDs of 128-bit RBSs. The average HDs for RBSs generated from arrhythmia subjects equals 63.94, which is also nearly equal to 50% of the length of the RBS as revealed in Fig. 8. In the same manner, we can prove that HD has a normal distribution even if the RBSs are generated from healthy or arrhythmia subjects, whatever the RBS length is between 128 and 2048 bits, and whatever the acquisition duration is between 1 and 4 seconds. Table 5 shows the average HDs between RBSs of different subjects resulting from the experiments. HDs between RBSs of different lengths and different acquisition durations were tested for healthy and arrhythmia subjects. From Table 5, it is clear that RBSs generated from ECG of different subjects are distinctive and can be used for security applications in WBSNs. Therefore, opponents will not be able to threaten a WBAN for a specific subject using the ECG signal of another.

From all of the above, it is clear that the results were satisfactory for the two datasets, healthy and arrhythmia subjects, and for all of the ECG acquisition durations and all of the RBS lengths.

The experiments show that the resulting RBSs have the required randomness for all the tested acquisition durations including 1 second. In previous studies such as [11], where RBS is generated based on IPI values, the generation of

TABLE 6. The proposed generator throughput compared with previous studies.

Scheme	Throughput
Xu et al. [14]	16 bit-per-second (60 pulse-per-minute)
	26.6 bit-per-second (100 pulse-per-minute)
Pirbhulal et al. [11]	16 bit-per-second (60 pulse-per-minute)
	26.6 bit-per-second (100 pulse-per-minute)
Camara et al. [13]	184 bit-per-second (60 pulse-per-minute)
	306 bit-per-second (100 pulse-per-minute)
Our approach	2048 bit-per-second (for 128 sample-per-second)
	5760 bit-per-second (for 360 sample-per-second)

binary sequences with good randomness requires the acquisition of ECG signal for several heartbeats (several seconds).

C. PERFORMANCE EVALUATION

1) THROUGHPUT

As mentioned earlier, by using the proposed generator, a number of *RS32* equals *len*/2 can be generated as a maximum. If the sampling rate is 128 sample-per-second, 64 *RS32* per second can be generated, which is equivalent to 2048 bits-per-second. If the sampling rate is 360 sample-per-second, 180 RS32 per second can be generated, which is equivalent to 5760 bits-per-second.

Table 6 shows the proposed scheme throughput (number of random bits generated per second) compared with previous studies [11], [13], and [14]. It is clear that the proposed scheme throughput outperforms the previous ECG-based RNG. Furthermore, the throughput of the proposed generator depends on the sampling rate, in contrast to previous studies where the throughput depends on the pulse rate. Even for a low sampling rate (128 sample-per-second), the proposed generator is still having the best throughput compared to previous studies. In our generator, the ECG signal perturbations, e.g., variation in ECG signal frequency, do not affect the throughput because the throughput is only affected by the sampling rate.

Unlike other RNGs, our proposed generator generates a large number of RBSs in a limited time. RNGs often have low throughput as a result of their dependence on natural phenomena, so they usually generate short random sequences over a relatively long time. PRNGs are used when the generation of many random sequences within a short time is required. Although our proposed generator is an RNG, it overcomes the low throughput of RNGs and has a high throughput as if it is a PRNG.

In addition, the min-entropy rate (the min-entropy throughput) was studied based on the min-entropy for the generated *RS32* which was calculated before. The min-entropy rate equals min-entropy per bit × throughput. We calculate it only for healthy subjects for the same reason as mentioned before. Fig. 9 shows the min-entropy rate calculated for the RBSs generated from the ECG signal of 25 healthy subjects. The mean of the min-entropy rate is 840.267 bit/s. Returning to Table 6 and according to the resulting min-entropy rate, it can be concluded that, although the calculated min-entropy rate is the lowest throughput of our generator, nevertheless it is still better than the throughput of all the previous studies.

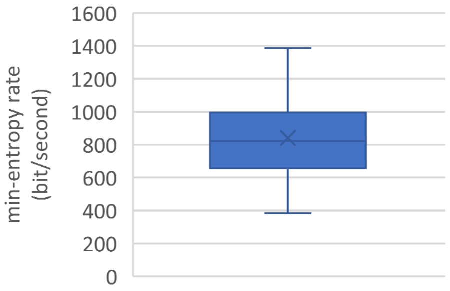

FIGURE 9. Box-and-whisker plot of the min-entropy rate for RBSs generated from ECG signal of healthy subjects.

2) COMPLEXITY

The proposed generator has a time complexity of O(n). Returning to Algorithm 1, it appears that the proposed generator is consist of one loop with n rounds. The loop body includes functions with operations such as addition, subtraction, multiplication, and modulus. The time complexity of each round equals O(1), consequently, the time complexity of the proposed generator equals O(n), where n is the number of the generated *RS32*.

It can be noted that the simplicity of mathematical and processing operations may lead to low resource consumption. However, despite the simplicity of the operations, the proposed generator can be used for cryptographic purposes.

V. CONCLUSION

In this paper, we proposed a new biometric-based random sequence generator that uses ECG signal samples to generate RBS. Its throughput is better than that of previous studies tens or hundreds of times. It generates RBSs of different lengths.

Proceeding from the fact that the simpler the processes, the lower the resource consumption, the proposed generator is designed to be based on very simple operations like addition, subtraction, etc., rather than complex ones like wavelet transforms, and hash functions used in previous studies.

Due to the distinctiveness of the generated sequences, the opponent cannot predict the random values generated from one subject using the ECG signal of another. Furthermore, research shows that IPI values can be eavesdropped from a distance using cameras. Fortunately, this approach is useless with our generator, because it is based on ECG signal samples instead of IPI, which cannot be detected from a distance. Moreover, the opponent cannot expose the ECG signal depending on the resulting RBSs.

There is also an additional feature that is worth mentioning. The proposed generator does not support external perturbation because the opponent cannot control the noise source (the heartbeats) of the subject.

Randomness and distinctiveness are empirically verified for RBSs generated from healthy subjects, which is the worst-case scenario, and arrhythmia subjects. It is proved that the generated RBSs are random enough to be used for cryptographic purposes.

As for future works, we seek to study the maximum length of the generated random sequences that ensure the required randomness.

REFERENCES

[1] S. Al-Janabi, I. Al-Shourbaji, M. Shojafar, and S. Shamshirband, "Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications," *Egyptian Inform. J.*, vol. 18, no. 2, pp. 113–122, 2017.

[2] Y. Al-Saeed, E. Eldaydamony, A. Atwan, M. Elmogy, and O. Ouda, "Efficient key agreement algorithm for wireless body area networks using reusable ECG-based features," *Electronics*, vol. 10, no. 4, p. 404, Feb. 2021.

[3] T. Jabeen, H. Ashraf, and A. Ullah, "A survey on healthcare data security in wireless body area networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 10, pp. 9841–9854, Jan. 2021.

[4] M. Masdari, S. Ahmadzadeh, and M. Bidaki, "Key management in wireless body area network: Challenges and issues," *J. Netw. Comput. Appl.*, vol. 91, pp. 36–51, Aug. 2017.

[5] R. Negra, I. Jemili, and A. Belghith, "Wireless body area networks: Applications and technologies," *Proc. Comput. Sci.*, vol. 83, pp. 1274–1281, Jan. 2016.

[6] N. Mahmoud, S. El-Sappagh, H. M. El-Bakry, and S. Abdelrazek, "A real-time framework for patient monitoring systems based on a wireless body area network," *Int. J. Comput. Appl.*, vol. 176, no. 27, pp. 12–21, Jun. 2020.

[7] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "PSKA: Usable and secure key agreement scheme for body area networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 1, pp. 60–68, Jan. 2010.

[8] H. Zhao, C. Chen, J. Hu, and J. Qin, "Securing body sensor networks with biometric methods: A new key negotiation method and a key sampling method for linear interpolation encryption," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, Aug. 2015, Art. no. 764919.

[9] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-22rev1a, 2010.

[10] C. Zenieh, "A new biometric based secret key exchange scheme for wireless body area networks," *Damascus Univ. J. Eng. Sci.*, vol. 37, no. 3, pp. 65–78, Oct. 2021.

[11] 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.

[12] G. Lo Re, F. Milazzo, and M. Ortolani, "Secure random number generation in wireless sensor networks," *Concurrency Comput., Pract. Exper.*, vol. 27, no. 15, pp. 3842–3862, Oct. 2015.

[13] 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, p. 2747, Aug. 2018.

[14] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li, "IMDGuard: Securing implantable medical devices with the external wearable guardian," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1862–1870.

[15] G.-H. Zhang, C. C. Y. Poon, and Y.-T. Zhang, "Analysis of using interpulse intervals to generate 128-bit biometric random binary sequences for securing wireless body sensor networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 1, pp. 176–182, Jan. 2012.

[16] G. Zheng, G. Fang, R. Shankaran, M. A. Orgun, J. Zhou, L. Qiao, and K. Saleem, "Multiple ECG fiducial points-based random binary sequence generation for securing wireless body area networks," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 3, pp. 655–663, May 2017.

[17] G. Chen, "Are electroencephalogram (EEG) signals pseudo-random number generators?" *J. Comput. Appl. Math.*, vol. 268, pp. 1–4, Oct. 2014.

[18] D. Nguyen, D. Tran, W. Ma, and K. Nguyen, "EEG-based random number generators," in *Proc. Int. Conf. Netw. Syst. Secur.* Berlin, Germany: Springer, 2017, pp. 248–256.

[19] S. R. Moosavi, E. Nigussie, S. Virtanen, and J. Isoaho, "Cryptographic key generation using ECG signal," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 1024–1031.

[20] B. Narwal and A. K. Mohapatra, "A survey on security and authentication in wireless body area networks," *J. Syst. Archit.*, vol. 113, Feb. 2021, Art. no. 101883.

[21] A. Calleja, P. Peris-Lopez, and J. E. Tapiador, "Electrical heart signals can be monitored from the moon: Security implications for IPI-based protocols," in *Information Security Theory and Practice* (Lecture Notes in Computer Science), vol. 9311. Cham, Switzerland: Springer, 2015, pp. 36–51.

[22] C. Zenieh, "Random bit sequence generator based on ECG signal," *Damascus Univ. J. Eng. Sci.*, vol. 38, no. 1, pp. 30–42, Mar. 2022.

[23] F. Răstoceanu, R. Rughiniş, Ş.-D. Ciocîrlan, and M. Enache, "Sensor-based entropy source analysis and validation for use in IoT environments," *Electronics*, vol. 10, no. 10, p. 1173, May 2021.

[24] W. Stallings, *Cryptography and Network Security Principles and Practice*, 5th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2011.

[25] *MIT-BIH Normal Sinus Rhythm Database*. Accessed: Apr. 2022. [Online]. Available: https://physionet.org/content/nsrdb/1.0.0/

[26] *MIT-BIH Long-Term ECG Database*. Accessed: Apr. 2022. [Online]. Available: http://physionet.org

[27] *MIT-BIH Arrhythmia Database*. Accessed: Apr. 2022. [Online]. Available: http://physionet.org/physiobank/database/mitdb/

[28] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle, "Recommendation for the entropy sources used for random bit generation," NIST, MA, USA, Tech. Rep. 800-90B, Jan. 2018.

[29] J. Walker. (Jan. 2008). *ENT—A Pseudorandom Number Sequence Test*. Fourmilab. [Online]. Available: https://www.fourmilab.ch/random/

[30] R. G. Brown. (2011). *Dieharder: A Random Number Test Suite*. [Online]. Available: https://webhome.phy.duke.edu/ rgb/General/dieharder.php

[31] 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, Apr. 2019.

CHRISTINE ZENIEH was born in Damascus, Syria, in 1989. She received the B.S. and M.S. degrees in computer engineering from Damascus University, Damascus, in 2012 and 2017, respectively, where she is currently pursuing the Ph.D. degree in computer and networks engineering with the Computer and Automation Engineering Department, Faculty of Mechanical and Electrical Engineering.

She is a member of the Technician Assembly with the Computer and Automation Engineering Department, Faculty of Mechanical and Electrical Engineering, Damascus University; and a member of the Technician Assembly with the Faculty of Computer Engineering and Informatics, Syrian Private University, Damascus. Her research interests include networking, networks security, and information security.

MOHAMED MAZEN AL-MAHAIRI received the B.Sc. degree in electronic engineering from Damascus University, Damascus, Syria, in 1986, and the Ph.D. degree in computer organization and architecture from Saint Petersburg Electrotechnical University "LETI," Russia, in 1993.

He is an Associate Professor with the Computer and Automation Engineering Department, Faculty of Mechanical and Electrical Engineering, Damascus University. His research interests include multiprocessor systems, embedded systems, and design. He is a member of Distinction and Creativity Agency, Syrian Computer Society, and *Damascus University Journal*.

MOUFID HADDAD was born in Hama, Syria, in 1963. He received the B.S. degree in electronic engineering from Aleppo University and the Ph.D. degree in computer networks from Saint Petersburg Electrotechnical University "LETI," Russia, in 1987 and 1994, respectively.

He is currently an Academic Member of the Technician Assembly with the Computer and Automation Engineering Department, Faculty of Mechanical and Electrical Engineering, Damascus University, Damascus, Syria. His research interests include networking and networks security.