Secure Data Transmission Beyond Tier 1 of Medical Body Sensor Network

Sohail Saif and Suparna Biswas

Abstract Medical body sensor network (MBSN), a three-tier architectural network, has been in wide use on demand for remote health monitoring and support in both urban and rural areas. Primary concern of such system is security of sensitive health data along with low end-to-end delay and energy consumption among others. This paper implements secure patient data transmission between tier 2 and tier 3 by ensuring confidentiality and integrity. Man-in-middle attack and distributed denial of service attack can be detected based on end-to-end delay in data transmission. Hash-based secret key is used for encryption which is generated using extracted biological information of user at coordinator PDA of MBSN. Using shared extracted biological information, secret key is regenerated at cloud-based medical server for decryption of data. Experimental results show using different symmetric key encryption techniques, maximum end-to-end delay is only 11.82% of 250 ms which is the maximum permissible delay limit for healthcare application.

Keywords Medical body sensor network · Confidentiality · Integrity · Hash Secret key · End-to-end delay · Man-in-middle attack · Denial of service attack

1 Introduction

With the rapid advancement in semiconductor technology and communication networks, wireless sensor network has been realized for solving problems in various domains including defence, health care, gaming and entertainment. Sensors are wearable or implanted in human body for regular sensing of vital physiological parameters for continuous monitoring and support purposes. The specialized wireless sensor net-

S. Saif · S. Biswas (✉)
Department of Computer Science & Engineering, Maulana Abul Kalam
Azad University of Technology, Bidhan Nagar, West Bengal, India
e-mail: mailtosuparna@gmail.com

S. Saif
e-mail: sohailsaif7@gmail.com

M. Chakraborty et al. (eds.), *Proceedings of International Ethical Hacking Conference 2018*, Advances in Intelligent Systems and Computing 811,
https://doi.org/10.1007/978-981-13-1544-2_33

work applied for remote health monitoring and support may be termed as wireless body area network (WBAN) or wireless body sensor network (WBSN) or medical body sensor network (MBSN), and people who may be in demand of such system are from both rural and urban areas [1]. Moreover, the purpose of using such system has both the perspectives: fitness regime and continuous health monitoring remotely without being manned and without any mobility restriction for elderly people living in rural areas. This system is advantageous over traditional doctor–patient system for mainly: (i) precision in data measurement, (ii) timeliness or proper interval of physiological data measurement and (iii) physically seeing a doctor for measuring blood pressure, sugar, temperature, heart rate [2], etc., which are irrelevant. But complete success of such remote healthcare system depends completely on how accurately actual measured data are being received at the diagnosis end within permissible delay limit of 10–250 ms [3]. Vital signals sent wirelessly may face threats in the form of (i) modified information—attacks towards integrity, (ii) leakage to sensitive patient information—attacks towards confidentiality, (iii) data access by illegitimate user—attacks towards authentication and (iv) data lost or delayed in transit—may cause havoc on patient's life. A lot of works have been done to address these issues [4]. In tier 1, sensor nodes send signal to the coordinator node over short range (distance around 2 m). Coordinator node then forwards the signals to the tier 2 through access point (AP). Here, various intra-BSNs, cellular networks and Internet connectivity take place. In tier 2 communication, APs and Internet are connected with cloud-based medical server from where concerned caregiver can access the vital signals. Signal being transmitted beyond tier 1 is vulnerable to all types of security attacks in transit through insecure wireless channel.

Figure 1 depicts security measures taken at tier 2 and tier 3 to ensure confidentiality and integrity here.

Rest of the paper is organized as follows: Sect. 2 describes related works, proposed work and algorithm are illustrated in Sect. 3, Sect. 4 elaborates experimental set-up followed by detailed experimental results in Sect. 5, and finally, the whole work is concluded in Sect. 6.

2 Related Work

Physiological data of patients transmitted through MBSN from patient to doctor are vulnerable for different attacks. Intruders can easily alter this vital health information which can be life threatening. Therefore, strong security techniques are needed to achieve proper confidentiality, authenticity and integrity of data. Researchers proposed key generation techniques from biological information [5], and some researches show the authentication [6, 7] techniques using biometric information of human body such as ECG, finger print and heart rate. Table 1 shows the recent works on various security aspects for BSN applications.

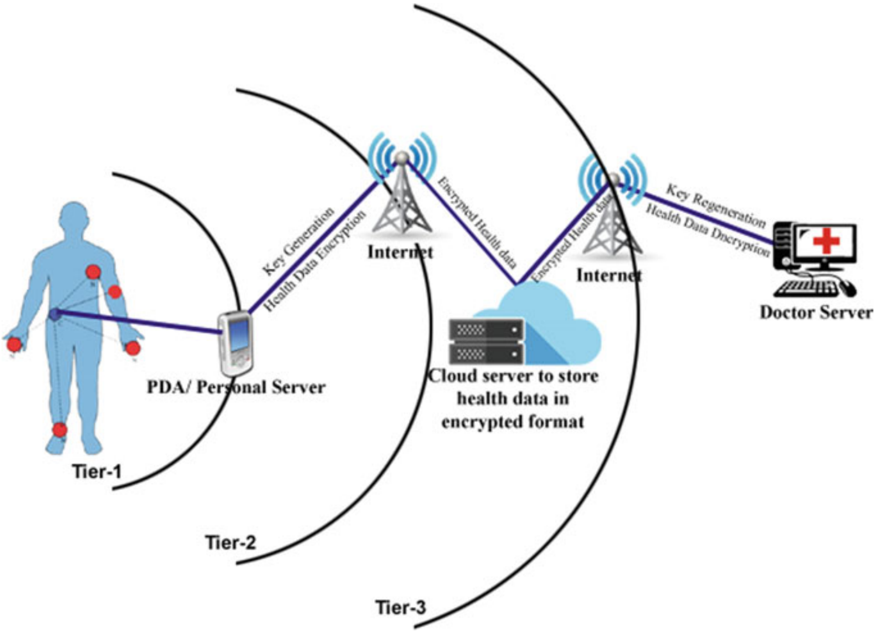

Fig. 1 Three-tier architecture of MBSN

Table 1 Comparative literature survey

Authors, year	Confidentiality	Authentication	Integrity	Implementation	Delay
Liu et al. [8], (2016)	✓	✓	×	✓	✓
Debiaoetal. [9], (2017)	×	✓	×	✓	×
Zhengetal. [10], (2016)	✓	✓	×	×	×
Ramlietal. [7], (2013)	×	✓	×	×	×
Razaetal. [4] (2016)	✓	✓	×	✓	✓

3 Proposed Work

Our proposed security system is implemented in cloud which will secure the data transmission between tier 2 and tier 3. Figure 2 shows the key generation process from fingerprint of patient, key sharing process, encryption and decryption.

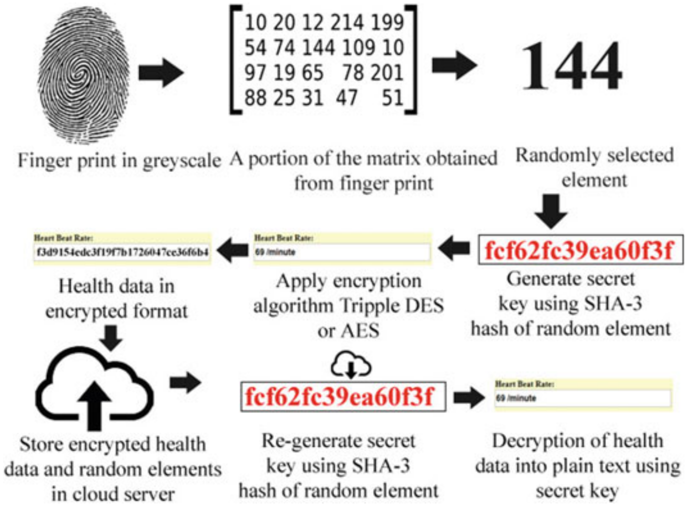
10 20 12 214 199
54 74 144 109 10
97 19 65 78 201
88 25 31 47 51
144
Finger print in greyscale
A portion of the matrix obtained from finger print
Randomly selected element
Heart Beat Rate:
f3d9154edc3f19f7b1726047ce36f6b4
Heart Beat Rate:
69 /minute
fcf62fc39ea60f3f
Health data in encrypted format
Apply encryption algorithm Tripple DES or AES
Generate secret key using SHA-3 hash of random element
fcf62fc39ea60f3f
Heart Beat Rate:
69 /minute
Store encrypted health data and random elements in cloud server
Re-generate secret key using SHA-3 hash of random element
Decryption of health data into plain text using secret key

Fig. 2 Key generation, encryption, key regeneration and decryption process

Firstly, fingerprint images of patients and doctors are captured, and it is converted to its equivalent matrix, and then the matrix is stored to cloud server along with doctor and patient id, respectively. For the authentication of doctor's identity, doctor's fingerprint information is used, and for the key generation purpose, patient's fingerprint information is used. A random element is selected from the fingerprint matrix. We have used two different encryption techniques: Algorithm 1 shows the steps using triple DES [11], while Algorithms 2 shows the steps using AES-128 [6, 12].

For AES, 128-bit hash of randomly selected element is computed using SHAKE-128(SHA-3) algorithm, and then, this hash is used as the encryption key. After encryption, encrypted health data and the random element are stored in the cloud database. During decryption, random element and encrypted health data are fetched, and decryption key is generated by computing the hash of random element again.

For 3DES, three numbers of 64-bit key are generated; for that, three random elements are chosen and 64-bit hash is calculated for all these elements using SHAKE-128(SHA-3) algorithm. Here, 3DES is followed by encryption–decryption–encryption for encryption, where in each step random element is stored in cloud along with key id (K1, K2, K3). For decryption, reverse process of encryption is followed which is decryption–encryption–decryption; here, for each step keys are fetched from cloud. Then, the hash of K1, K2, K3 is again computed which are used as a decryption key.

Algorithm 1: Implementation using Triple DES

```
BEGIN:
Procedure: Thumb image to matrix conversion
For i = 1 to n do
    scan thumb and get image
    Img[i]= scanned image
    grayscale[i]= img[i]
    M[i]:=imread(‘grayscale[i]’)
    generate matrix M
    thumb_info:=M
    store thumb_info in cloud server
end for
Procedure: Authentication, key generation, encryption and transmission of health data
if doctor_request = true then
    if doc_thumb_stored = doc_thumb_sent then
        for i = 1 to n do
            a[i] = vital signals of the patient's measured
            for i = 1 to 3 do
                fetch thumb_info from cloud server
                rand_ele[i]= choose random element from thumb_info
                pkey[i]= SHA-3(rand_ele[i]) // compute 64 bit hash of element
            end for
        end for
        enc[i] = DES encryption of the a[i] using pkey[i]
        dec[i] = DES decryption of enc[i] using pkey[i]
        enc[i] = DES encryption of dec[i] using pkey[i]
        Store enc[i] and rand_ele[i] in cloud server
    Else
        Request decline
    end if
end if
Procedure: key re-generation and decryption of health data
for i = 1 to ndo
    Fetch enc[i] and rand_ele[i] from cloud server
    for i = 1 to 3 do
        dkey[i]= SHA-3(rand_ele[i]) // compute 64 bit hash of random element
    end for
```

```
        dec[i] = DES decryption of enc[i] using dkey[i]
        enc[i] = DES encryption of dec[i] using dkey[i]
        a[i]   = DES decryption of enc[i] using dkey[i]
    end for
END
```

Algorithm 2: Implementation using AES-128

```
BEGIN:
Procedure: Authentication, key generation, encryption and transmission of health data
if doctor_request = true then
    ifdoct.biosignal_pat = doct.biosignal_sentthen
        for i = 1 to ndo
            a[i] = vital signals of the patient's measured
            Fetch thumb_info from cloud server
            rand_ele[i]= choose random element from thumb_info
            pkey[i]= SHA-3(rand_ele[i]) // compute 128 bit hash of element
            enc[i] = AES encryption of the a[i] using pkey[i]
        end for
        Store enc[i] and rand_ele[i] in cloud server
    Else
        Request decline
    end if
end if
Procedure: key re-generation and decryption of health data
for i = 1 to n do
    Fetch enc[i] and rand_ele[i] from cloud server
    dkey[i]= SHA-3(rand_ele[i]) // compute 128 bit hash of random element
    dec[i] = AES decryption of enc[i] using dkey[i]
end for
END
```

Notations used in the above algorithms are as follows:

1. doctor_request: Doctor sending request to patient for health data
2. doc_thumb_sent: Doctor sending his thumb signal with request of health data.
3. doc_thumb_stored: Thumb information of doctor stored in cloud.
4. thumb_info: Thumb information of patient and doctor in matrix form.
5. n: Number of vital signals.

Table 2 Simulation environment of cloud server

Description	Value/Name
Server location	Burlington, Massachusetts, USA
Processor	2.30 GHZ dual core
RAM	2 GB
Storage	5 GB HDD in RAID
Bandwidth	10 GB/month
Apache version	2.4.27
PHP version	7.1.9
MYSQL version	5.7.19
Architecture	X84_64
OS	Cent OS 6.5

Table 3 Simulation environment of PDA

Description	Value/name
System model	HP G-62 Notebook
Processor	Intel(R) Core (TM) i3 CPU M380 @ 2.53 GHz
RAM	4 GB
Storage	500 GB
Architecture	X84_64
OS	Windows 7.1

4 Experimental Set-Up

For our experiments, a cloud-based database server is used to store physiological signal of patients in encrypted format. A cloud server is also known as virtual private server, but its hardware components are physically located in different places. The main reasons behind using the cloud server are remote access of data, flexibility, cost-effectiveness, etc. Back end of our implemented system is controlled by PHP and AJAX, and front end is designed using HTML 5 and CSS. We have considered a notebook PC here as a PDA device which forwards the physiological signal of patients to the cloud. Table 2 and Table 3 show the simulation environment of PDA device and cloud-based server, respectively. Tables 4 and 5 show the hardware resource utilization during the secure transmission of patient's health data from patient to doctor via cloud server.

We can see from the above tables that resource utilization is very low, so our implementation can be easily adopted in a low-end system also.

Table 4 Resource utilization of cloud server

Description	Usage	Limit	Faults
CPU usage	16.7%	100%	0
I/O usage (Kbps)	1410	8192	0
IOPS	16	1024	0
No. of processes	13	100	0
RAM usage	321 MB	2048 MB	0

Table 5 Resource utilization of PDA

Description	Usage	Limit
CPU usage (%)	29	100
RAM usage (GB)	2.04	4.00
No. of processes	42	100
Network usage	9%	4 MBPS

5 Results and Discussion

We have performed the experimental test in five phases with five runs of 10 min each. The physiological information of patient such as body temperature, heart rate and blood pressure is transmitted in real time from patient to doctor upon request of doctor. In the first phase, physiological data of patients are sent to doctor via cloud server in plain text format without any security with an interval of 1 min. Secondly, physiological data are sent to doctor during attack on transmission. Here, in our scenario we have considered application layer-based http flood attack (DDOS) and man-in-middle (MITM) attack. For the DDOS attack simulation, we have used a Python-based attacking tool named Golden Eye. To simulate the environment, an attack is targeted to apache-based PDA/personal server's IP address from an another PDA device; during simulation, we first sent 20 GET method request to the targeted IP address per second using http protocol for 12 min (with an interval of 1 min) and recorded the end-to-end delay. Next, we sent 50, 110 and 150 GET method requests and recorded the end-to-end delay, respectively, but for the 150 request the targeted system is crashed (system turned-off) because of memory exhaust. We also recorded the CPU utilization of the targeted PDA device during the attack. For the man-in-middle attack, it is periodically and randomly targeted where an attacker reads the health information during transmission and again re-transmits it to doctor. Comparison of average end-to-end delay and resource utilization between DDOS attack and non-attacked period are shown in Figs. 3 and 4, respectively. Comparison of average transmission delay between man-in-middle attack, dos attack and normal traffic is shown in Fig. 5.

Figure 3 shows that end-to-end delay for transmission of physiological signal is increasing during the attack. It is under the permissible delay when requests are 6

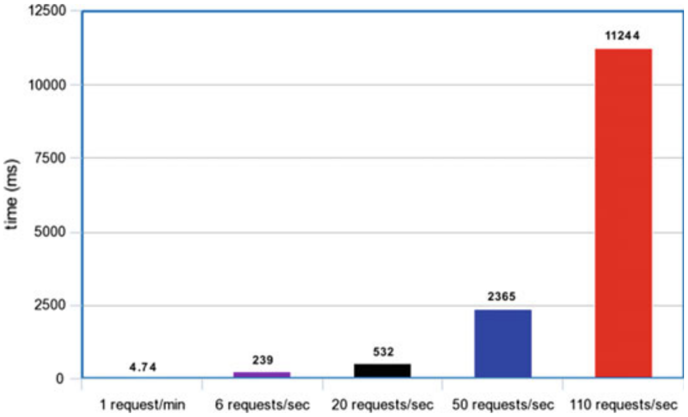

Fig. 3 End-to-end delay of physiological data transmission in plain text during DDOS attack

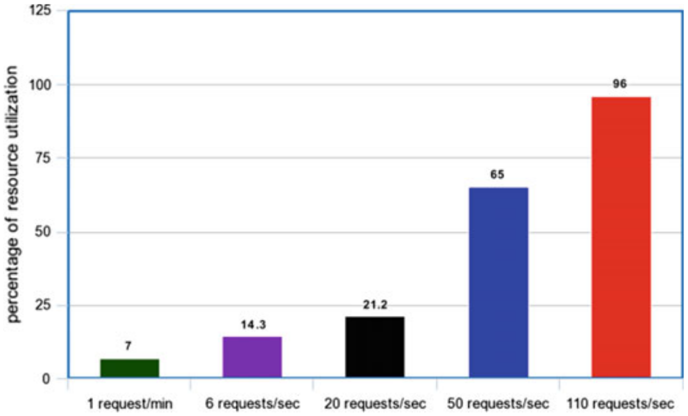

Fig. 4 Resource utilization of PDA device during transmission of physiological data in plain text during DDOS attack

per second but after that it crossed the limit, so here we can detect the attack because of delay. Figure 5 shows that during man-in-the-middle attack, end-to-end delay is more than normal transmission, so if there is any extra delay then we could say that there is an attack.

In third stage, physiological data of patients are transmitted from PDA to doctor via cloud server with confidentiality and authenticity. Here, secure key distribution is done in order to prevent man-in-middle attack on secret key. For our implementation, we have used doctor’s thumb information to achieve authentication, for confidentiality light weight faster symmetric encryption algorithm AES and triple DES is used. To generate secret key, patient’s thumb information has been used. Average data size of the physiological signal of patients is 3.21 KB. Here, we have provided

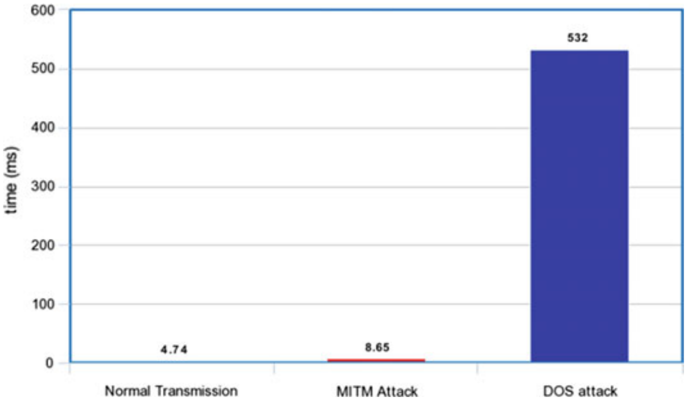

Fig. 5 Comparison of end-to-end delay during normal transmission, MITM and DDOS attack

the averages of five different sets of data. Explanations of various parameters used are as follows:

1. Doctor's thumb authentication time (DTAT): This reflects the required time to compare and validate doctor's bio-signal (thumb print) to the bio-signal stored in cloud database when doctor wants to see patient health data.
2. Data read time (DRT): This means the required time to read the vital signals of patient's from the PDA device.
3. AES encryption time (AES-ET): This shows the time required to encrypt patient's health data using encryption algorithm and the key generation from patient's thumb information.
4. Data store time (DST): This means the time required to save encrypted health data and the information to regenerate the secret key in the cloud database server.
5. Data fetch time (DFT): This shows the time required to fetch encrypted health data and the secret key information.
6. AES decryption time (AES-DT): This reflects the time required to regenerate the secret key from the fetched information and to decrypt the health data into readable format.
7. DES encryption–decryption–encryption time (DES-EDET): This shows the time required for encryption process and the generation of 3 no's of 64-bit key from patient's thumb information.
8. DES decryption–encryption–decryption time (DES-DEDT): This shows the time required for decryption process and the regeneration of 3 no's of 64-bit key from patient's thumb information.

For detailed view and better understanding, total end-to-end delay and its different components of secure patient health data transmission using AES and triple DES encryption with authenticity and confidentiality are plotted graphically in Fig. 6 and Fig. 8, respectively. Results of Fig. 7 show that DTAT lies between 2.23 and 2.67 ms, DRT lies between 0.29 and 0.41 ms, AES-ET lies between 5.32 and 6.01 ms, DST

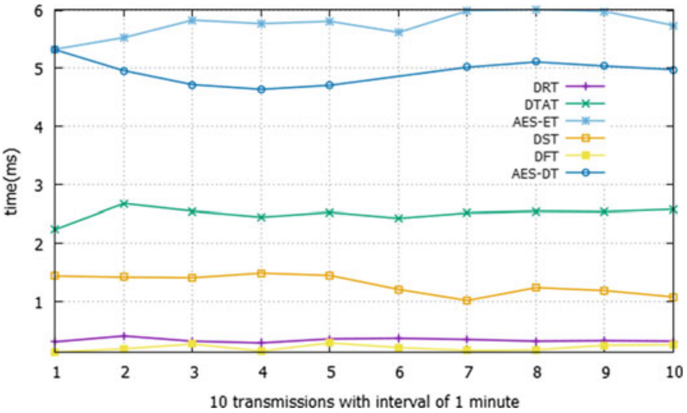

Fig. 6 End-to-end delay of physiological data transmission using AES

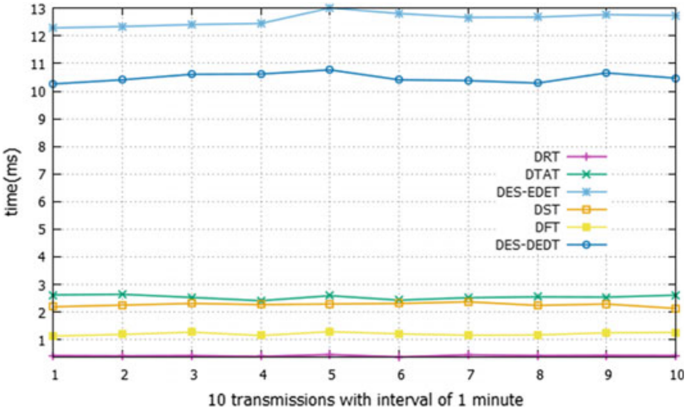

Fig. 7 End-to-end delay of physiological data transmission using triple DES

lies between 1.02 and 1.49 ms, DFT lies between 0.12 and 0.31 ms and AES-DT lies between 4.71 and 5.32 ms. Similarly from Fig. 8, we can understand that DES-EDET and DES-DEDT are increased slightly because of three layers of DES encryption and decryption process and generation of 3 no's of 64-bit key. Average total time required for both AES and triple DES using 128- and 192-bit keys is, respectively, 15.05 ms and 29.54 ms. Hence, end-to-end delay is only 6.02 and 11.82% of 250 ms which is the permissible delay limit for healthcare application.

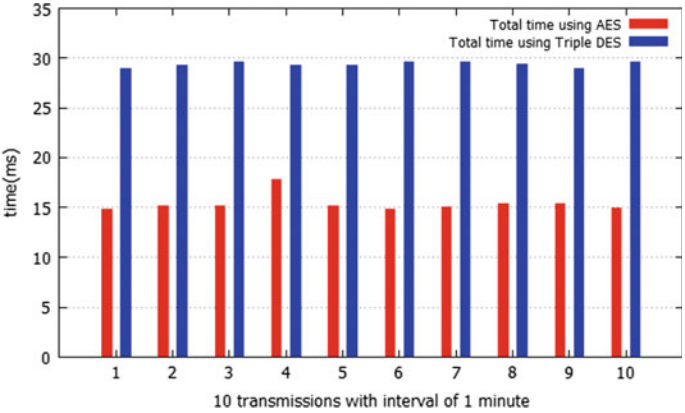

Fig. 8 Comparison between end-to-end delay of physiological data transmission using AES and triple DES

6 Conclusion

This work implements symmetric key encryption techniques such as AES and triple DES for confidentiality of data, and secret key used for encryption and decryption is hash-based key generated using SHA-3 to ensure integrity beyond tier 2 and between tier 2 and tier 3. This secure system implements a complete security to the vital signals being transmitted in open wireless network exploiting biological information extract of the MBSN user for secret key generation. Experimental results satisfy delay constraints of specific application, e.g. healthcare domain demanding real-time streaming of data in spite of additional security measures applied.

References

1. Li, H.-B., Takahashi, T., Toyoda, M., Katayama, N., Mori, Y., Kohno, R.: An experimental system enabling WBAN data delivery via satellite communication links. In: 2008 IEEE International Symposium on Wireless Communication Systems, Reykjavik, pp. 354–358 (2008)
2. Mukhopadhyay, S.C.: Wearable sensors for human activity monitoring: a review. IEEE Sens. J. **15**, 1321–1329 (2015)
3. Movassaghi, S., Abolhasan, M., Lipman, J., Smith, D., Jamalipour, A.: Wireless body area networks: a survey. IEEE Commun. Surv. Tutor. 1–29 (2013)
4. Raza, S.F., Naveen, C., Satpute, V.R., Keskar, A.G.: A proficient chaos based security algorithm for emergency response in WBAN system. In: 2016 IEEE Students' Technology Symposium (TechSym), Kharagpur, pp. 18–23 (2016)
5. Karmakar, K., Saif, S., Biswas, S., Neogy, S.: WBAN Security: study and implementation of a biological key based framework. In: International Conference on Emerging Applications of Information Technology, 12–13 Jan 2018

6. Saif, S., Gupta, R., Biswas, S.: Implementation of cloud assisted secure data transmission in WBAN for healthcare monitoring. In: International Conference on Advanced Computational and Communication Paradigms, 8–10 Sept 2017
7. Ramli, S.N., Ahmad, R., Abdollah, M.F., Dutkiewicz, E.: A biometric-based security for data authentication in wireless body area network (WBAN). In: ICACT 2013, 27–30 Jan 2013, pp. 998–100 (2013)
8. Liu, J., Zhang, Z., Chen, X., Kwak, K.S.: Certificateless remote anonymous authentication schemes for wireless body area networks. IEEE Trans. Parallel Distrib. Syst. **25**(2), 332–342 (2014)
9. He, D., Zeadally, S., Kuma, N., Hyouk Lee, J.: Anonymous authentication for wireless body area networks with provable security. IEEE Syst. J. 1–12 (2017)
10. Zheng, G., Fang, G., Shankaran, R., Orgun, M., Zhou, J., Qiao, L., Saleem, K.: Multiple ECG fiducial points-based random binary sequence generation for securing wireless body area networks. IEEE J. Biomed. Health Inf. **21**(3), 655–663 (2017)
11. He, D., Chan, S., Zhang, Y., Yang, H.: Lightweight and confidential data discovery and dissemination for wireless body area networks. IEEE J. Biomed. Health Inform. **18**(2), 440–448 (2014)
12. Saleem, S., Ullah, S., Yoo, H.S.: On the security issues in wireless body area networks. **3**(3), 178–184 (2009)