Received: January 20, 2023. Revised: April 7, 2023. 440

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Securing Signal Encryption Based on Reduced Round Homomorphic AES

Areej A. Ahmed[1,2]* **Magda M. Madboly[1]** **Shawkat K. Guirguis[1]**

[1]*Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria, Egypt*
[2]*Research and Development Center, Ministry of Electricity, Baghdad, Iraq*
* Corresponding author's Email: areej_abdulmunem@yahoo.com

Abstract: Cryptography plays a vital role in protecting information that has increased as a result of digitalization. Personal sensitive information, including electrocardiogram (ECG) signals, is widely transferred around the world, and as a result, protecting the data from unauthorized access by attackers is critical. One of the algorithms that is frequently used is the advanced encryption standard (AES) algorithm, due to its remarkable reliability and its usage in a wide range of applications. However, key exchange is still necessary to execute computations on the encrypted data, whereas time is considered the essence of the efficiency of the encryption algorithm. This study proposes and investigates the use of the proposed improved reduced round AES algorithm in conjunction with the new proposed fully homomorphic encryption (FHE) in order to preserve data privacy and eliminate key exchange restrictions. This research was employed to process, encrypt, and decrypt digital ECG signals. The suggested algorithm improves the security level and encryption and decryption times by using fewer rounds of encryption. In order to achieve this goal, it was suggested that the evaluation process be added to the AES algorithm as an extra level of security. With the evaluation process, it is possible to execute a number of computation operations homomorphically on the encrypted data without decrypting it. Additionally, the number of rounds in the AES encryption was decreased from 10 to 5. Results indicate that the proposed algorithm would take 5.39475×10^{32} years to break, making it more efficient than the traditional AES and others. Also, the proposed algorithm has a classification sensitivity of 95.83% while simultaneously displaying an accuracy of 100%. Additionally, it may be utilized for real-time internet of things (IoT) applications.

Keywords: Cryptography, AES, Reduced round, Security, Homomorphic encryption (HE), Encryption time.

1. Introduction

Cryptography is a popular technology for securing data by keeping it safe from prying eyes. There has been real growth in the evolution of cryptographic algorithms during the previous two decades [1]. AES stands out as the most effective, extensively used, and reliable cryptographic algorithm due to its non-linear structure, which prevents any unauthorized access [2]. Preserving personal information must be maintained during three different stages: 1. Acquisition, when personal data is obtained from the user; 2. Storage, during which the data is kept in the cloud for later access; and 3. Computation, which involves processing the data for future analysis. The current AES encryption algorithm can achieve data privacy only in the acquisition and storage stages. Unfortunately, maintaining data privacy during the computation stage requires transmitting the data between the cloud and the mobile device repeatedly. Throughout this transmission, personal data should be encrypted; therefore, in the cloud, it requires key exchange to decrypt data before performing computations. Then, it must be re-encrypted to be sent back to the user's device. It is generally agreed that cryptography has to be reinforced and that new variations and alterations should be made available [3]. So instead of this traditional approach, an emerging improved reduced round AES with a new fully homomorphic encryption algorithm is proposed. The FHE

technique is especially effective when computing is performed in the cloud or by a third party, whereas data privacy must always be maintained. Through the use of this approach, it is possible to encrypt data, store it, and then perform computation on the encrypted data in the cloud without showing it. This means the data can only be decoded by the owner of the secret key. This study encrypts ECG signals and demonstrates its usefulness in real-time IoT contexts, enabling safe channel transfer. In the proposed AES algorithm, the number of cipher rounds is reduced to five to minimize the encryption and decryption times and detect its effect on the algorithm's strength. The effect of sampling frequency on the signal's accuracy is also examined in this research, as is the effect of ECG length on the system's performance. At first, electrocardiogram signal preparation is accomplished utilizing the Pan and Tompkins approach [4]. Afterward, the suggested algorithm is used to encrypt an ECG, which provides an evaluation process as a linear polynomial function with the three recommended keys. These keys protect ECG signals from prying eyes, keep sensitive information private, and fortify the proposed algorithm against hacking. The ECGs are then classified. Lastly, the security of the suggested algorithm was evaluated. Results indicate that, compared to others, the proposed algorithm is faster at encrypting data. Also, it is shown that the standard AES's throughput is equivalent to 34% of the suggested algorithm's throughput for encryption and equal to 44% for decryption. Furthermore, the suggested changes do not reduce the security of the standard AES because the algorithm's nonlinear character is preserved but rather enhanced. Compared to a typical AES algorithm, the necessary time to crack the cipher is multiplied by over 50 million times, making it more resistant to attacks.

The remaining sections are grouped as follows: Section 2 details the previous literature. Section 3 discusses the suggested work and its methodology. Section 4 presents and compares the findings from the analysis of the suggested algorithm to those of the standard algorithm and other works. Section 5 talks about the conclusion.

2. Related work

Homomorphic encryption is a collection of conceptually safe encryption techniques that allow mathematical operations to be done directly on the ciphertext rather than the plaintext [5]. HE techniques were categorized into: Partially homomorphic encryption (PHE): In such a technique, it is limited to performing a single mathematical operation (adding or multiplying) on the encoded data at a time, as in Pailler (additive HE) and RSA (multiplicative HE) [6]. Somewhat homomorphic encryption (SHE): It performs a restricted number of several mathematical operations on the encoded data. Fully homomorphic encryption (FHE): This technique permits an infinite number of mathematical operations on the encoded data, such as Gentry [7]. This section surveyed the state of existing literature on encryption in the cloud.

The AES algorithm was employed to protect the ECG signals while transmitting them through an untrusted channel [8]. The ECG signal was encrypted before being sent, and once it reaches its recipient, the signal is decrypted so that analysis can be performed on it. Xilinx and Python are used to apply the AES Protocol.

Firstly, the ECG signals are gathered from the MIT-BIH arrhythmia database. At first, timestamps are taken from the signals and saved in a text file, which is then converted to binary representation. After that, the encryption process is applied by using a random 128-bit key generated by the AES. The decryption process is performed by reversing all of the previous encryption steps. In this system, the decryption process requires exchanging the key to perform the diagnosis. So that data will be exposed to an unauthorized user. Therefore, this system doesn't keep the data private and doesn't provide internal diagnosis.

A hybrid FHE-Gentry combined with AES in cloud servers is presented to maintain efficacy, reliability, and integrity [9]. The effectiveness of their approach for continuous monitoring of patient health in identifying potential problems was examined. The proposed method proves the need for a recryption process to eliminate decoding mistakes. According to the findings of the hybrid method, the recryption process significantly requires more time compared to other processes (key generation, encryption, and decryption) since it is more complex. In other words, the recryption process consumes 99.9% of the total execution time. This is a huge percentage and cannot be underestimated. As a result, the suggested algorithm is considerably more impractical by using Gentry's FHE method. Also, it is mentioned that when computing the heart rate for a stored ECG signal, the algorithm requires high computational and storage resources, requiring a novel strategy for storing data on the cloud.

A safe and confidential ECG system was proposed [10]. This study merges the FHE method with signal processing via the QRS complex. It encrypts and decrypts the ECG signal. The FHE

method is applied to two thresholds. The encoded threshold values transform the signal; therefore, only the physician can interpret the result. This work has a notification or warning feature if the detection is dangerous. Nevertheless, this effort is constrained by the problem of greater data, which impairs the effectiveness of the system security. Despite using the FHE algorithm in this system, it was applied to only one operation (the addition), while the multiplication operation was not performed. Also, this work lacks security analysis.

An approach for analysing ECGs with R-peak detection was proposed by [11]. This method has been proven to be effective, quick, and flexible. It's also a step forward from the method proposed by Pan and Tompkins. However, this approach is limited to acquiring electrocardiogram (ECG) information. It fails to establish a connection with a server-side database, and thus it cannot access further data for analysis. Signals were not even encrypted, which is a major problem must be solved.

A security method that mainly involves the utilization of PHE methods to protect the ECG signals [12] was presented. This is done to ensure privacy and to stop any distortion, copying, or duplication of the data. In this work, the ECG signal has been encrypted with the rivest-shamir-adleman (RSA) technique by utilizing the public key. The result demonstrates that the RSA method achieves an accuracy of 90%. In spite of this percent, this work doesn't have any security analysis. Also, it is a multiplicative algorithm that depends on the natural calculations of the RSA algorithm.

The AES algorithm was shown and how it may be used with the Huffman coding method to encrypt ECG data within the electronic healthcare application [13]. Also, a lossless compression method is used in order to achieve lossless data compression. Despite this, the time necessary to compress and transmit the data reduces the overall efficiency of this system. Also, findings showed that, while encrypting the whole ECG signal, the AES algorithm was computationally expensive. Health information has the potential to save or cost a person's life, so it's crucial that it be transferred as fast as possible and maintained efficiently, especially in real time. Since this is the case, therefore, the AES algorithm needs to be modified so that it may be used in IoT healthcare systems.

A FHE-based algorithm was implemented for safe and reliable ECG data transmission [14]. Once data is gathered from the MIT-BIH database, the ECG signal undergoes preprocessing before being encrypted using the Pan and Tompkins algorithm. Homomorphic addition is then implemented on the ECG data by employing the Gentry FHE of the SDC algorithm. Although the security of the system is improved by demanding the private key subsequent to the homomorphic procedure, only a limited number of signals (27 of 48) were taken from the MIT-BIH arrhythmia database to assess its efficiency. This refers to the existence of bias for some of the signals. In spite of the fact that this system is using a FHE, but it was employing and evaluating only the addition operation instead of the addition and multiplication operations. Furthermore, the system has a detection error of about 14.8%.

A strong hybrid ECG encryption method dependent on multiple DNA layers and AES was introduced [15] to decrease the time it takes to encrypt data and increase security for healthcare applications. The suggested method successfully demonstrated its ability to secure signals while utilizing cloud computing. However, the study's scope is limited to ECG signals rather than other types, which may have different transmission requirements and metrics. Despite the fact that the proposed method's DNA rules generate and execute four keys, it takes longer to encrypt data, rendering it inappropriate for use in digital real time medical applications.

A new, multiple-level protection method was created that makes use of ECG signals, an AES algorithm that has been applied with the signal processing, and an artificial bee colony [16]. This method can prevent plain-text attacks on ECG signals transmitted over healthcare applications by producing the primary key and a set of rule keys. In this technique, encrypting ECG data shortens the signal length and, by extension, the number of necessary computation processes. Despite the fact that using this method increases security, it lacks any built-in diagnostic capabilities. Because it relies on key exchange (key sharing) and can't verify user authenticity, it is unsafe. Also, it cannot connect to the ECG database and evaluate the acquired signals, so it takes up a lot of storage space.

A frequency-domain watermarking technique was introduced [17] for encoding sensitive patient data within ECG signals. In this technique, the signal is converted into a two-dimensional image, and the frequency information of the image is then captured via the integer wavelet transform. Then the 2D image is converted into a 1D signal that is in accordance with the ECG watermarked signal. As a final step, the obtained coefficients are processed, and the watermark bits are incorporated. This technique was evaluated on the electrocardiogram records available in the MIT-BIH Arrhythmia Database. According to the results, the generated

watermark signal is nearly identical to the host signal. And this is acceptable enough. This technique is solid against many attacks. Since this technique does not encode the watermark, it is theoretically possible for a third party familiar with the integration process to read the medical records by removing the watermark.

The suggested improved reduced round AES with the FHE algorithm is more comprehensive and goes beyond the scope of prior studies with lightweight features while improving security, privacy, and classification sensitivity.

3. Research methodology

The ECG signals were gained from the MIT-BIH Arrhythmia database [18] to evaluate the effectiveness of the proposed algorithm. In all, this database has 48 patients' heartbeat recordings, each of which lasted for 30 minutes in a resting state. Each ECG recording is 21600 samples long, with a sampling rate of 360 Hz [19]. The heart rate ranged from 24 to 173 beats per minute, according to this database.

This research is an extension of our previous research [20] which was the foundation for this one. This study suggested an Improved Reduced Round AES algorithm that was combined with our earlier proposed FHE algorithm which is based on computationally lightweight a 4 × 4 matrix of bytes in order to protect sensitive data being transmitted over a network from being accessed by an unauthorized third party during computations. This research is an extension of our previous research which was the foundation for this one. Two changes were made to AES: a decrease in the encryption iteration from 10 to 5 rounds and adding our proposed FHE algorithm as an additional layer. The reduced round number decreases the processing time. The FHE strengthens the AES. Fig. 1 presents the proposed reduced rounds homomorphic AES methodology.

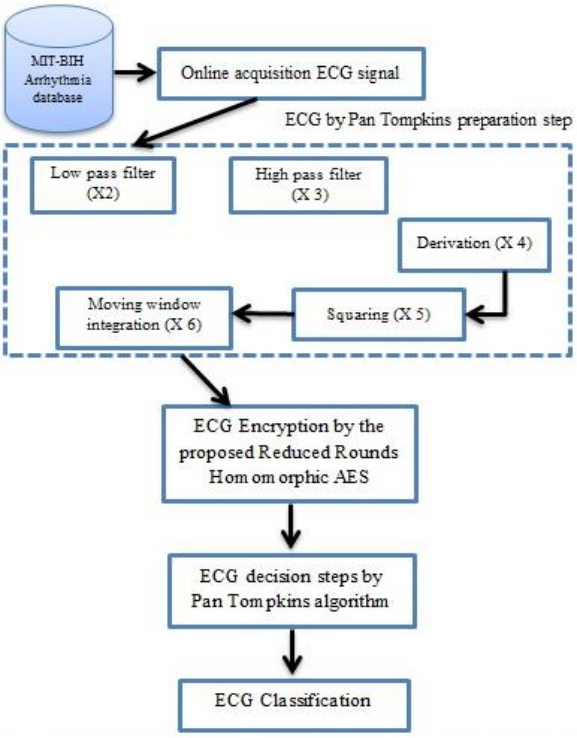


Figure. 1 The proposed reduced rounds homomorphic AES methodology

3.1 Data preparation

The Pan Tompkins method is often used to recognize QRS complexes due to its higher sensitivity and accuracy [21]. This method is broken into preparation and decision steps. The QRS complex is defined as a collection of the Q, R, and S waves of the electrocardiogram. It is typically the focus of attention, and it indicates ventricular depolarization [22]. In this study, it is suggested to use this algorithm with the ECG signal at sample rates of 100, 200, 300, and 400 Hz instead of only one sampling rate (200) to verify the effect of sampling frequency on the accuracy. In the preparation step, the original signal (X1) goes through the following processes after acquired online from the database:

1. The band-pass filter: The band-pass filter contains both low-pass filter (LPF) and high-pass filter (HPF).The band-pass filter decreases the effect of noise and incorrect detection due to signal artifacts. Low-pass filter (X2) was employed to eliminate high-frequency noise while still capturing the low-frequency signals. The low-frequency noise was filtered out using a high-pass filter (X3).
2. The derivative process (X4): The QRS complex was distinguished from other waves during the derivation procedure. The P- and T-waves, which have a low frequency, were quelled. And the signal-to-noise ratio improves. This increases the overall sensitivity of the detector.
3. Squaring function (X5): The signals were squared, which amplified the already considerable QRS complex's peak amplitudes. The incorrect detection is caused when the larger amplitudes of T-waves can be decreased by the square function.
4. Moving window integration (MWI) (X6): This process is done to get data from the waveform characteristics, including the slope of the R wave.

After the preparation of the signal, it is time for the decision step, which is done after the data encryption. The thresholds are employed in this

process to determine if the MWI's output is a QRS complex or noise peak. The latter is discarded later. The threshold is just a little higher than the peak noise levels, and it is updated automatically to enhance QRS complex detection. Then the RR interval is determined to measure the heart rate.

3.2 Data encryption

AES qualifies as a symmetric cryptographic algorithm with a block size of 128 bits. 128, 192, and 256 bit keys are supported. It depends on the number of rounds. So, when encrypting with a 128 bit key, it takes 10 rounds, whereas 192 bit and 256 bit keys demand 12 and 14 rounds, respectively. ECG signals were acquired and read online from the MIT-BIH arrhythmia database.

AES is a great design space in which to investigate FHE techniques, as it allows for both parallel and algebraic calculation. It was also effective across a number of other platforms. The proposed algorithm is presented as fully homomorphic encryption to accommodate the necessary operations (addition and multiplication) to calculate the heart rate.

To get ready for ECG encryption, and after an ECG signal is gained from Moving Window Integration by the Pan Tompkins method, it is time to convert ECG signals (x6) to integer representation (by multiplying them by 10^6 by applying a MATLAB function) and in 4 × 4 square matrix. The proposed 128-bit block AES algorithm based on a 4 × 4 matrix of bytes will be basically described below:

1. The initialization of AES components and the key generation procedure: The proposed algorithm provides a single key, or "secret key," which is a random 128-bit (16-byte string as a 4 × 4 byte integer matrix), as the means of encryption. The encryption keys should be stored in a separate location from other system parts, and they should be accessible only by the final user. The key generation process as in Eq. (1):

$$(\text{ke}) \leftarrow \text{KeyGen}(\text{key}) \tag{1}$$

where key represents security variable, and ke represents secret key. Then, the round keys (R) (constant number) are generated from the encryption key so that the Key Expansion function can utilize them later. In the proposed algorithm, the round number is reduced from 10 to 5 rounds in order to improve the encryption and decryption times and meet the requirements of recent healthcare systems. Each round in an AES algorithm must have its own distinct key block. Both the polynomial transformation matrix and the inverse transformation matrix are made so that they can be used in MIX columns.

2. The AES encryption procedure: In the encryption process and at the sender's side, 16 bytes of plaintext (pla) are transformed into 16 bytes of ciphertext (C) utilizing the encryption key (ke) as shown in Eq. (2).

$$(C) \leftarrow \text{AES Enc}_{ke}(\text{pla}, \text{R}) \tag{2}$$

where pla is the plain integrated ECG signal (x6), R is the round key, and C represents the ciphertext. The plaintext and ciphertext of the proposed technique were both laid out in a 4 by 4 matrix. Alternatively, and more precisely, the sender encrypts the plain ECG signals using the secret key by performing four consecutive operations in each of the five rounds: the the SubBytes, ShiftRows, Mix Columns, and AddRoundKey operation. All encryption and decryption standard rounds have these four functions except the last one, which requires just three functions (no MixColumns). The output of the last round of execution is taken and considered to be the ciphertext (C), which is then transmitted to the right intended destination.

3. The AES evaluation procedure: As with any other homomorphic algorithm, AES must have an evaluation procedure to realize the homomorphic encryption concept. Prior to decryption, an evaluation process is performed on the server side. The evaluation procedure takes ciphertexts (C) that are obtained from the encryption procedure and generates alternative ciphertexts (C'), also in a 4 × 4 matrix format. Its aim is to apply arithmetic operations to the ciphertext rather than the plaintext in a homomorphic manner. The evaluation function that is applicable to all computations is denoted by f, which is described in Eq. (3).

$$(C') \leftarrow \text{Eval}_{\text{R},HR}(f, C) \tag{3}$$

where f is defined as the generic evaluation function, and C' is an evaluated ciphertext. The suggested algorithm forms f into a Linear Polynomial Function [23] owing to its simplicity and efficiency, which is utilized to generate a new ciphertext and provides more security. Both evaluation keys, the round key and the heart rate, are proposed to be used in the presented evaluation function in order to create the alternative ciphertexts (C') as shown in Eq. (4).

$$(C') = \text{R} * C + HR \tag{4}$$

where R represents the round key, and HR is the heart rate. It was suggested that (HR) and (R) be variable values instead of constant values to make the algorithm safer and impossible to break. Variable values are expensive to hackers because they must be verified if they're variables, which complicates decryption. Constant values reflect the probability of a single result, whereas variable values give several results. To further improve the security, it is recommended that the HR become a value containing an integer of four decimals that ultimately contributes to the algorithm becoming more secure. The threshold has a role in this effect, which is calculated using decimal places. Besides determining HR using Eq. (5):

$$\text{HR (bpm)} = \frac{\text{Signal size}}{\text{time (m)}} \tag{5}$$

where bpm = beats per minute. The hypothesized algorithm is Fully Homomorphic, which means it can handle both multiplicative and additive homomorphic characteristics, as shown in Eq. (4). The precise decoding of ciphertext is crucial to the success of homomorphic encryption. As a result, once the evaluation is complete, the ciphertexts should remain in the same format. In addition, the size of the ciphertext should be stable.

4. The AES decryption procedure: The decryption procedure consists of reversing both the evaluation and the encryption steps. In other words, decryption is done by using the inverse of the evaluation function (a linear polynomial function) and then the four AES decryption functions with the secret key to get plaintext out of 128 bits of ciphertext, as shown in Eqs. (6) and (7):

$$(C) = \frac{C'}{R} - HR \tag{6}$$

$$(\text{pla}) \leftarrow \text{Dec}_{ke}(C', R) \tag{7}$$

Furthermore, the same number of rounds is used for decryption with inverse transformation, and the order of round functions differs, namely, InvShiftRow, InvSubByte, AddRoundKey, and InvMixColumn. Similar to the Mix-Column function, the Inv-Mix-Column needs a matrix. If the two matrices are inverses of one another, it is simple to show that the two transformations are inverses of one another [13].

3.3 Data classification

The original value of the ECG signal will be different after encryption, the heart rate will be computed, and the signal will be classified as having a normal or abnormal heartbeat. Normal means the heart rate ranges from 60 to 100 bpm, while abnormal heartbeats represent arrhythmia, which is either bradycardia if the heart rate is less than 60 beats per minute or tachycardia if the heart rate is greater than 100 beats per minute. This will enable the researcher and analysts to do deeper analysis and diagnosis.

Statistical analysis is then used to determine the algorithm's efficiency by calculating several metrics, as shown in the next section.

4. Results and analysis

The proposed algorithm was simulated using a recognized simulation program, "MATLAB version R2018b." In this study, an Intel ® Core ™ i5-4200M CPU processor running at 2.5 GHz and 8 GB of RAM were utilized.

4.1 Analysis of accuracy and classification sensitivity

Arrhythmias can be identified by online acquiring and analysing ECG readings from the MIT-BIH database. At first, the Pan and Tompkins technique is used for ECG preparation. After that, the signals were encrypted using the proposed algorithm. In this study, the suggested algorithm is tested by using all of the samples from the MIT-BIH database at a sampling rate of 360 Hz. After decryption, the proposed system classifies the signal as normal, tachycardia, or bradycardia. A normal heartbeat when the heart rate (HR) is between 60 and 100 beats per minute (bpm). The HR is abnormal bradycardia when the HR is less than 60 bpm or tachycardia when the HR is greater than 60 bpm. The classification efficiency of the algorithm is evaluated via statistical analysis by calculating its sensitivity level. The classification sensitivity measures the percentage of the algorithm's capability to identify true pulses, as in Eq. (8). And the performance of the decryption process is evaluated through its accuracy, as in Eq. (9).

$$Sen\ (\%) = \frac{TP}{TP+FN} \tag{8}$$

$$Accuracy\ (\%) = \frac{TP}{TP+PN+FN} \tag{9}$$

Where TP is the number of true positive beat detected, FP is the number of false positive beat, and FN is the number of false negative beat [24].

Table 1 shows the results of the experiment. There is only a slight distinction in heart rate

Table 1. Results comparison between the proposed algorithm and MIT-BIH database

ECG Sample No.	HR measured by the Proposed system (bpm)	HR in MIT-BIH (bpm)	Classification
100	75.5337	70-89	Normal
101	62.1084	55-79	Normal
102	72.842	72-78	Normal
103	69.2198	62-92	Normal
104	79.1226	69-82	Normal
105	87.962	78-102	Normal
106	64.3349	49-87	Normal
107	71.8118	68-82	Normal
108	94.9405	44-78	Normal
109	94.7411	77-101	Normal
111	70.549	64-82	Normal
112	84.3731	74-91	Normal
113	59.6493	48-87	Bradycardia
114	62.5736	51-82	Normal
115	64.8998	50-84	Normal
116	79.5878	74-86	Normal
117	51.0093	48-66	Bradycardia
118	75.8992	54-91	Normal
119	66.0296	61-84	Normal
121	61.9755	55-83	Normal
122	82.3127	67-97	Normal
123	50.3447	41-65	Bradycardia
124	52.9035	47-64	Bradycardia
200	84.7059	69-111	Normal
201	63.4044	31-61	Normal
202	70.7484	49-69	Normal
203	100.656	63-173	Tachycardia
205	88.5269	80-99	Normal
207	78.8567	57-90	Normal
208	95.8377	91-134	Normal
209	99.8918	82-116	Normal
210	87.1644	63-158	Normal
212	91.3183	63-108	Normal
213	106.372	101-113	Tachycardia
214	75.002	49-92	Normal
215	111.722	81-124	Tachycardia
217	73.141	69-103	Normal
219	71.4463	38-75	Normal
220	68.0567	58-74	Normal
221	80.053	47-110	Normal
222	82.6118	49-84	Normal
223	86.4334	75-94	Normal
228	74.005	54-80	Normal
230	74.9687	63-99	Normal
231	52.2056	49-69	Normal
232	59.5164	24-28	Bradycardia
233	101.819	98-110	Tachycardia
234	91.3515	84-99	Normal

between the suggested algorithm and the MIT-BIH database. Once the ECG signal was deciphered, 46 out of 48 samples in the MIT-BIH database had a correct classification. And only two samples (210 and 217) were incorrectly classified because they contained large noise. This means that the proposed algorithm achieves a classification sensitivity of 95.83%, as shown in Table 2, which is better than [14], which has 92.59%, and only 27 of 48 signals were tested. Specifically, this refers to the fact that some of the signals are biased [14]. Even though an FHE is being used, this method is only employing and assessing addition and excluding multiplication. In addition, the algorithm has detection errors of about 14.8%. While [12] has 90% sensitivity. It is a multiplicative algorithm and reliant on the RSA algorithm's natural computation; this method lacks any sort of security analysis.

The accuracy of the proposed algorithm was 100%. This means it is completely reliable while decrypting (i.e., retrieving the original signal). Based on these results, the FHEAES is able to address the limitations highlighted in other studies like [11], [16], specifically, its inability to provide an internal diagnosis. Furthermore, they have no method of confirming the authenticity of human input and require key exchange, which leads to privacy leaks. Also, they lack the ability to both link to a database and analyze the obtained signals, high storage is required. Also, they are unable to safeguard signals.

At the same time, the use of the suggested FHE ensures that only the user knows the secret key and can perform decryption, keeping data private. The proposed algorithm can conduct internal diagnostics on the encrypted data with excellent accuracy (through the use of the Pan and Tompkins method). Furthermore, all data from the database can be gained, processed, encrypted, and analysed online by the proposed algorithm and then encrypted.

4.2 Keyspace assessment

A brute-force attack is a trial-and-error approach used by attackers for decoding login information and encryption keys to obtain unauthorized access to systems. Keyspace analysis is employed to evaluate the algorithm's strength against brute-force attacks. As can be observed in Table 3, the attacker would require around 5.39475×10^{32} years to break the proposed algorithm. This exceeds the optimum key size required for security against brute-force attacks. In addition, the required cipher-breaking time is enhanced by more than fifty million times in comparison with the conventional AES algorithm and [8, 15, 16]. So, it offers great security against brute-force attacks. The following reasons

Table 2. Sensitivity comparison

Study	Method	Evaluated operation	Security analysis	Number of Tested MIT-BIH signals	Sensitivity	Limitations
[10]	FHE + pre-processing	Addition only	No	Limited	No	Constrained by the problem of greater data, which impairs the effectiveness of the security system. Existence of bias for some of the signals.
[12]	PHE + RSA	Multiplication	No	Limited only 20 signals	90%	It is a multiplicative algorithm that depends on the natural calculations of the RSA algorithm. Existence of bias for some of the signals.
[14]	FHE	Addition only	No	Limited only 27 signals	92.59%	Existence of bias for some of the signals. Has a detection error of about 14.8%.
Proposed algorithm	FHE+ reduced round AES	Addition & Multiplication	Yes	All of the 48 signals	95.83%	No. (Address all of the other studies limitations).

Table 3. Breaking time of the proposed algorithm

Approach	Keyspace	Breaking Time (years)
AES	2^{128}	1.07895×10^{25}
[16]	2^{128}	1.07895×10^{25}
[15]	$2^{4}\times2^{128}\times3\times10$	5.17934×10^{25}
[8]	2^{128}	1.07895×10^{25}
Proposed algorithm	$2^{128}\times5\times10^{7}$	5.39475×10^{32}

contributed to this excellent finding: Firstly, the suggested approach employed three different encryption keys as opposed to only one:

The heart rate (HR) and round number (R) with AES are utilized to generate the secret key for the suggested algorithm, as detailed below:

AES key size = 128 (2^{128} possible probability).

R = 5 possible values (1 to 5) in this experiment 5 rounds are used.

HR = 7 digits containing four decimal places: means 7 positions and each position has 10 possible values (0 to 9): $=10^{7}$ possible probability.

As a second point, this study proposes making the latter two keys (the HR and R) have variable values instead of constant values, which indicates several possible solutions. While the constant value only gives a single probability for a solution. So, the hacker must make a prediction as to whether or not the numbers are variable. Finally, the HR is expanded to four decimal places in accordance with the decimal threshold value, which improves the security level of the proposed algorithm against hackers.

Where Keyspace = (Key1 × Key2 ×..× Keyn), Keyspace for the proposed algorithm = (AES key × R × HR), (R) represent the Round Key, and (HR) is the Heart Rate.

4.3 Correlation analysis

The correlation coefficient metric is used to verify the strength of the algorithm against a statistical attack. An expert cryptanalyst can employ this data to perform a statistical attack and determine the secret key, allowing retrieving the original signal. The values of the correlation coefficient range from -1 to 1, with -1 indicating no correlation and 1 indicating high correlation. While 0 refers to a null correlation. It was computed between the original and encrypted ECG signals. The correlation coefficient is calculated according to [25], all available pairs of columns and rows in the raw and encrypted ECG data were selected. Then, the linear correlation value between every pair was determined. Even though there are only five rounds in the proposed algorithm, the correlation results in Table 4 show that they are good based on database signals. This shows that the proposed algorithm still has a great transformation feature as an encryption

Table 4. Correlation coefficients results

Record No.	Correlation coeffecient	Record No.	Correlation coeffecient
100	0.00965457	201	-0.043519
101	-0.0320642	202	-0.0228434
102	-0.0561683	203	-0.0288637
103	-0.00505078	205	-0.00929113
104	0.0201622	207	-0.00147643
105	-0.0351239	208	-0.0105467
107	-0.0212288	209	0.00020672
106	-0.0196758	210	0.00280202
108	0.0148198	212	-0.0149866
109	-0.0217911	213	-0.0005013
111	0.00137962	214	0.0186444
112	-0.0179313	215	0.0138414
113	-0.0253399	217	-0.0101867
114	-0.0119271	219	0.012838
115	-0.0519887	220	-0.0183225
116	-0.0124454	221	-0.041763
117	0.00980772	222	0.0542994
118	-0.0271551	223	-0.00518063
119	-0.0407918	228	-0.0152499
121	-0.0276936	230	-0.0216763
122	-0.00862294	231	-0.0280106
123	-0.0320509	232	-0.0193257
124	-0.0104753	233	0.00806382
200	-0.00751552	234	0.0362874

algorithm. The correlation coefficients are almost a null (small and close to zero). Thus, there is no correlation between the raw and the encrypted signals. As a result, the suggested technique is secure against statistical attacks. This is due to the usage of three keys that are indicated to have changeable values with decimal places.

4.4 Analysis of encryption/decryption time and throughput

To evaluate the efficiency of the proposed cryptographic algorithm, time and throughput are calculated. Encryption and decryption times are measured five times for each record of the 48 ECG signals in the MIT-BIH Arrhythmia database to rule out the possibility of experimental bias. Then, they were averaged in order to be employed as the encryption and decryption times for every ECG signal. The experiments also calculated the time required to encode and decode files of varying sizes (1 MB, 10 MB, 48 MB, and 96 MB). A signal length of 1000 is equivalent to 96 MB. In this part, the proposed algorithm was tested against AES [13] and FHE (Gentry) [9] as shown in Tables 5 and 6. According to the findings, Fig. 2 shows that the proposed algorithm precedes AES, bee, and FHE (Gentry) in the encryption and decryption processes,

Table 5. Encryption time comparison

ECG Length in (MB)	Proposed System ET (s)	[9] ET (s)	[13] ET (s)
1	0.166	1.025	3.625
10	1.66	10.25	36.25
48	7.968	49.2	174.0
96	15.657	98.40	348.00

Where ET represents Encryption Time.

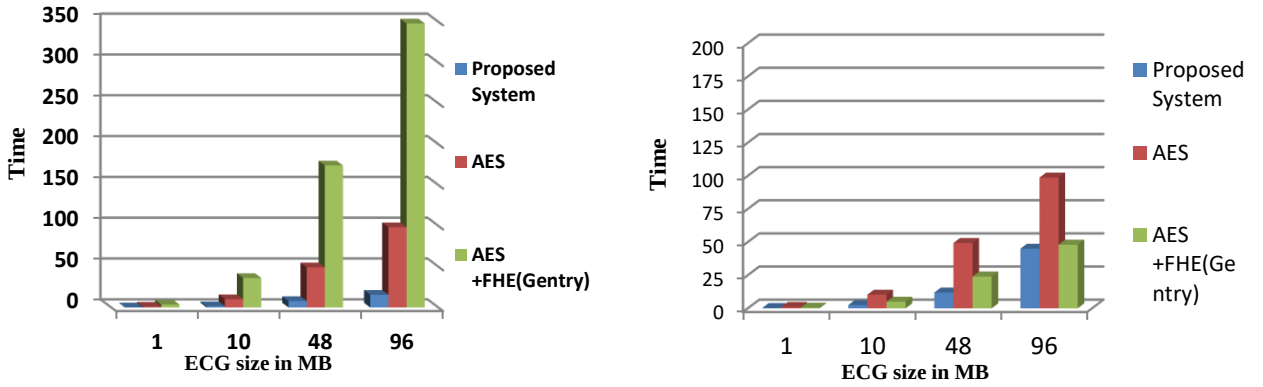
Table 6. Decryption time comparison

ECG Length in (MB)	Proposed System DT (s)	[9] DT (s)	[13] DT (s)
1	0.248	1.028	0.50
10	2.48	10.28	5.00
48	11.904	49.34	24.00
96	44.967	98.70	48.00

Where DT represents Decryption Time.

despite the logical fact that it requires more time to process due to the inclusion of the evaluation process. As mentioned in the related work section, in [9] decoding errors should be avoided by using a recryption process. And the time needed to perform the recryption process is substantial, about 99.9% of the total execution time. This is a huge percentage and cannot be underestimated. So, Gentry's FHE technique is not feasible because of the excessive need for computational capabilities and data storage. Further, as reported in [13], the system performance was reduced because of the additional time needed to compress and send the data. Also, it was found that using the AES algorithm to encode an entire ECG signal was extremely expensive and has more complexity because of the key rules. Therefore, as an alternative to the previous algorithms, the suggested algorithm provides an evaluation procedure, which is faster than a recryption procedure and requires less time and computing power than usual due to its dependence on computationally inexpensive matrix operations. Also, using the reduced round number of cipher iterations, combined with the computationally light matrix operations on which the proposed algorithm relied, explains the time savings achieved by this algorithm. So that data encryption and decryption with the proposed algorithm are faster than with other techniques. Therefore, it's a lightweight algorithm that can be used in a wide variety of IoT contexts.

The throughput of each signal is calculated using the data size in bytes, and then dividing it by the average encryption and decryption time required.



(a) (b)

Figure. 2 Time consumption of encryption and decryption (a) Encryption time comparison and (b) Decryption time comparison

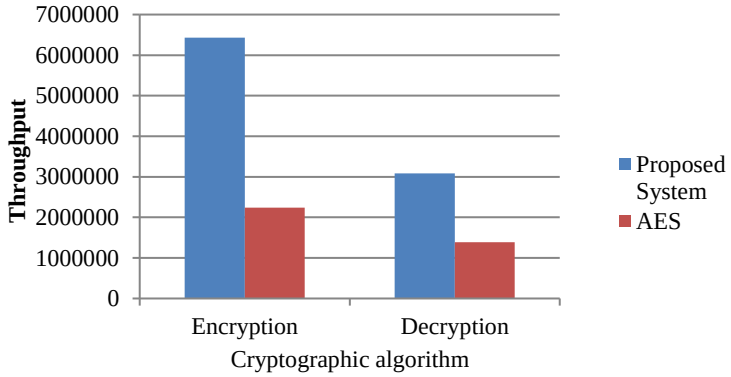


Figure. 3 Throughput of encryption and decryption

And the total throughput was calculated by averaging all of the throughputs. It represents the encryption and decryption speed measurements. The throughput is computed using the following Eq. (7):

$$\text{Throughput} = \frac{\text{plaintext (bytes)}}{\text{encryption/decryption time (second)}} \quad (7)$$

Fig. 3 illustrates the average throughput of the encryption and decryption. It is observed that the encryption throughput of the traditional AES is equal to 34% of the throughput of the proposed algorithm, while it is equal to 44% for the decryption throughput. Thus the proposed algorithm is better than the traditional AES and has a high throughput.

4.5 The effect of sampling frequency on the accuracy

As described in section 3, ECG signals are gained and read online from the MIT-BIH database. After that, the essential Pan and Tompkins algorithm is used in order to collect data from ECG signals, prepare them for encryption, and analyse them later to identify arrhythmia. During this preparation, ECG signals pass through multiple filters low and high pass filter, derivation, squaring, and window integration to get rid of high-frequency noise.

Based on the sampling frequency of the ECG signal, the aforementioned filtering options are modified to better fit the signal's properties [26]. Since the Pan and Tompkins algorithm is a complicated one, it requires additional effort for implementation. Resampling is necessary for each signal that is not sampled at a sampling frequency of exactly 200 Hz. Given that its filters were built to fit a sampling rate of 200 Hz, any frequency above or below that will affect performance [27]. Different sample frequencies (100 Hz, 200 Hz, 300 Hz, and 400 Hz) were used for the ECG signals in the experimental tests of this study. By using various sample frequencies, it is possible to observe the ECG signal to reduce the impact of various forms of noise and to discover the effect of the sampling frequency. It was shown in Fig. 4 (a), that there was no necessity for a low pass filter (in the band pass filter) on P, QRS, or T waves at sampling frequencies specifically at 100 Hz based on the analysis of the amplitude of the signals. Furthermore in Figs. 4 (c) and (d), there was no need for a high

Table 7. The values of encryption / Decryption time and the correlation coefficient in different signal lengths

Metrics	ECG signal length of				
	500	1000	2000	3000	4000
Encryption Time (s)	6.25267	15.5467	36.4433	56.7317	76.248
Decryption Time (s)	18.7875	42.6215	114.248	258.247	456.461
Correlation Coefficient	0.0468842	0.00965457	-0.0422168	0.0178425	0.0203377

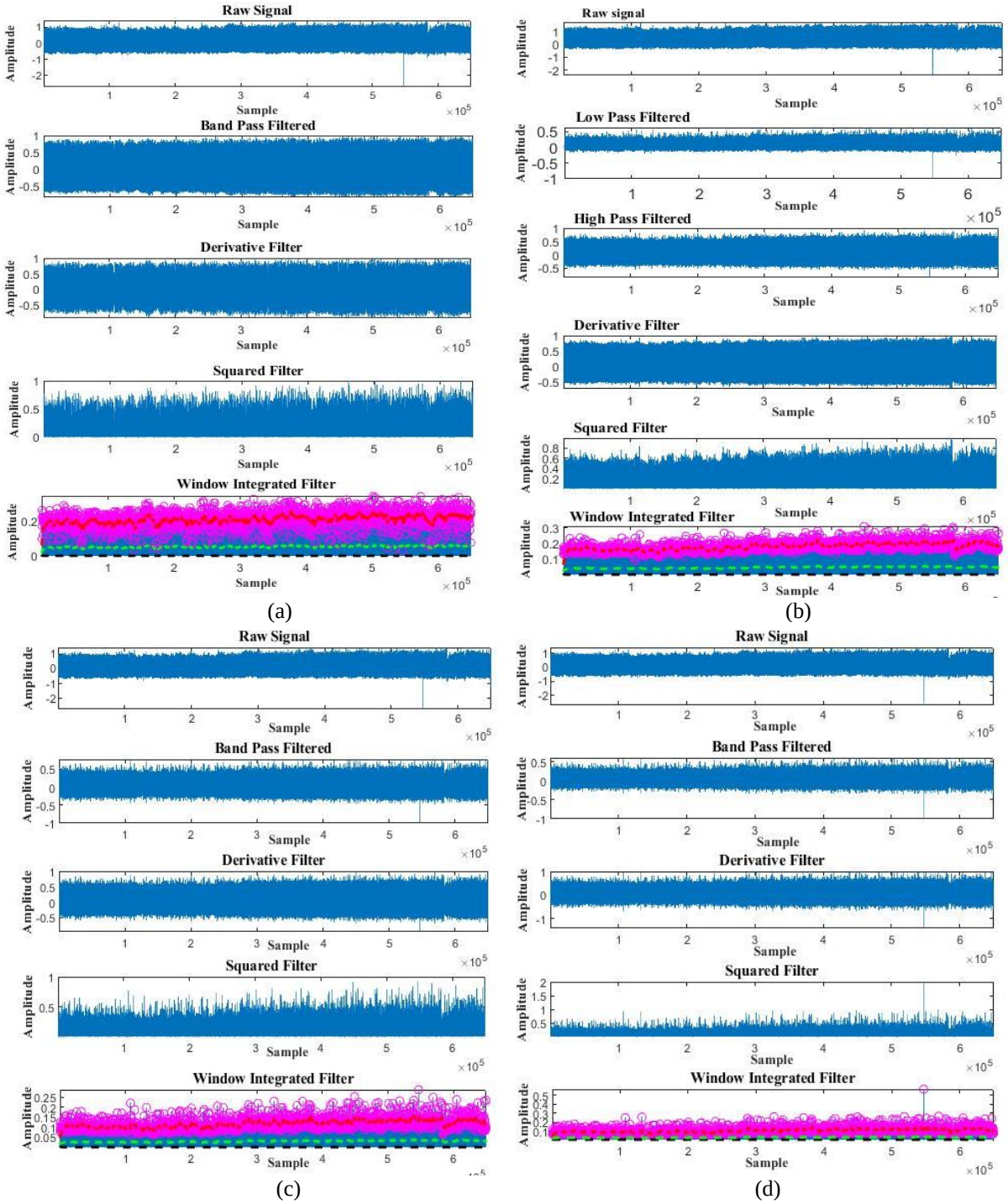


Figure. 4 Preparation filters at different sampling rate: (a) Preparation filters at sampling rate 100 Hz, (b) Preparation filters at sampling rate 200 Hz, (c) Preparation filters at sampling rate 300 Hz, and (d) Preparation filters at sampling rate 400 Hz

pass filter (in the band pass filter) at sampling frequencies more than 200 Hz (i.e. at 300 Hz and 400 Hz). While in Fig. 4 (b), there were both of the low and high pass filters.

Figs. 4 (a), (b), (c), and (d) illustrate that when the sampling frequency was raised, the SNR and amplitudes of the ECG signals decreased. When comparing the low pass filter to the high pass filter, it is observed that the high pass filter produced greater amplitudes of the P, QRS, and T waves from the ECG data.

4.6 The effect of ECG length on the system performance

In this study, two metrics were measured: an encryption/decryption time and the correlation coefficient, to investigate the encryption performance of the proposed system while varying the signal length and to identify the optimal signal length that can be used. In this experiment, the statistical metrics were evaluated using a various signal length: 500, 1000, 2000, 3000, and 4000. Moreover, this experiment is repeated five times for all ECG samples in the MIT_BIH database, then the average of them is calculated and the results are compared against each other to exclude the potential for bias.

Table 7 demonstrates that when the signal length was increased, the value of the encryption time was nearly doubled, while there was a great and rapid growth in the decryption time. Although the results of signal length 2000 have the optimal correlation coefficient value (a negative value), it is observed that it takes a long time for both the encryption and decryption. So, it was excluded from identifying the optimal signal length. Therefore, the signal length of 2000, 3000, and 4000 is unsuitable for real-time systems due to its required time. The signal length of 500 requires a few seconds of encryption and decryption in trade off the sufficient visualization of the signal that was provided.

The correlation coefficient values were all good and in the satisfying range (between 1 and -1) [28]. This indicates that the proposed system is secure with multiple lengths. So, the signal length of 1000 is recommended as an optimal length to be used due to its reasonable results compared to others.

4.7 Histogram analysis

The histogram is a graph of statistical data that shows how the data is distributed. Histogram analysis of the encrypted signal is the simplest approach for demonstrating the efficiency of signal encryption against a statistical attack. Since the aim of any effective signal cryptography is to produce an encrypted signal that resembles randomness, it is preferable to appear as a uniformly distributed histogram. Results in Fig. 5 confirm that this applies to the proposed algorithm, as the histogram of the encrypted signal is uniformly distributed and significantly differs from the original signal. Additionally, it cannot be discriminated statistically from the original signal. This means that the proposed algorithm is reliable against statistical or histogram attacks. The three suggested keys and the linear polynomial function are responsible for this, since they provide resistance and make statistical analysis attacks on encrypted data more difficult.

4.8 Mean square error (MSE)

The MSE is one of the numerical metrics that was employed to evaluate the performance of the proposed system by quantifying the variations between the original and decrypted ECG signals [29]. Smaller MSE values are preferable because they indicate more reliable results. The findings determined that the MSE value is zero for all MIT-BIH database signals examined, since the signal does not miss any information when decoding. This indicates that the proposed algorithm is able to address the noise introduced by the FHE technique since it makes use of computationally lightweight matrix operations. So, the suggested algorithm is efficient because the decrypted ECG signals are identical to the original ones (see Fig. 6), In contrast to the watermarking technique [17], where some information is lost after decryption.

It is shown how to use a FHE and the improved reduced round AES algorithm to encrypt an ECG in real time. The suggested algorithm provides an evaluation process as a simple linear polynomial function with the three proposed keys for better protection and privacy preservation. The solidity of the algorithm against attacks was increased through the use of the three keys, which have variable values with decimal places. Also, the proposed algorithm improves the encryption and decryption times and keeps the correlation coefficient level by reducing the number of AES rounds along with the use of the lightweight matrix to suit real-time applications. Different sampling frequencies were employed in the Pan and Tompkins algorithm to choose the optimum one. Multiple security analyses of the ECG signals, which are encrypted by the proposed algorithm, are performed and then compared to other studies. Further, the proposed method can establish an online connection with the database to gain, encrypt, perform calculations, and diagnose

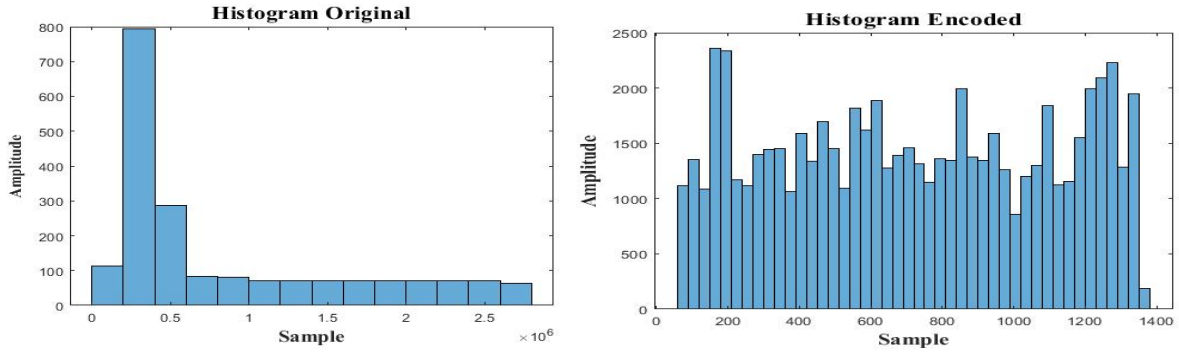


Figure. 5 Histogram analysis of original and encrypted signal of sample 100

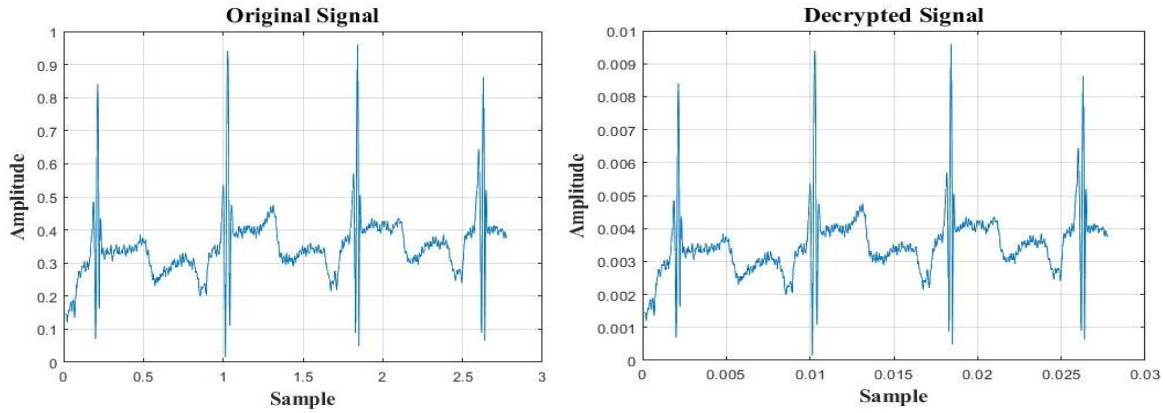


Figure. 6 Original and decrypted ECG of sample 100

the signals. As it is mentioned, the suggested algorithm addresses and exceeds the other research limitations.

The proposed algorithm has been experimentally and statistically proven to be secure against common forms of statistical attack. These results can be attributed to the extraordinary delicacy of the three different keys, alongside the fact that the evaluation procedure has a great degree of randomness.

5. Conclusion

The primary goal of ECG signal encryption is to avoid unauthorized access. This research presents a signal encryption algorithm based on an improved reduced round version of AES with fully homomorphic encryption. Employing homomorphic encryption assisted us get rid of key exchange, thus maintaining privacy. In spite of using fewer rounds of encryption with the additional homomorphic layer to AES, results reveal that the improvement provides more security than AES, enabling safe signal transfer. And the nonlinear feature of the algorithm is preserved. Also, there is a remarkable reduction in the encryption and decryption times, so it's useful in real-time IoT contexts. Also, both decryption accuracy and arrhythmia classification sensitivity have significantly improved to be 100% and 95.83%, respectively. Further, the proposed algorithm is reliable against many attacks because it needs a long time to be broken. Compared to the original AES technique, this algorithm obviously increases the time needed to crack the ciphertext by more than 50,000,000 times. The proposed algorithm can be tested by encrypting other signals and seeing how changing the key affects the algorithm's efficiency.

Conflicts of interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Author contributions

Conceptualization, AAA, MMM, and SKG; methodology, AAA and SKG; software, AAA; validation, SKG, and MMM; formal analysis, AAA, MMM, and SKG; investigation, MMM, and SKG; resources, AAA; data curation, AAA; writing—original draft preparation, AAA; writing—review and editing, MMM and SKG; visualization, AAA; supervision, MMM and SKG; project administration, AAA, MMM and SKG; funding acquisition, AAA.

References

[1] C. Marcolla, V. Sucasas, M. Manzano, R.

Bassoli, F. H. P. Fitzek, and N. Aaraj, "Survey on Fully Homomorphic Encryption, Theory, and Applications", In: *Proc. of the IEEE*, Vol. 110, No. 10, pp. 1572-1609, 2022. doi: 10.1109/JPROC.2022.3205665.

[2] F. T. A. Hussien, A. M. S. Rahma, H. B. A. Wahab, and M. Arif, "A Secure Environment Using a New Lightweight AES Encryption Algorithm for E-Commerce Websites", *Journal of Security and Communication Network*, Vol. 2021, pp. 1-15, 2021, doi: 10.1155/2021/9961172.

[3] R. Salavi, M. M. Math, and U. P. Kulkarni, "A Comprehensive Survey of Fully Homomorphic Encryption from Its Theory to Applications", *Cyber Security and Digital Forensics*, pp. 73-90, 2022, doi: 10.1002/9781119795667.ch4.

[4] M. N. Imtiaz and N. Khan, "Pan-Tompkins++: A Robust Approach to Detect R-peaks in ECG Signals", In: *Proc. of the 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Las Vegas, NV, USA, pp. 2905-2912, 2022, doi: 10.1109/BIBM55620.2022.9995552.

[5] S. J. Mohammed and D. Basheer, "From Cloud Computing Security towards Homomorphic Encryption: A Comprehensive Review", *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, Vol. 19, No. 4, pp. 1152-1161, 2021, doi: 10.12928/TELKOMNIKA.v19i4.16875.

[6] G. K. Mahato and S. K. Chakraborty, "A Comparative Review on Homomorphic Encryption for Cloud Security", *IETE Journal of Research*, 2021, doi: 10.1080/03772063.2021.1965918.

[7] A. Viand, P. Jattke, and A. Hithnawi, "SoK: Fully Homomorphic Encryption Compilers", In: *Proc. of 2021 IEEE Symposium Conf. on Security and Privacy (SP), IEEE Computer Society*, pp. 1092-1108, 2021, doi: 10.1109/SP40001.2021.00068.

[8] G. Sravya and P. R. Ranjani, "AES Algorithm for Secure Electro Cardiogram Signal Transmission", *GIS SCIENCE JOURNAL*, Vol. 9, No. 7, pp. 2629-2638, 2022.

[9] Ö. Kocabas and T. Soyata, "Towards privacy-preserving medical cloud computing using homomorphic encryption", *Virtual and Mobile Healthcare: Breakthroughs in Research and Practice, IGI Global*, pp. 93-125, 2020, doi: 10.4018/978-1-5225-9863-3.ch005.

[10] A. Bahattarai and D. Peng, "An Integrated Secure Efficient Computing Architecture for Embedded and Remote ECG Diagnosis", *SN COMPUT. SCI*, Vol. 4, No. 1, 2022, doi: 10.1007/s42979-022-01465-7.

[11] Z. M. Zubaer, K. Thappa, and S. Yang, "Improving R Peak Detection in ECG Signal Using Dynamic Mode Selected Energy and Adaptive Window Sizing Algorithm with Decision Tree Algorithm", *Sensors (Basel, Switzerland)*, Vol. 21, No. 19, pp. 1-17, 2021, doi: 10.3390/s21196682.

[12] N. Naeem, F. Khan, T. Yaqoob, and S. Tahir, "Privacy-Preserving Computing via Homomorphic Encryption: Performance, Security, and Application Analysis", *Handbook of Research on Cybersecurity Issues and Challenges for Business and FinTech Applications, IGI Global*, USA, pp. 288-313, 2023, doi: 10.4018/978-1-6684-5284-4.ch015.

[13] M. E. Hameed, M. M. Ibrahim, N. A. Manap, and A. A. Mohammed, "An enhanced lossless compression with cryptography hybrid mechanism for ECG biomedical signal monitoring", *International Journal of Electrical and Computer Engineering*, Vol. 10, No. 3, pp. 3235~3243, 2020, doi: 10.11591/ijece.v10i3.pp3235-3243.

[14] M. U. Shaikh, W. A. W. Adnan, and S. A. Ahmad, "Sensitivity and Positive Prediction of Secured Electrocardiograph (ECG) Transmission using Fully Homomorphic Encryption Technique (FHE)", In: *Proc. of 2020 IEEE-EMBS Conf. on Biomedical Engineering and Sciences (IECBES)*, pp. 292-297, 2021, doi: 10.1109/IECBES48179.2021.9398792.

[15] J. K. Madhloom, M. K. A. Ghani, and M. R. Baharon, "Ecg encryption enhancement technique with multiple layers of AES and DNA computing", *Intelligent Automation & Soft Computing*, Vol. 28, No. 2, pp. 493–512, 2021, doi: 10.32604/iasc.2021.015129.

[16] B. K. Abdullah, R. D. Mahdi, T. I. Mohamed, R. A. Jaleel, M. A. Salih, and M. M. A. Zahra, "A novel secure artificial bee colony with advanced encryption standard technique for biomedical signal processing", *Periodicals of Engineering and Natural Sciences*, Vol. 10, No. 1, pp.288-294, 2022, doi:10.21533/pen.v10i1.2610.

[17] A. Khaldi, M. R. Kafi, and B. Meghni, "Electrocardiogram signal security by digital watermarking", *Journal of Ambient Intelligence and Humanized Computing*, pp 1-13, 2022, doi: 10.1007/s12652-022-04101-7.

[18] https://archive.physionet.org/physiobank/database/mitdb/ (accessed 19/3/2023).

[19] V. R. Vimal, P. Anandan, and N. Kumaratharan, "Heart disease diagnosis using electrocardiography (ecg) signals", *Journal of Intelligent Automation & Soft Computing*, Vol. 32, No. 1, pp. 31–43, 2022, soi: 10.32604/iasc.2022.017622.

[20] A. A. Ahmed, M. M. Madboly, and S. K. Guirguis, "Securing Data Transmission and Privacy Preserving Using Fully Homomorphic Encryption", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 277-289, 2023, doi: 10.22266/ijies2023.0228.25.

[21] A. K. Singh and S. Krishnan, "ECG signal feature extraction trends in methods and applications", *BioMedical Engineering OnLine*, Vol. 22, No. 1, pp. 1-36, 2023, doi: 10.1186/s12938-023-01075-1

[22] A. Bujnowski, K. Osiński, P. Przystup, and J. Wtorek, "Non-Contact Monitoring of ECG in the Home Environment-Selecting Optimal Electrode Configuration", *Sensors*, Vol. 22, No. 23, 2022, doi: 10.3390/s22239475.

[23] https://byjus.com/maths/polynomial-functions/ (Accessed 19/3/2023).

[24] Y. Zhang, A. Gu, Z. Xiao, Y. Xing, C. Yang, J. Li, and C. Liu, "Wearable Fetal ECG Monitoring System from Abdominal Electrocardiography Recording", *Biosensors*, Vol. 12, No. 7, 2022, doi: 10.3390/bios12070475.

[25] B. Adithya and G. Santhi, "A DNA Sequencing Medical Image Encryption System (DMIES) Using Chaos Map and Knight's Travel Map", *International Journal of Reliable and Quality E-Healthcare*, Vol. 11, No. 4, pp. 1-22, 2022, doi: 10.4018/IJRQEH.308803.

[26] S. Shimauchi, K. Eguchi, R. Aoki, M. Fukui, and N. Harada, "R-R Interval Estimation for Wearable Electrocardiogram Based on Single Complex Wavelet Filtering and Morphology-Based Peak Selection", *IEEE Access*, Vol. 9, pp. 60802-60827, 2021, doi: 10.1109/ACCESS.2021.3070604.

[27] A. Wihantara, I. D. G. H. Wisana, A. Pudji, S. Luthfiyah, and V. A. Athavale, "QRS Complex Detection On Heart Rate Variability Reading Using Discrete Wavelet Transform", *Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics*, Vol. 4, No. 4, pp. 153-159, 2022, doi: 10.35882/ijahst.v2i5.236.

[28] C. Che, P. Zhang, M. Zhu, Y. Qu, and B. Jin, "Constrained transformer network for ECG signal processing and arrhythmia classification", *BMC Med Inform Decis Mak*, Vol. 21, No. 1, pp. 1-13, 2021, doi: 10.1186/s12911-021-01546-2.

[29] H. Wen, Z. Chen, J. Zheng, Y. Huang, S. Li, L. Ma, Y. Lin, Z. Liu, R. Li, L. Liu, W. Lin, J. Yang, C. Zhang, and H. Yang, "Design and Embedded Implementation of Secure Image Encryption Scheme Using DWT and 2D-LASM", Vol. 24, No. 10, pp. 1-19, 2022, doi: 10.3390/e24101332.