IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 March 2024, accepted 10 May 2024, date of publication 20 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3402691

RESEARCH ARTICLE

Data Privacy Technique for the Data Transmitted in Wireless Body Area Network

DIVYA S.[1], PREMA K. V.[2], (Member, IEEE), AND BALACHANDRA MUNIYAL[1], (Member, IEEE)

[1]Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India
[2]Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India

Corresponding authors: Prema K. V. (prema.kv@manipal.edu) and Balachandra Muniyal (bala.chandra@manipal.edu)

ABSTRACT Within Wireless Body Area Networks (WBANs), wearable devices (WDs) capture vital parameters and process this information. The processed data is then conveyed to a wearable device controller (WDC) before being securely archived within a medical server. Data privacy preservation becomes imperative, preventing unauthorized access and potential compromise. The focal point of the presented research centers on safeguarding data privacy. This objective is accomplished through a two-fold approach: firstly, by encrypting the data within the wearable device controller before storage in the server; secondly, by regulating access to the data for different entities seeking to retrieve it. The innovation of this work lies in its emphasis on ensuring privacy within the wearable device controller itself. It is shown that privacy is obtained using minimum computations. Also, security analysis is performed by identifying various threats and how it is addressed. The encryption process, key generation, and key refreshing are executed employing the MATLAB programming environment.

INDEX TERMS Data privacy, MATLAB, modified blowfish, nonce generation, wireless body area network.

I. INTRODUCTION

In the context of healthcare applications, there is a necessity for wireless body area networks (WBANs) to identify and mitigate potential health threats. This capability is achieved through remote monitoring of patient's health conditions. However, the security and privacy of patient data pose considerable challenges that demand particular focus. These concerns are of utmost importance, as they significantly impact patients' medical diagnosis and treatment, as discussed in [1] and [2].

Illustrated in Figure 1 is a common WBAN configuration applied in healthcare scenarios. In this setup, numerous wearable devices and a singular wearable device controller are affixed to the human body. These wearable devices(WD) gather specific physiological metrics, which are then relayed to a medical server. The transmission process is facilitated through an intermediary device known as a wearable device controller(WDC) [3]. Data can be further accessed by various entities to treat the patients.

Since the patient's health data is sent from the WDC to the medical server using a wireless medium, ensuring the security of this data is of utmost importance. If the transmitted data becomes corrupted, it could endanger the patient's life.

Medical decisions are based on the received data [4]. Therefore, security is required at several levels:

- Data transmission from the WD to the WDC.
- Data transmission from the WDC to the medical server.
- The data stored in medical servers, wearable device controllers, and wearable devices.

Security and privacy can be provided by using cryptographic algorithms. The algorithm applied should use fewer resources. Also, it should have minimum computations as the devices are resource-constrained [4].

Therefore, the research aims to securely send the sensed data from wearable devices to the server and apply access

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

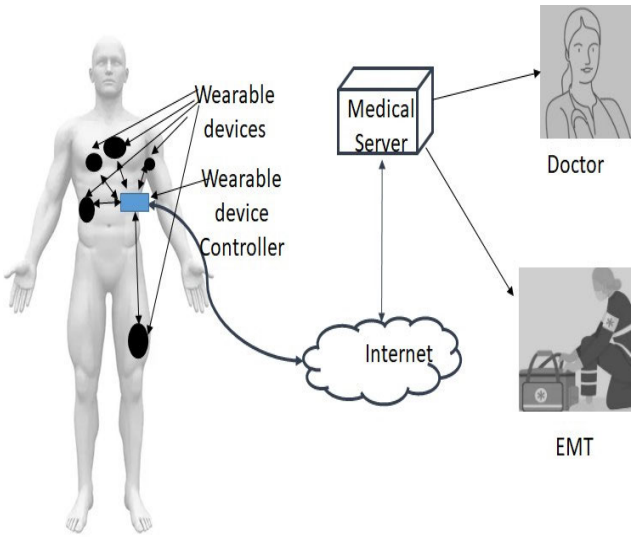


FIGURE 1. WBAN data communication.

control mechanisms to various entities using an encryption algorithm that performs less computation and utilizes less storage space. The key used for encryption has proven unique and random in the work. Also, the work ensures that only a particular entity can access a particular field in the data by encrypting the fields with different entity keys. Various threats mitigated are discussed in the paper.

Preliminaries required for the work are discussed in section II. A literature survey performed for achieving the privacy of the data transmitted and stored in the medical server, performing authentication, and for selecting the encryption algorithm is discussed in section III. In section IV, the methodology to achieve the objective is indicated. Section V indicates the results and analysis of the work done. Section VI indicates the conclusion and discussion of the work.

II. PRELIMINARIES

This section comprises a discussion of data encryption algorithms.

A. BLOWFISH ALGORITHM

The Blowfish algorithm is a symmetric block cipher employing 16 encryption rounds [5], [6]. This cryptographic process involves the utilization of two arrays, namely the P-array and the S-boxes. The P-array encompasses eighteen sub-keys, each consisting of 32 bits labelled as P1 through P18. Complementing this, the S-boxes comprise four distinct units, each composed of 256 entries, each with a bit length of 32. The Pseudocode in Algorithm 1 defines the encryption process:

The keys are used in reverse order for decryption. The Pseudocode in Algorithm 2 defines the decryption process:

B. MODIFIED BLOWFISH ALGORITHM

In the context of the paper [7], an adapted rendition of the Blowfish algorithm aimed at diminishing the number of encryption rounds is introduced. Specifically, if the result of a bitwise XOR operation on all bits of the encryption key equals 0, the encryption process will make use of sub-keys 1 through 10 from the P-array. Conversely, should the result be 1, the algorithm will opt for sub-keys 18 through 9 from the P-array. This tailored approach results in a reduced total of 8 encryption rounds being executed.

The Pseudocode in Algorithm 3 defines the encryption process:

The Pseudocode in Algorithm 4 defines the decryption process:

Algorithm 1 Blowfish Encryption

for j= 1 to 16 **do**
  $RE_j = LE_{j-1} \oplus P_j$;
  $LE_j = F[RE_j]\ RE_{j-1}$;
end for
$LE_{17} = RE_{16} \oplus P_{18}$;
$RE_{17} = LE_{16} \oplus P_{17}$;

Algorithm 2 Blowfish Decryption

for j= 1 to 16 **do**
  $RDj = LDj - 1 \oplus P_{19-j}$;
  $LD_j = F[RD_j]\ RD_{j-1}$;
end for
$LE_{17} = RE_{16} \oplus P_1$;
$RE_{17} = LE_{16} \oplus P_2$;

Algorithm 3 Modified Blowfish Encryption

if The XOR of individual bits of the key is Zero, **then**
  for$k$= 1 to 8 **do**
    $RE_k = LE_{k-1} \oplus P_k$;
    $LE_k = F[RE_k\ ] \oplus RE_{k-1}$;
  end for
  $LE_9 = RE_8 \oplus P_{10}$;
  $RE_9 = LE_8 \oplus P_9$;
else
  for k= 18 to 11 **do**
    $RE_{19-k} = LE_{19-k-1} \oplus P_k$;
    $LE_{19-k} = F[RE_{19-k}] \oplus RE_{19-k-1}$;
  end for
  $LE_9 = RE_8 \oplus P_9$;
  $RE_9 = LE_8 \oplus P_{10}$;
end if

III. LITERATURE SURVEY

The following literature survey was conducted to achieve privacy, generate nonce for performing the authenticity of the devices and compare various cryptographic algorithms. Samaher Al-Janab et al. [8] in the paper reviewed WBAN communication architecture, security and privacy requirements, security threats, and the challenges involved in WBANs. As part of open areas for research, it has been

Algorithm 4 Modified Blowfish Decryption

if The XOR of individual bits of the key is Zero, **then**
  for k= 9 to 16 **do**
    $RD_{k-8} = LD_{k-8-1} \oplus P_k$;
    $LD_{k-8} = F[RD_{k-8}] \oplus RD_{k-8-1}$;
  end for
  $LD_9 = RD_8 \oplus P_{18}$;
  $RD_9 = LD_8 \oplus P_{17}$;
else
  for k= 10 to 3 **do**
    $RD_{11-k} = LD_{11-k-1} \oplus P_k$;
    $LD_{11-k} = F[RD_{11-k}] \oplus RD_{11-k-1}$;
  end for
  $LD_9 = RD_8 \oplus P_1$;
  $RD_9 = LD_8 \oplus P_2$;
end if

indicated that security and privacy are better platforms for healthcare applications.

Milad et al. [9] conducted a performance evaluation by comparing the speed of the Blowfish and Skipjack algorithms in file encryption. The study concluded that Blowfish outperformed Skipjack across various file sizes and contents.

In the research by Mota et al. discussed in [10], an in-depth comparative analysis of encryption algorithms was conducted. Parameters such as encryption and decryption times, memory usage, latency, power consumption, jitter, and security level were considered. The study's findings indicated that among 3DES, DES, and AES, the Blowfish algorithm demonstrated superior performance, making it the preferred choice.

Patil et al., as cited in [11], compared encryption algorithms, including DES, 3DES, Blowfish, and AES. Their study focused on memory usage and implementation time. Results highlighted Blowfish's efficiency in terms of memory consumption and encryption speed, along with its resilience against guessing attacks. The conclusion was that Blowfish emerged as the most suitable algorithm, especially when constrained by time and memory.

Patel and Kamboj, as outlined in [12], introduced an enhancement scheme for the Blowfish algorithm. They achieved this by reducing the number of encryption rounds using a special key, bolstering security against brute-force attacks. Additionally, the scheme was shown to reduce encryption and decryption times.

Quilala et al. work, referenced in [13], involved the development of a modified Blowfish encryption variant utilizing 128-bit data and key lengths, with eight encryption rounds. Performance evaluation based on encryption time and avalanche effect indicated that the modified algorithm was slower due to its larger block size. However, the algorithm was also more secure, exhibited a higher avalanche effect, and showed faster encryption than the Twofish algorithm.

In the paper, Divya et al. [7] proved that the proposed method reduces the encryption time by 50% compared with the Blowfish algorithm. The calculated value for an avalanche of the Modified Blowfish is 48.5%, and the Blowfish is 47.14%. As the avalanche percentage increases, the security achieved is better.

Ding and Tan as reported in [14], conducted a comparison of random number generators, observing their influence on Particle Swarm Optimization (PSO) performance. The study identified the Mersenne Twister algorithm as the most efficient random number generator.

Bhattacharjee et al. work, detailed in [15], categorized pseudo-random number generators into linear feedback shift registers, linear congruential generators, and cellular automata. Empirical testing encompassing statistical and graphical assessments was performed on selected generators, with the SFMT-64-bit generator emerging as the top performer.

Wu et al., as described in [16], presented a secure authentication scheme for wireless body area networks. The scheme featured three stages—setup, registration, and authentication —wherein a nonce and timestamp were incorporated into transmitted messages. The study demonstrated a 31.58% reduction in computation cost.

Chunka and Banerjee [17], proposed a scheme that consists of four phases out of which one of the phases is authentication and Key Agreement Phase. The nonce generated is random and is used for authentication purposes and eliminates replay attacks and it indicates the freshness of the message transmitted.

Hussain et al. [18], proposed a scheme for mutual authentication in WBANs. It consists of three phases: Initialization, Registration, and Authentication. In the authentication phase, a random nonce is generated for authentication purposes and also to eliminate replay attacks.

Chatterjee, mentioned in [19], proposed a nonce-based authentication protocol rooted in public-key cryptography. The nonce was employed to thwart replay attacks and indicate message freshness.

Al-Janabi et al. [20] proposed a key management scheme where keys are refreshed by randomly selecting values from already existing biometric measurements that have been measured by sensor nodes. The re-keying technique is used where the new key does not depend on the old key value.

Ali and Khan [21] proposed a key management scheme for WBANs. One-way hash function is applied to the previous key to obtain the new key. Here key updating is performed as the new key is obtained from the old key. Key refreshing maintains forward secrecy in the work.

Irum et al. [22] proposed a hybrid key management scheme for intra-WBAN and inter-WBAN communications. Key refreshing is performed by applying a keyed hash function to the feature key. Key updating is performed in the work.

Chatterjee et al. [23] proposed an access control technique with an access list comprising a user identity, a user access privilege mask, and an access group ID for each user. Each access group can access data according to the privileges given to that group. A user access privilege mask is a binary number

TABLE 1. Symbols and their explanation.

Symbols	Explanation
WBAN	Wireless Body Area Network
WD	Wearable device
WDC	Wearable device controller
K_{ECG}	Key generated from ECG
K_{si}	Session key for Device i
K_{server}	Server key
LS byte	Least significant byte
MS byte	Most significant byte
Di	Device i data

where each bit represents specific information or services that an authenticated user can access. It allowed the user to authenticate and access data inside a WBAN sensor node under certain privileges.

Soni and Singh [24] proposed a data communication scheme that can withstand various security and privacy attacks while consuming fewer computational resources. In the work, nonce is not used. Only the time stamp is used to authenticate the messages.

The literature survey shows that Blowfish and SFMT are suitable algorithms for the proposed work. A random number is used as nonce, and separate timestamps are used to authenticate the devices in all the works mentioned, increasing overhead. Also, in the works mentioned, more messages are communicated between devices to provide privacy. As the devices are resource-constrained, there is a requirement to reduce the overhead and the number of messages exchanged between any two devices in the wireless body area network.

IV. METHODOLOGY

The block diagram of the proposed work is indicated in Figure 2. In the proposed work, the data from the wearable device is encrypted using a modified encryption algorithm, and the session key of the device is transmitted to the wearable device controller. The data received from all the wearable devices are collected and combined at the wearable device controller, encrypted, and appended with different entity data. It is then encrypted using the server key and sent to the medical server. Different fields in the data are encrypted using different keys to give access only to the required entity. At the medical server, the data is decrypted using the server key. Whenever any entity is required to access the data, it decrypts it using the key. The different symbols used in the work are explained in Table 1.

The following assumptions are made in the proposed work:

1) One wearable device controller is attached to multiple wearable devices.
2) The symmetric [4], [25], [26], [27] keys are used for the encryption of the data in the proposed work.
3) The size of the two symmetric keys K_{ECG} and K_{si} used in the work is 128-bit each:
   a) K_{ECG} is the key generated using an ECG signal and is available only at the wearable device controller device.
   b) K_{si} is the session key for a particular wearable device i. Each wearable device will have a different session key that varies from one device to another. Due to this, if one of the devices compromises with the attacker, other device keys are not known to the attacker. The key K_{si} is present at the individual devices and the wearable device controller and will be used by the devices to encrypt the data sent to the wearable device controller.
4) A wearable device controller ID is generated using an ECG signal. It is a 128-bit value and is present at the controller.
5) The entities involved are the patient (wearable device, wearable device controller), doctor, emergency medical technician (EMT), and the medical server (administrator). The doctor should be able to access all the patient's data. The EMT should be able to access only the emergency data, not all recorded data from previous dates. The EMT can access the relevant medical data only if emergencies occur. The administrator should not be able to access any stored patient-related information.

The generation of the keys K_{ECG}, K_{si}, and Wearable device controller ID is outside the scope of this paper and is indicated in the paper [28].

In this proposed work, the data to be accessed by different entities (patient, EMT, doctor) are encrypted by the respective entity keys at the wearable device controller. The data related to all the entities is then combined, encrypted, and transmitted to the medical server. Since the data received is encrypted twice, the server does not know the content. Only the person with the key for decryption will be able to view the actual data.

Table 2 compares the blowfish encryption algorithm with the modified blowfish algorithm. It is shown that the number of rounds used is less in [7], and the amount of computation also reduces. Therefore, the proposed work uses a modified blowfish algorithm for encryption.

The proposed work includes the following steps:

1) Generation of the subkeys (initialization of the P array and S boxes using the key used in the work) for the modified Blowfish algorithm [7]: The subkeys are generated by applying the Blowfish algorithm similar to the method used in the paper [7]. To obtain all the p-array and S-array entries based on a key, a total of 521 iterations are performed.
2) Transmission of the data sensed by the wearable device to the wearable device controller.
   a) The Nonce Generation:
   The nonce is included with the data to ensure the uniqueness of the message transmitted between any two devices and to prevent a replay attack. The nonce is also used to ensure the message is fresh and sent by an authenticated device.

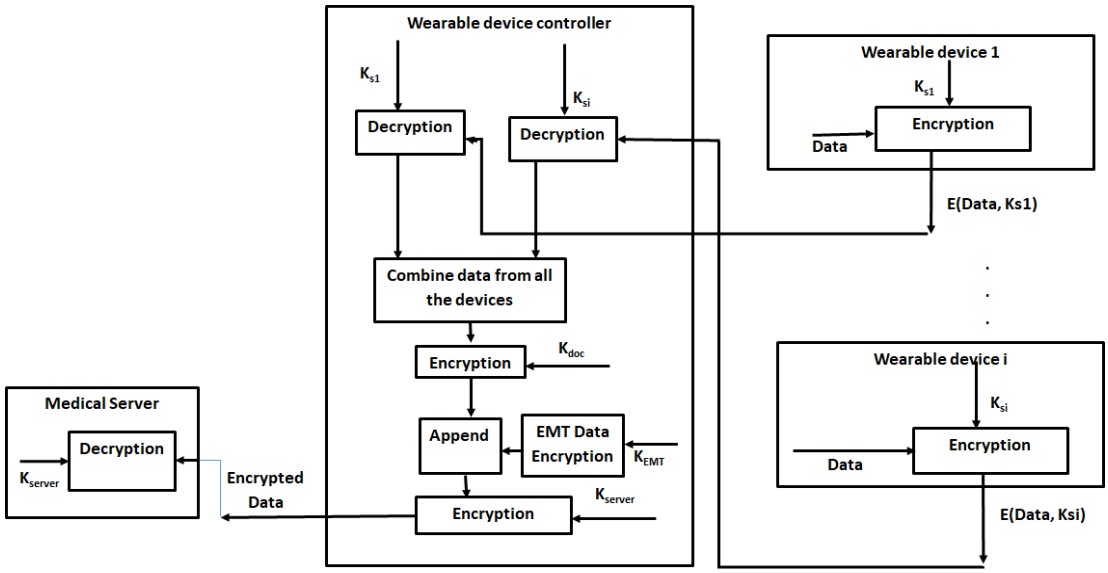


FIGURE 2. Block diagram of proposed work.

TABLE 2. Comparison of the encryption techniques.

Work	Memory in Bytes	iterations for key generation	Bytes used for encryption	Rounds used	Remarks
Blowfish	4,168	521	4,168	16	All the rounds are used for encryption
Quilala et al. [13]	2,128	266	2,128	8	All the rounds are used for encryption
Modified Blowfish [7]	4,168	521	4,136	8	Out of 16 rounds random 8 rounds are used for encryption

In the work, to verify the authenticity of the wearable device sending the data, a nonce is generated and transmitted along with the data transmitted to the wearable device controller. The following steps are involved in the nonce generation.

- The nonce is generated from the key K_{si}, date, and time.
- Date and time are represented as a 64-bit value.
- As the key K_{si} is a 128-bit value, the least significant 64-bits or most significant 64-bits from the key K_{si} are chosen using a random number.
- 64-bits from the key K_{si} are then XORed with date and time to generate a 64-bit nonce.

$$\text{nonce (64 bit)} = K_{si}(\text{LS 64 bits or MS 64 bits}) \oplus (\text{date and time}) \quad (1)$$

b) Seven parameters or samples from the wearable device are collected before transmitting to the wearable device controller.

c) The wearable device number is appended with D_i (combined samples), part of the key K_{si}, and the generated nonce.

d) The part of the key K_{si} used for nonce generation is appended along with the nonce. The Key K_{si} is appended to verify the nonce transmitted at the receiver end. Only 64 bits of the key K_{si} are appended to reduce overhead.

e) The data is then encrypted using the key K_{si} of the device i.

$$\begin{aligned}&\text{Encrypted Data at device i}\\ &= E\{[\text{Device number}\\ &\quad ||Di||K_{si}(\text{LS 64-bit or MS 64-bit})\\ &\quad \|\text{nonce}], K_{si}\end{aligned} \quad (2)$$

f) Data transmitted to the wearable device controller is obtained by appending the encrypted value of K_{si}'s LS 64-bit if the nonce is generated using MS 64-bit, using the wearable device controller ID to the encrypted data. Appending of the key K_{si} is done to identify the device that is transmitting

the data. Since other devices should not obtain this identity, it is encrypted, which improves the privacy of the identity. Data transmitted is given as follows:

$$\begin{aligned}&\text{Transmitted data by the device i} = E\{K_{\text{si}}\\&\text{(LS 64-bit or MS 64-bit, } WDCID\}\\&||E\{[\text{Device number}||Di||\\&K_{\text{si}}(\text{LS 64-bit or MS 64-bit})||nonce], K_{\text{si}}\}\end{aligned} \tag{3}$$

3) Reception of the data by the wearable device controller.
   a) The wearable device controller identifies the wearable device by decrypting the first 64 bits from the received data using the WDCID.
   b) The data received is then decrypted by the wearable device controller using the respective device key, K_{si}.

$$\begin{aligned}&\text{Decrypt Data of device i} = D\{E\{[Di||K_{\text{si}}\\&(LS64\text{-bit or MS 64-bit})||\\&nonce], K_{\text{si}}\}, K_{\text{si}}\}\end{aligned} \tag{4}$$

   c) The wearable device controller collects and decrypts all the data from the wearable devices connected to the wearable device controller.
4) Encryption of the combined data at the wearable device controller.
   a) The data received from all the wearable devices are combined with their device number as single data.
   b) It is appended with wearable device controller ID and then encrypted using a modified blowfish algorithm and the doctor key K_{doc} (computed using the method described in the paper referenced as [28]) as the key.

$$\begin{aligned}&\text{Encrypted Data} = E\{WDCID||[(D1||D2||\\&\quad\ldots Dn)], K_{\text{doc}}\}\end{aligned} \tag{5}$$

   where D1, D2, and Dn are wearable device data.
5) Attaching EMT-related data to the above-mentioned data.
   - Encrypting of EMT values:
     Patient ID (64-bit value) is attached to the EMT data. To identify the EMT values, the data from the wearable devices is compared with the normal values of the sensed physiological signals. If there is variation, the total number of varied data and the device number are mentioned in the EMT data. Patient ID and EMT values are then encrypted using EMT key K_{EMT} (the key is computed using the method described in the paper referenced as [28]).

$$\begin{aligned}&\text{Encrypted EMT Data}\\&= E\{\text{Patient ID}||[(\text{Device1 number}\\&||\text{number of Device1 varied data}||\\&\text{Device2 number}||\text{number of Device2 varied}\\&\text{data}\ldots||\text{Devicen}\\&\text{number}||\text{number of Device n varied data}), K_{\text{EMT}}\}\end{aligned} \tag{6}$$

   - Appending of Flag field based on EMT-related data The Flag consists of 16 bits and is set to all ones if there are more than four instances of emergency values within the data. The wearable device controller appends the Flag's value by examining the emergency values present.

$$\text{EMT} = \text{Flag}||\text{Encrypted EMT data} \tag{7}$$

6) Encryption of the data before transmitting it to the medical server.
   The following needs to be performed for transmission of the data from the wearable device controller to the medical server:
   a) Key K_{server} generation:
      The study employs the Mersenne Twister random number generator [27], [29], chosen for its demonstrated efficiency, as highlighted in the referenced paper [30]. In Figure 3 [28], the utilization of the Mersenne Twister for Pseudo-random number generation is depicted, employing recurring and tempering techniques. The recurrence process involves a Linear Feedback Shift Register (LFSR) to produce individual bits of the state, with 624 elements constituting the shift register, each element being of size 32-bits.
      SFMT (Single Instruction Multiple Data-Oriented Fast Mersenne Twister) represents the evolved 128-bit iteration of the Mersenne Twister algorithm. It facilitates concurrent processing through multi-stage pipelines and Single Instruction Multiple Data (SIMD) operations. This version outpaces the original Mersenne Twister by a factor of two, accomplished by leveraging a block generation function and SIMD operations to populate a 32-bit integer array in one operation efficiently. SFMT extends its support to a range of periods from $2^{607} - 1$ to $2^{216091} - 1$, exhibiting superior equidistributional properties across dimensions compared to the conventional Mersenne Twister. By utilizing the final 8 bits of the K_{ECG} key as the initiating seed for the SMFT random number generator, the K_{server} key is formed. This K_{server} key encrypts the transmitted data from the wearable device controller to the medical server. As the derivation of the K_{server} key involves K_{ECG}, distinct K_{server} keys are generated for each wearable device controller to facilitate communication. These unique K_{server} keys are provided to the wearable device controller and the medical server during the initial setup.

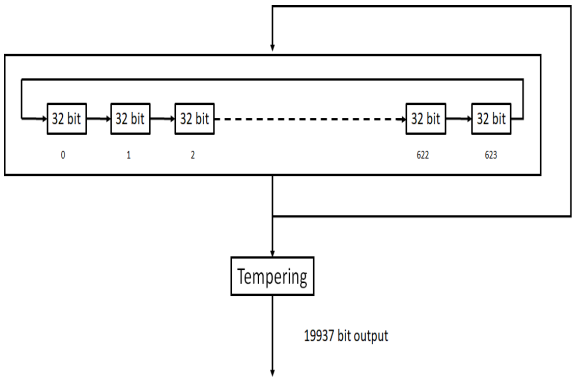


FIGURE 3. Mersenne Twister.

b) The encrypted data is appended with the nonce. It is then appended with the EMT values (EMT) and is now encrypted using the modified Blowfish encryption algorithm and the key K_{server}.

$$\text{Encrypted Data to server} = E\{[EMT||\text{Encrypted Data}||64-\text{bit date and time}||nonce], K_{\text{server}}\} \quad (8)$$

c) The nonce is appended to the data to guarantee the distinctiveness of the transmitted message between two devices, preventing replay attacks. Additionally, the nonce ensures message freshness and verifies the sender's authenticity.

- A unique nonce is created and sent alongside the data being transmitted to the medical server to ensure the genuineness of the transmitted data from the wearable device controller.
- The nonce is produced using the Key K_{server} and the current date and time.
- Date and time are represented as a 64-bit value.
- Since Key K_{server} consists of 128 bits, a selection is made between the least significant 64 bits or the most significant 64 bits of K_{server}, utilizing a random number.
- A 64-bit nonce is generated by performing an XOR operation between 64 bits obtained from K_{server} and the date and time.

$$\text{nonce (64 bit)} = K_{\text{server}}(\text{LS 64 bits or MS 64 bits}) \oplus (\text{date and time}) \quad (9)$$

7) Design of Key K_{server} refreshing phase:
The functional key is regularly renewed to prevent unauthorized access to sensitive medical information through key-based attacks. This key will be refreshed at scheduled intervals, predetermined, and allocated within the device during its initial integration into the human body.

- The key K_{server} is periodically refreshed.
- The block diagram of key K_{server} refreshing is shown in Figure 4. By EXORing the date and time that is represented as a 128-bit value with the key K_{server}, a new key K_{server} is generated.
After refreshing, K_{server} becomes,

$$K_{\text{server}} = \text{Date Time} \oplus K_{\text{server}} \quad (10)$$

- This refreshing of the key is performed both by the wearable device controller and the medical server.
- As the date and time are taken for refreshing, the wearable device controller and the server will refresh K_{server} at the same time.
- This ensures that both the wearable device and the server will have the same key simultaneously.
- The time of refreshing the key K_{server} for various devices is kept different.

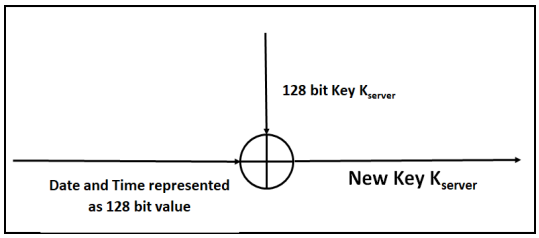


FIGURE 4. K_{server} key refreshing.

8) Storage of the data in the medical server.

a) The data received from the wearable device controller is decrypted at the medical server end using the key K_{server}.

$$\text{Decrypted Data in server} = D\{\text{Encrypted Data to server}, K_{\text{server}}\} \quad (11)$$

b) Stored data is still in the encrypted form as the data is still encrypted using the key K_{doc} and key K_{EMT}.

9) Accessing data by various entities:

a) Accessing data by the EMT:
When the Flag value is all high, EMT data is available and will be decrypted using the EMT key K_{EMT} and modified blowfish algorithm.

$$\text{Decrypted EMT Data} = D\{\text{Encrypted EMT data}, K_{\text{EMT}}\} \quad (12)$$

b) Accessing data by the doctor:
Whenever the doctor is required to access the patient data, the data will be decrypted using key K_{doc} and a modified blowfish algorithm.

$$\text{Decrypted data} = D\{\text{Encrypted data}, K_{\text{doc}}\} \quad (13)$$

c) Accessing data by the administrator:
The administrator cannot access any data from the server as he has neither the EMT nor the Doctor key.

V. RESULT AND ANALYSIS

The efficiency of the work done is analyzed by comparing various key refreshing techniques with the proposed method, by proving the randomness of the refreshed keys, comparing nonce generation methods in various papers with the proposed work and also by providing security analysis by identifying various threats.

The analysis of the proposed work is done using Matlab. The computation was carried out on the configured setup as 8GB RAM on Intel(R) core(TM) i5-7200U CPU at 2.5GHz.

A. THE RANDOMNESS OF THE KEY K_{SERVER} AFTER REFRESHING

The following tests were conducted to check for the randomness of the keys generated after refreshing:

1) Runs test:
A run refers to a continuous sequence of similar bits. If an identical K-bit sequence is followed by a bit of the opposite value, this pairing is termed a 'run length' and denoted as 'k'. The runs test assesses whether the occurrence of zero runs and ones varying in length is equally probable in a random sequence.
In the runs test, the hypothesis under consideration is that the sequence is generated randomly. Evidence against this null hypothesis is condensed into a test statistic, which is utilized to calculate a P-value. If the resulting P-value exceeds 0.01, the null hypothesis is accepted, suggesting randomness in the sequence. Conversely, if the P-value is less than or equal to 0.01, the null hypothesis is rejected, indicating that the sequence is not random. The following steps were performed to obtain the P-value:
   - Calculate the initial proportion π of ones in the input sequence using the formula:
   $$\pi = \frac{\Sigma jEj}{n} \tag{14}$$
   - For instance, if E= 1001101011, then n=10 and $\pi = 6/10 = 3/5$.
   - Next, compute the test statistic using the formula:
   $$V\,(obs) = \sum_{k=1}^{n-1} r(k) + 1, \tag{15}$$
   where r(k)=0 if $E_k = E_k + 1$, and r(k)=1 otherwise.
   - Finally, compute the P-value using the following expression:
   $$\text{P - value} = \frac{\text{erfc}(\text{Vn}(\overline{\text{obs}}) - 2\text{n}\pi(1-\pi)}{(2\sqrt{2\text{n}\pi(1-\pi)})} \tag{16}$$
   where erfc represents the complementary error function. The Runs test is executed on the refreshed key, denoted as Kserver. The graph depicted in Figure 5 illustrates the P-value across various instances of generated Kserver keys. Analysis of the graph indicates that, for all examined cases, the calculated P-value exceeds 0.01. Consequently, we can infer that the generated K_{server} possesses the characteristic of a random number sequence.

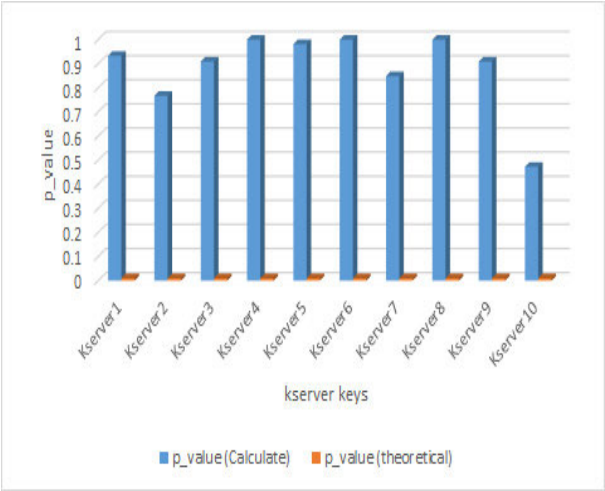


FIGURE 5. Runs Test for the keys K_{server}.

2) Frequency Test within a block:
A frequency test is executed to assess whether the occurrence of 'ones' in a block comprising M-bits is roughly around M/2.
When conducting the frequency test within a block, the null hypothesis posits that the sequence is generated randomly. The test statistic is applied to calculate a P-value, which reflects the level of evidence against the null hypothesis. A conclusion is drawn that the sequence indicates randomness if the resulting P-value is greater than 0.01. Conversely, the sequence lacks randomness if the P-value obtained is less than or equal to 0.01.
The following steps were performed to obtain the P-value:
   - The input bit sequence is divided into N nonoverlapping blocks of size n/M, with any extra bits discarded. For instance, if n = 10, M = 3, and the input is 0110011010, three blocks (N=3) are formed: 011, 001, and 101. The last 0 is excluded.
   - The proportion πi of ones in each M-bit block is determined using the following equation:
   $$\chi 2(obs) = 4M \sum_{i=0}^{N} (\pi i - 1/2)^2 \tag{17}$$
   - The P-value is calculated as follows:
   $$P - value = igamc(N/2, \chi 2(obs)/2), \tag{18}$$
   where igamc represents the incomplete gamma function for Q(a,x).
   Note that Q(a,x) = 1-P(a,x).

The plot in Figure 6 displays P-values corresponding to variously generated Kserver keys. Notably, a considerable

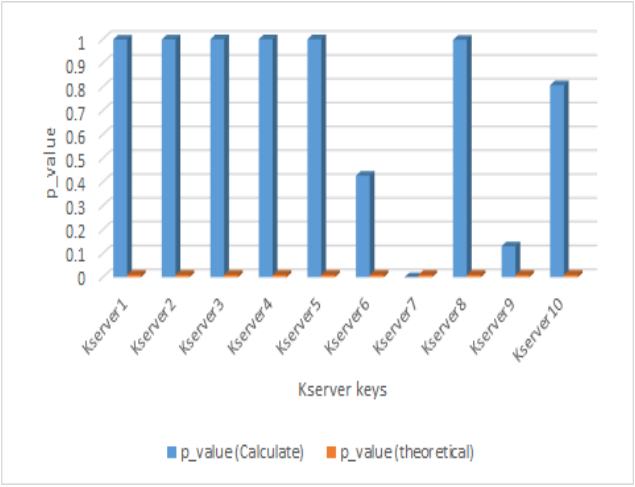


FIGURE 6. Frequency test within the block for the keys K_{server}.

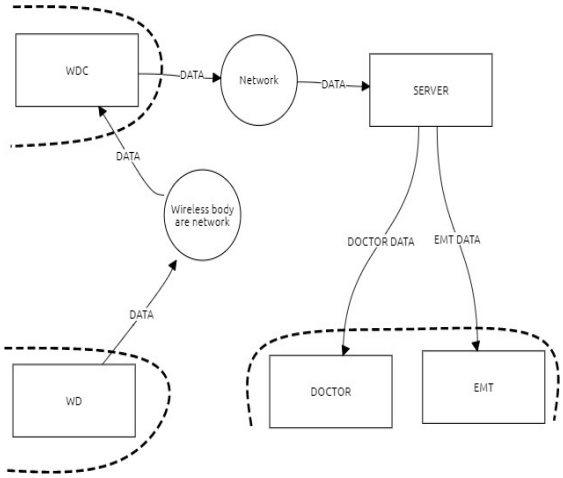


FIGURE 7. Proposed threat model.

number of cases yield P-values exceeding 0.01. This observation leads to the inference that the generated Kserver possesses the characteristics of a random number.

B. COMPARISON OF NONCE GENERATION WITH THE EXISTING WORK

Table 3 compares the proposed approach for nonce generation and other similar methods based on the intended use of the nonce. The novel technique generates nonce for multiple purposes, including authentication, prevention of replay attacks, and signaling the message's freshness. In all the referenced papers, a random nonce is generated. In the paper by Wu et al. [16], along with the random nonce, a separate timestamp is appended. The proposed work generates a nonce from the K_{server} and timestamp. So, the nonce generated is unique, and it is proved that it is random. The overhead bits in the proposed work are less compared to Wu et al. [16].

C. COMPARISON OF KEY REFRESHING WITH THE EXISTING WORK

Table 4 illustrates a comparison between the proposed methodology and other related approaches in terms of key refreshing operations. The proposed technique introduces a Kserver capable of enhancing data privacy. It is noteworthy that this approach involves fewer computations in contrast to the existing methods.

D. COMPARISON OF PRIVACY TECHNIQUE WITH THE EXISTING WORK

Chatterjee et al. [23] is compared with the proposed work and is indicated in Table 5. The comparison shows that the messages communicated between the devices and the entities is less, which is one of the requirements of WBAN.

E. DETAILS OF SECURITY ANALYSIS

The proposed work accomplishes several key objectives, including establishing a confidential key, periodic renewal of the generated key, facilitation of mutual authentication between the wireless device controller and the medical server, and data encryption.

The literature [29], [31] addresses a range of potential attacks. This subsection explores how the proposed approach effectively safeguards against diverse attacks, protecting sensitive patient data from adversarial threats. OWASP Threat Dragon is a modeling tool for creating threat model diagrams in a secure development lifecycle. It records possible threats, decides on their mitigations, and visually indicates the threat model components and surfaces.

Figure 7 indicates the threat model for various entities in the proposed work. Table 6 indicates various attacks encountered at entities and whether they have been mitigated. A description of the mitigation of the attacks is given below:

1) Eavesdropping attack:
   Should an unauthorized party gain access to the transmitted messages, it could lead to an eavesdropping attack.
   However, this risk is mitigated by utilizing the modified Blowfish encryption algorithm for securing messages exchanged between the wearable device controller and the medical server and wearable device and wearable device controller. Consequently, the vulnerability to eavesdropping attacks is effectively eradicated.
2) Replay Attack:
   This form of attack arises when an intruder attempts to resend messages that were sent earlier.
   To counteract this, a nonce value is included in every instance of data transmission. This measure thwarts replay attacks. Additionally, message freshness is ensured by leveraging the current system date and time in the nonce computation process. The attacker's ability to replicate the nonce is hampered, given its calculation is contingent upon the entity key, system date, and time.
3) Impersonation Attack:
   This attack occurs when a malicious actor assumes the role of an authorized user to compromise the system.

TABLE 3. Comparison of the proposed work with other work.

Work	Nonce generation	nonce length	authentication	Replay attack	freshness	Timestamp
Chatterjee [19]	random number is used	NA	✓	✓	✓	
Wu et al. [16]	A random nonce is generated	NA	✓	✓	✓	✓
Chunka and Banerjee [17]	Nonce generated is random	NA	✓	✓	✓	
Hussain et al. [18]	A random number is used	NA	✓	✓		
Proposed method	Timestamp and Key Kserver	64 - bits	✓	✓	✓	✓

TABLE 4. Comparison of the proposed key refreshing technique with other work.

Work	Key Refreshing method	Technique	Remarks
Al-Janabi et al. [20]	Random selection of values from already existing biometric measurements that have been measured by sensor nodes	Re-keying	To avoid cryptanalytic attacks or long term attack, keys are refreshed at regular intervals.
Ali and Khan [21]	One way hash function is used for key refreshing	Key Update	To maintain forward secrecy
Irum et al. [22]	By applying keyed hash function on feature blocks	Key Update	To avoid cryptanalytic attacks
Proposed method	By XORing date and time with Key K_{server}	Key Update	Eliminate key compromising attack and to maintain forward secrecy

TABLE 5. Comparison of privacy technique with the existing work.

Work	Method	Access	Remarks
S. Chatterjee et al. [23]	Allows the user to authenticate at the sensor node under certain access privileges Both. the user and the sensor node from which the user wants to access data can establish a secret session key between them and access data.	Communication between entities and the devices	Sensor nodes are accessed for the required data by entity.
Proposed work	Individual entity data are stored in the server in encrypted form using an entitity key.	The server is accessed for required data by the entities.	There is no communication between entities and devices.

TABLE 6. Mitigation of attacks by various devices.

Attacks	WD -> WDC	WDC -> Server	WD	WDC	Server
Eavesdropping attack	✓	✓			
Replay attack	✓	✓			
Impersonation attack			✓	✓	✓
Message modification attack	✓	✓			
Key compromising attack			✓	✓	✓
Anonymous and Unlinkable attack	✓	✓			

The likelihood of such an attack is quite low because the wearable device ID is generated from an individual's ECG data, which authenticates the wearable device controller. Additionally, this ID is stored in an encrypted format within the server, further enhancing its security.

4) Message Modification Attack:
Because encryption is applied to the transmitted data, any potential attacker must decrypt the message. Decryption, however, requires knowledge of the encryption key. Therefore, this type of attack remains unfeasible. Additionally, the information stored on the server is encrypted, ensuring that only individuals possessing the proper decryption key can access and decipher the content.

5) Key Compromising Attack:
A potential scenario involves the risk of a wearable device controller, responsible for transmitting data to a server, being vulnerable to compromise. This could enable an attacker to acquire crucial information illicitly. However, the situation is mitigated by the periodic execution of key refreshing during operations. This process generates novel encryption keys at regular intervals, significantly raising the difficulty level for an attacker seeking to ascertain the current key.

6) Anonymous and unlinkable sessions: The intercepted communication data does not allow an attacker to access the identity of the wearable device and the wearable device controller engaged in communication. Moreover, it is impossible to establish connections

between different sessions of the same device since the transmitted data does not include any information that could potentially disclose the device's identity.

VI. CONCLUSION

In research work, the data communicated is encrypted using the Blowfish encryption algorithm, which has been proven fast and requires less memory [9], [10]. Since the data is encrypted, the privacy of the data transmitted and stored is maintained.

The nonce is generated within the wearable device controller by utilizing the current system date and time, signifying the precise moment of its generation. As a result, the produced nonce serves both as a timestamp and for authentication, given that the device key contributes to its generation. This ensures the nonce's uniqueness. Additionally, the method is computationally efficient due to the absence of intricate operations.

Since the update of the key K_{server} is synchronized using the date and time between the wearable device controller and the medical server, there is no delay in this update process. Con-sequently, the refreshed key K_{server} will be identical at both ends, generated through a minimal computational step involving a single XOR operation. Additionally, communication between the wearable device controller and the medical server is optimized, reducing the number of exchanged messages. This efficiency is because the refreshed keys do not need to be repeatedly transmitted between the two devices.

The paper demonstrates that the key K_{server} possesses a random number property, as verified through the runs test and frequency test within the block. As a result, it can be inferred that the keys are indeed secure. Furthermore, the approach employed for key refreshment is highly efficient, as it requires minimal computation.

The privacy of data is maintained by giving access to only the entity to whom the data belongs from the server. Therefore, the amount of access by the entity to the wearable device controller is minimal.

As part of future work, only if there is EMT data can the field for EMT be kept in the data transmitted. This reduces the length of the data transmitted rather than the fixed length currently proposed in the work done.

REFERENCES

[1] S. González-Valenzuela, X. Liang, H. Cao, M. Chen, and V. C. M. Leung, *Body Area Networks*. Berlin, Germany: Springer, 2013, pp. 17–37, doi: 10.1007/5346_2012_26.

[2] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Gener. Comput. Syst.*, vol. 78, pp. 956–963, Jan. 2018.

[3] S. M. Raazi, H. Lee, S. Lee, and Y.-K. Lee, "BARI+: A biometric based distributed key management approach for wireless body area networks," *Sensors*, vol. 10, no. 4, pp. 3911–3933, Apr. 2010.

[4] R. V. Sampangi, S. Dey, S. R. Urs, and S. Sampalli, "A security suite for wireless body area networks," 2012, *arXiv:1202.2171*.

[5] S. Divya, K. V. Prema, and B. Muniyal, "Privacy preservation mechanism for the data used in image authentication," in *Proc. IEEE Int. Conf. Distrib. Comput., VLSI, Electr. Circuits Robot. (DISCOVER)*, Aug. 2019, pp. 1–6.

[6] W. Stallings, *Cryptography and Network Security Principles and Practice*, 3rd ed. London, U.K.: Pearson Education, 2017.

[7] S. Divya, K. Prema, and B. Muniyal, "Modified blowfish encryption algorithm for wireless body area network," in *Proc. Int. Conf. Appl. Techn. Inf. Security.* Singapore: Springer, 2022, pp. 98–108.

[8] 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 Informat. J.*, vol. 18, no. 2, pp. 113–122, Jul. 2017.

[9] A. A. Milad, "Comparative study of performance in cryptography algorithms (blowfish and skipjack)," *J. Comput. Sci.*, vol. 8, no. 7, pp. 1191–1197, Jul. 2012. [Online]. Available: https://www.researchgate.net/publication/250614450

[10] A. V. Mota, S. Azam, B. Shanmugam, K. C. Yeo, and K. Kannoorpatti, "Comparative analysis of different techniques of encryption for secured data transmission," in *Proc. IEEE Int. Conf. Power, Control, Signals Instrum. Eng. (ICPCSI)*, Sep. 2017, pp. 231–237.

[11] P. Patil, P. Narayankar, N. D. G., and M. S. M., "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and blowfish," *Proc. Comput. Sci.*, vol. 78, pp. 617–624, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050916001101

[12] R. Patel and P. Kamboj, "Security enhancement of blowfish block cipher," in *Smart Trends in Information Technology and Computer Communications*, A. Unal, M. Nayak, D. K. Mishra, D. Singh, A. Joshi, Eds. Singapore: Springer, 2016, pp. 231–238.

[13] T. F. G. Quilala, A. M. Sison, and R. P. Medina, "Modified blowfish algorithm," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 12, no. 1, p. 38, Oct. 2018.

[14] K. Ding and Y. Tan, "Comparison of random number generators in particle swarm optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2014, pp. 2664–2671.

[15] K. Bhattacharjee, K. Maity, and S. Das, "A search for good pseudo-random number generators: Survey and empirical studies," 2018, *arXiv:1811.04035*.

[16] L. Wu, Y. Zhang, L. Li, and J. Shen, "Efficient and anonymous authentication scheme for wireless body area networks," *J. Med. Syst.*, vol. 40, no. 6, p. 134, 2016.

[17] C. Chunka and S. Banerjee, "An efficient mutual authentication and symmetric key agreement scheme for wireless body area network," *Arabian J. Sci. Eng.*, vol. 46, no. 9, pp. 8457–8473, 2021.

[18] S. J. Hussain, M. Irfan, N. Z. Jhanjhi, K. Hussain, and M. Humayun, "Performance enhancement in wireless body area networks with secure communication," *Wireless Pers. Commun.*, vol. 116, no. 1, pp. 1–22, Jan. 2021.

[19] K. Chatterjee, "An improved authentication protocol for wireless body sensor networks applied in healthcare applications," *Wireless Pers. Commun.*, vol. 111, no. 4, pp. 2605–2623, Apr. 2020.

[20] S. Al-Janabi, A. J. Dawood, and E. H. Hassan, "Development and simulation of enhanced key management scheme for WBANs," *J. Univ. Hum. Develop.*, vol. 1, no. 2, pp. 280–287, 2015.

[21] A. Ali and F. A. Khan, "Energy-efficient cluster-based security mechanism for intra-WBAN and inter-WBAN communications for healthcare applications," *EURASIP J. Wireless Commun. Netw.*, vol. 2013, no. 1, pp. 1–19, Dec. 2013.

[22] S. Irum, A. Ali, F. A. Khan, and H. Abbas, "A hybrid security mechanism for intra-WBAN and inter-WBAN communications," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 8, Aug. 2013, Art. no. 842608.

[23] S. Chatterjee, A. K. Das, and J. K. Sing, "A novel and efficient user access control scheme for wireless body area sensor networks," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 26, no. 2, pp. 181–201, Jul. 2014.

[24] M. Soni and D. K. Singh, "Privacy-preserving secure and low-cost medical data communication scheme for smart healthcare," *Comput. Commun.*, vol. 194, pp. 292–300, Oct. 2022.

[25] A. Ali and F. A. Khan, "Key agreement schemes in wireless body area networks: Taxonomy and state-of-the-art," *J. Med. Syst.*, vol. 39, no. 10, pp. 1–14, Oct. 2015.

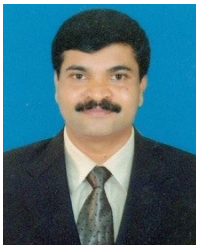
[26] A. K. Das, S. Zeadally, and M. Wazid, "Lightweight authentication protocols for wearable devices," *Comput. Electr. Eng.*, vol. 63, pp. 196–208, Oct. 2017.

[27] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "EKG-based key agreement in body sensor networks," in *Proc. IEEE INFOCOM Workshops*, Apr. 2008, pp. 1–6.

[28] S. Divya, K. V. Prema, and B. Muniyal, "Efficient key generation techniques for wireless body area network," *Int. J. Wireless Inf. Netw.*, vol. 30, no. 3, pp. 270–281, Sep. 2023.
[29] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517301492
[30] M. S. Ramya, P. S. Soman, and L. R. Deepthi, "A novel approach for image security using reversible watermarking," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 338–343.
[31] P. C. Paul, J. Loane, G. Regan, and F. McCaffery, "Analysis of attacks and security requirements for wireless body area networks-a systematic literature review," in *Proc. Eur. Conf. Softw. Process Improvement.* Edinburg, UK: Springer, 2019, pp. 439–452.

DIVYA S. received the B.E. degree in electrical and electronics engineering and the M.Tech. degree in digital electronics and advanced communication from Manipal Academy of Higher Education, Manipal, India, where she is currently pursuing the Ph.D. degree. She is an Assistant Professor Selection Grade with Manipal Institute of Technology, Manipal. She has 17 years of experience in teaching. Her current research interests include wireless body area and network security.

PREMA K. V. (Member, IEEE) is currently a Professor and the Head of the Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru, India. She has a teaching experience of 32 years. She has guided many Ph.D. students and has many publications in national and international conferences/journals to her credit. Her publication topics are the Internet of Things, agriculture, batch processing (computers), batch processing (industrial), client-server systems, cloud computing, computer centers, computer network security, crops, data handling, data integrity, data privacy, decision trees, digital signatures, energy conservation, feature selection, learning (artificial intelligence), least squares approximations, naive Bayes methods, nearest neighbor methods, neural nets, parallel processing, plant diseases, power-aware computing, and public key cryptography. She has reviewed many papers for different international journals/conferences.



BALACHANDRA MUNIYAL (Member, IEEE) received the B.E. degree in computer science and engineering from the University of Mysuru and the M.Tech. and Ph.D. degrees in computer science and engineering from Manipal Academy of Higher Education, Manipal, India. Currently, he is a Professor with the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal. He has 30 years of teaching experience in various institutes. His research interest includes network security. He has more than 60 publications in national and international conferences/journals.