Privacy Preserved Medical Service Provider Selection in Cloud-based WBAN

Anupam Pattanayak (✉ anupam.pk@gmail.com)

Indian Institute of Information Technology Guwahati <https://orcid.org/0000-0001-8183-0978>

Mou Dutta

Indian Institute of Information Technology Guwahati

Subhasish Dhal

Indian Institute of Information Technology Guwahati

Research Article

Keywords: Smart Healthcare, MSP Selection, Security and Privacy, MDRQ, Homomorphic Encryption

Posted Date: August 9th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-751602/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Privacy Preserved Medical Service Provider Selection in Cloud-based WBAN

Anupam Pattanayak · Mou Dutta · Subhasish Dhal

Received: date / Accepted: date

Abstract Continuous health status monitoring of the persons equipped with Wireless Body Area Network (WBAN) is one of the key emerging technologies in the area of e-healthcare. In this technology, WBANs and Medical Service Providers (MSPs) are connected via a cloud infrastructure, where WBAN continuously uploads the present readings of physiological parameters via base station such as smart phone, smart watch, etc. Cloud analyzes the physiological parameters and whenever any medical emergency of the user is detected, it checks the infrastructural parameters including the location information of all MSPs connected to it and finds the most appropriate MSP to handle the medical emergency. This MSP selection task has to be performed efficiently so that the medical treatment of the user can start as early as possible. In the existing works, a trusted third party, namely, trusted authority (TA) actively participates in the MSP selection process, which may take undue advantages. The aim of the work in this paper is to remove the role of TA during MSP selection process. However, TA will participate in the system setup phase such that it cannot obtain any advantage in compromising the MSP selection task. The proposed scheme has been scrutinized through an adversary model and simulated using ProVerif tool to verify the security and privacy.

Anupam Pattanayak (✉)
Computer Science and Engineering, Indian Institute of Information Technology Guwahati,
Assam, India - 781015
E-mail: anupam.pattanayak@iiitg.ac.in, anupam.pk@gmail.com

Mou Dutta
Computer Science and Engineering, Indian Institute of Information Technology Guwahati,
Assam, India
E-mail: mou.dutta@iiitg.ac.in

Subhasish Dhal
Computer Science and Engineering, Indian Institute of Information Technology Guwahati,
Assam, India
E-mail: subhasis@iiitg.ac.in

Keywords Smart Healthcare · MSP Selection · Security and Privacy · MDRQ · Homomorphic Encryption

1 Introduction

In a smart society, one natural expectation is that, if a person experiences any sudden medical abnormality, the medical emergency event should be automatically detected and thereafter reported instantly so that medical treatment could be given at the earliest. The advancements in wireless sensor technologies and bio-sensors that can be implanted in human body made these demands a reality through the use of WBAN and cloud. These are used for reading the physiological parameters continuously by WBAN and monitoring this private health information (PHI) of human in a periodic and regular intervals by cloud [1], [2], [3].

In WBAN, responding to a medical emergency is one of the fundamental aim [2]. However, appropriate MSP selection during emergency is also another very important task, as it can provide most appropriate treatment as per patient's need. Lacking proper treatment may put the patient's life at risk. Abnormality of one or more physiological readings of users are analyzed by the cloud, and an emergency is triggered if a combination of physiological readings is not in the normal range. In this situation, the CS helps the Medical User (MU) to select an appropriate MSP based on the nature of emergency. However, there are many security and privacy implications that may restrict this technology for adoption in real life. The MUs, who wish to access this service, have to put almost entire details of their physiological data to the platform, which is outside of their privacy range. This can become a vital security and privacy risk for the users. Therefore, the task of automatic MSP selection that preserves the security and privacy of user is very important in e-healthcare. As per our knowledge, this automatic MSP selection problem has been addressed only in [4]. In this work, the authors have used a third party i.e., trusted authority (TA), which they have considered as fully trusted. However, in this work, the internal agents of TA can access the medical information of the users, which in turn can be misused, and that will put the users in a security and privacy risk. This paper proposes a framework for MSP selection in medical emergency, where the TA is utilized only to setup the system. This framework ensures that an internal agent of TA cannot access any medical information of the user even if the TA participates in setting up the system. Following is the summary of the contributions in this paper:

- I. In the event of emergency, the MSP whose attributes matches most closely with the requirements of MU needs to be selected. In view of this, we have proposed a secure and privacy preserved framework for MSP selection in response to medical emergency of MU, where MU's PHI and location information privacy is preserved.

- II. Our proposed framework handles the single point of failure problem, and prevents the third party from taking any advantage in compromising security and privacy.
- III. Emergency signal is generated whenever the PHI reading is outside the normal range. We have identified false positive emergency signal in [4]. We have designed the proposed framework, which is free from this problem.
- IV. Security and privacy of the proposed framework has been formally analyzed using suitable adversarial model, which ensures the required security and privacy of the proposed framework.

Rest of the paper has been organized as follows. Related works are briefed in Section 2. Next, our proposed framework has been described in Section 3. Security and privacy of the proposed framework has been analyzed in Section 4. In Section 5, performance of the proposed framework has been analyzed and compared with the scheme proposed in [4]. Finally, the work has been concluded in Section 6.

2 Related Works

Several articles [1], [2], [5] [6], [7] have discussed different aspects of WBAN. In particular, S. Movassaghi et al. [1] have discussed in details the IEEE 802.15.6 specifications for WBAN. X. Liang et al. [8] proposed a privacy-preserved emergency call scheme without considering MSP selection problem. K. Zhang et al. [9] have proposed a privacy-preserved priority-based health data aggregation scheme for cloud-assisted WBAN. They have emphasized in the process of aggregation and classification of health data. Q. Huang et al. [10] proposed a secure collaboration and health data sharing scheme. The authors in this work have considered only the privacy preserved collaboration between the users suffering from similar kinds of symptoms, and how they can share the health data among themselves. J. Sun et al. [11] have proposed a privacy-preserved emergency response system in which they assume that a pre-decided primary physician will always access all the health records directly from the WBAN implanted to the user. Chun-Ta Li et al. [12] have proposed a secure cloud-assisted architecture. In this work also, pre-defined medical care providers can directly access and process the medical data of the user. Shin-Yan Chiou et al. [13] proposed an approach to allow medical staff to quickly attend the user during an emergency. In response to an accident notification, the server immediately alerts the pre-defined medical staff for necessary treatment. A cloud-assisted privacy preserved mobile health monitoring system (CAM) has been proposed by H. Lin et al. in [14]. In their work, the patient monitoring program has been modeled as a binary decision tree based on the ranges of the physiological parameters. More specifically, the attribute vector of this measurement is uniquely mapped to a binary bit block with multi-dimensional range query (MDRQ) [15], [16]. Thus, the authors in this work have considered only the emergency detection using MDRQ scheme. We have observed that none of the schemes proposed in [8], [9], [10], [11], [12], [13], [14] considers

the MSP selection problem, and they have emphasized on other aspects of medical emergency. The work proposed by W. Yu [4] considers the secure and privacy preserved MSP selection problem. In this work, the authors have used a centralized third party, namely, Trusted Authority (TA), which is considered as fully trusted. However, the curious internal agents of TA have the scope to compromise the health privacy of the medical users. Moreover, this scheme has the problem of false positive emergency report, which may panic an user. This has been proved in Claim 1.

Claim 1: The scheme proposed by W. Yu [4] generates false positive emergency alert.

Proof: In [4], [14] the authors have used the MDRQ tree as depicted in Fig. 1. Assume the normal range of a PHI is $[0010, 1010]$, and actual reading from MU is 1001, which is normal. Using the approach used in [4], first we obtain the set $S_{[0010, 1010]} = \{001, 01, 100, 1010\}$. The complement of this set is $S_{[0010, 1010]}' = \{\epsilon, 0, 00, 000, 0000, 0001, 1, 10, 101, 1011, 11\}$. Now, let us compute the set $S_{1001} = \{\epsilon, 1, 10, 100, 1001\}$. We see that, $S_{[0010, 1010]}' \cap S_{1001} \neq \phi$. The result illustrates that the current health status of MU is not normal, i.e., emergency, which is not correct. Hence, the proposal of W. Yu [4] generates false positive emergency alert. Thus our claim is proved.

Fig. 1 MDRQ Tree

As we can see that except the work by W. Yu [4], none of the existing works [8], [9] [10], [11], [12], [13], [14] have discussed the problem of secure and privacy preserved MSP selection problem in case of medical emergency. Unfortunately, the scheme proposed by W. Yu in [4] has the privacy leakage problem. This motivates us to design a security and privacy preserved framework to handle the MSP selection problem.

3 Our Proposal

In this section, we present our proposed framework. There are mainly four components, namely, TA, CS, MUs and MSPs. TA is used only during system initialization process, CS is the computing facilitator, MU is equipped with

Table 1 Notations

Symbol	Description
MU	Medical user
MSP	Medical service provider
TA	Trusted authority
CS	Cloud server
PHI	Private health information
q	Prime number
$\mathbb{Z}_q^*$	Multiplicative Group
$\mathbb{G}$	Additive Group of order q
$\mathbb{G}_T$	Multiplicative Group of order q
g	generator of $\mathbb{G}$
e	Bilinear pairing function $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$
n	Cardinality of MU and MSP attributes
r	Random number
N	Nonce
I_i	PHI index
θ_{i1}, θ_{i2}	Range of i^{th} PHI
δ_i	Random values to shift PHI range
H_i	Hash function
P, Q	Invertible matrices
T	Integrity information of MSP
MS, MS'	Private and Public Master key of MUs
β, PK_{MU}	Private and Public key of MU
PR_{CS}, P_{CS}	Private and Public key of CS
PR_{MSP}, PK_{MSP}	Private and Public key of MSP
$E_K()$	Encryption under key K
$D_K()$	Decryption under key K
H'	Transformed PHI
$Normal$	Signal to denote PHI status normal
$\$$	Random signal to denote PHI status not normal
$F_s()$	selection function

WBAN and MSPs are competing for attending the medical emergency. Three algorithms have been designed for the proposed framework. Algorithm 1 is used to initialize several parameters along with the generation and distribution of different keys. Algorithm 2 is used for detecting the medical emergency of the MUs and Algorithm 3 is used for selecting the appropriate MSP from the competing MSPs. Table 1 describes the notations used throughout the rest of this paper.

3.1 Initialization Phase

Algorithm 1 does the system initialization. TA, MU, MSP, and CS participate in this phase. TA generates random numbers r, a and decides the range information of all the PHIs. For example, the range of i^{th} PHI is $\{\theta_{i1}, \theta_{i2}\}$. TA then shifts the ranges of the PHIs by random values $\delta_i (1 \leq i \leq n)$ and obtains the set $S_{[\theta_{i1}+\delta_i, \theta_{i2}+\delta_i]} = \{id\}$ using MDRQ tree, similar to [4]. Then for each $id \in S_{[\theta_{i1}+\delta_i, \theta_{i2}+\delta_i]}$, TA computes C_1 , and sends these C_1 values to CS. Thus,

TA sends C_1 values for all the PHIs over a secure communication channel. CS computes its private and public key pair PR_{CS}, P_{CS} , and sends public key to TA securely. Then, TA generates private and public master secret key pair MS, MS' and sends P_{CS}, r, δ_i, MS to MU, and P_{CS}, MS' to MSP over secure communication channels. The system initialization has been shown in Fig. 2.

Algorithm 1 for Initialization Phase

1. TA generates random numbers $r, a \in \mathbb{Z}_q^*$
 2. TA generates id based on PHI range using MDRQ tree, and random values $\delta_i (1 \leq i \leq n)$ to obtain set $S_{[th_{i1}+\delta_i, th_{i2}+\delta_i]} = \{id | id \in S_i\}$
 3. For each id , TA Computes $C_1 \leftarrow Normal \oplus H_2(e(H_1(id)^a, g^r))$ and sends C_1 values for all the ids indexed by I_i along with $A \leftarrow g^a$ to CS securely
 4. CS computes private and public key pair PR_{CS}, P_{CS} , and sends P_{CS} to TA securely
 5. TA generates private and public key pair $MS, MS' \leftarrow g^{MS}$ for master secret
 6. TA sends P_{CS}, r, δ_i, MS to MU, and P_{CS}, MS' to MSP securely
-

Fig. 2 Initialization of Proposed Protocol

3.2 Emergency Detection Phase

In this phase, it is checked whether the current reading of MU health parameters are in normal/abnormal range, and the MU is alerted whenever the reading found to be in abnormal range. The proposed framework has been designed such that the computation for detecting the medical status is performed

in CS without leaking any sensitive MU information to it. This process is performed through Algorithm 2. MU transforms the present reading of i^{th} PHI by adding δ_i received from TA. It then generates the set $S_{h'_i} = \{id' | id' \in S_{path_i}\}$ collecting the IDs in the path from root to the corresponding transformed PHI value node in MDRQ tree similar to [4]. MU then randomly generates private key β , nonce N , and computes its public key PK_{MU} . Now, for all id' in $S_{h'_i}$, MU computes C'_1, C'_2 , which are integrity information for CS, and N is used for authentication. MU then encrypts the nonce value N and Index of PHI I_i to form $E_{P_{CS}}(N || I_i)$. MU sends all the computed C'_1, C'_2 along with $PK_{MU}, E_{P_{CS}}(N || I_i)$ to CS. Then, CS verifies integrity of the information received from MU using equation 1.

$$\begin{aligned}
& e(g^{C'_1} \cdot PK_{MU} \cdot g^N \cdot g^{I_i}, C'_2) \\
& = e(g^{C'_1} \cdot g^\beta \cdot g^N \cdot g^{I_i}, C'_2) \\
& = e(g^{C'_1 + \beta + N + I_i}, C'_2) \\
& = e(g^{C'_1 + \beta + N + I_i}, g^{\frac{1}{C'_1 + \beta + N + I_i}}) \\
& = e(g, g)^{(C'_1 + \beta + N + I_i) \times \frac{1}{C'_1 + \beta + N + I_i}} \\
& = e(g, g)
\end{aligned} \tag{1}$$

CS stops if the integrity verification is unsuccessful. Otherwise, CS checks for medical emergency using equation 2. If CS obtains only \$ i.e., abnormal value as the output during comparison between all C_1 and C'_1 , it alerts the MU with $E_{PK_{MU}}(N)$. MU then decrypts N and does the validity check to confirm the emergency.

Algorithm 2 for emergency detection phase

1. MU transforms PHI using δ_i , and generates the set $S_{h'_i} = \{id' | id' \in S_{path_i}\}$
 2. MU randomly generates β , nonce N , and computes public key $PK_{MU} \leftarrow g^\beta$
 3. For all id' , MU computes $C'_1 \leftarrow H_1(id')^r, C'_2 \leftarrow g^{\frac{1}{C'_1 + \beta + N + I_i}}$, encrypts N and I_i to form $E_{P_{CS}}(N || I_i)$
 4. MU sends C'_1, C'_2 for all id' along with $PK_{MU}, E_{P_{CS}}(N || I_i)$ to CS
 5. CS verifies the integrity using equation 1, and stops on unsuccessful integrity verification
 6. CS checks for medical emergency using equation 2
 7. **If** CS does not find any value 'Normal' as output during comparison between all C_1 and C'_1 **Then**
 CS alerts MU with $E_{PK_{MU}}(N)$
 MU decrypts N and checks validity of
 the decrypted N to detect emergency event
-

$$\begin{aligned}
& C_1 \oplus H_2(e(C'_1, A)) \\
&= Normal \oplus H_2(e(H_1(id)^a, g^r)) \oplus H_2(e(C'_1, A)) \\
&= Normal \oplus H_2(e(H_1(id)^a, g^r)) \oplus H_2(e(H_1(id')^r, A)) \\
&= Normal \oplus H_2(e(H_1(id), g)^{a \times r}) \oplus H_2(e(H_1(id'), g)^{a \times r}) \\
&= \begin{cases} Normal & \text{if } id = id' \\ \$ & \text{otherwise} \end{cases}
\end{aligned} \tag{2}$$

3.3 MSP Selection Phase

This is an important phase of the proposed framework. In this phase, a particular MSP which has the closest infrastructural match with the requirements of concerned MU is selected in such a way that any third party other than the selected MSP and MU cannot compromise privacy of the user. Algorithm 3, illustrates the MSP selection process. In this algorithm, we use fully homomorphic operations to add or multiply two encrypted values and partial homomorphic operation to multiply a scalar quantity with an encrypted value to form another encrypted value [17],[18], [19]. MU first generates invertible matrices P, Q of size 3×3 and $n \times n$, respectively, and their inverse matrices P^{-1}, Q^{-1} . It conceals its location using P^{-1} , and attributes using Q^{-1} to form C_{L_u} and C_{A_u} illustrated in equations 3 and 4, respectively.

$$C_{L_u} = \begin{pmatrix} p'_{11} \cdot x_1 & p'_{12} \cdot x_2 & p'_{13} \cdot x_3 \\ p'_{21} \cdot x_1 & p'_{22} \cdot x_2 & p'_{23} \cdot x_3 \\ p'_{31} \cdot x_1 & p'_{32} \cdot x_2 & p'_{33} \cdot x_3 \end{pmatrix} \tag{3}$$

$$C_{A_u} = \begin{pmatrix} q'_{11} \cdot u_1 & q'_{12} \cdot u_2 & \cdots & q'_{1n} \cdot u_n \\ q'_{21} \cdot u_1 & q'_{22} \cdot u_2 & \cdots & q'_{2n} \cdot u_n \\ \vdots & & & \\ q'_{n1} \cdot u_1 & q'_{n2} \cdot u_2 & \cdots & q'_{nn} \cdot u_n \end{pmatrix} \tag{4}$$

MU then encrypts its location and attributes in E_{L_u} and E_{A_u} respectively. After this, MU computes its signature H and the ciphertexts $E_{PK_{MU}}(P)$, $E_{PK_{MU}}(Q)$, and then sends these along with $C_{L_u}, E_{L_u}, C_{A_u}, E_{A_u}$ to CS. Now, every MSP receives the signature H_1 and P, Q, PK_{MU}, N, H broadcasted by CS. Each MSP first checks the integrity and verifies whether the received information are from correct MU and CS or not. After successful integrity verification, the MSP generates private and public key pair PR_{MSP}, PK_{MSP} . MSP then conceals its location in C_{L_s} and attributes in C_{A_s} using equations 5 and 6, respectively.

$$C_{L_s} = E_{PK_{MU}} \begin{pmatrix} p_{11} \cdot y_1 & p_{12} \cdot y_2 & p_{13} \cdot y_3 \\ p_{21} \cdot y_1 & p_{22} \cdot y_2 & p_{23} \cdot y_3 \\ p_{31} \cdot y_1 & p_{32} \cdot y_2 & p_{33} \cdot y_3 \end{pmatrix} \tag{5}$$

Algorithm 3 for MSP selection phase

1. MU generates invertible matrices P of size 3×3 , Q of size $n \times n$
2. MU conceals its location $\{x_1, x_2, x_3\}$ in P^{-1} and its attributes $\{u_1, u_2, u_3, \dots, u_n\}$ in Q^{-1} to form C_{L_u}, C_{A_u} using equations 3 and 4, respectively
3. MU encrypts its location in $E_{L_u} \leftarrow E_{PK_{MU}} \left(\sum_{i=1}^3 x_i^2 \right)$, and attributes in $E_{A_u} \leftarrow E_{PK_{MU}} \left(\sum_{i=1}^n u_i^2 \right)$
4. MU generates signature
 $H \leftarrow h_{MS} \left(E_{PK_{MU}}(P), E_{PK_{MU}}(Q), N, PK_{MU} \right)$, and sends it along with $C_{L_u}, E_{L_u}, C_{A_u}, E_{A_u}, N, E_{PK_{MU}}(P), E_{PK_{MU}}(Q)$ to CS
5. CS generates signature
 $H_1 \leftarrow h_{PR_{CS}} \left(E_{PK_{MU}}(P), E_{PK_{MU}}(Q), PK_{MU}, H, N \right)$, and broadcasts $E_{PK_{MU}}(P), E_{PK_{MU}}(Q), PK_{MU}, N, H, H_1$
6. MSP checks integrity of $E_{PK_{MU}}(P), E_{PK_{MU}}(Q), PK_{MU}, H, N$ using H_1, P_{CS} , and then the integrity of $E_{PK_{MU}}(P), E_{PK_{MU}}(Q), PK_{MU}, N$ using H, MS' , and stops on unsuccessful integrity
7. MSP generates private and public key pair $PR_{MSP}, PK_{MSP} \leftarrow g^{PR_{MSP}}$
8. MSP conceals its location in $C_{L_s} (= (C_{L_{s1}}, C_{L_{s2}}, C_{L_{s3}}))$ in $C_{L_{si}} \leftarrow \prod_{j=1}^3 E_{PK_{MU}}^{y_j}(p_{ji})$ ($1 \leq i \leq 3$) and attributes in $C_{A_{si}} \leftarrow \prod_{j=1}^n E_{PK_{MU}}^{a_j}(q_{ji})$ ($1 \leq i \leq n$)
9. MSP encrypts its location in $E_{L_s} \leftarrow E_{PK_{MU}} \left(\sum_{i=1}^3 y_i^2 \right)$ and attributes in $E_{A_s} \leftarrow E_{PK_{MU}} \left(\sum_{i=1}^n a_i^2 \right)$
10. MSP computes the ciphertext
 $T \leftarrow E_{PK_{MU}} \left(PK_{MSP} \parallel \sum_{i=1}^3 y_i^2 \parallel \sum_{i=1}^3 y_i \right)$
11. MSP encrypts $C_s \leftarrow E_{P_{CS}}(C_{L_s}, E_{L_s}, C_{A_s}, E_{A_s}, N, T)$, and sends C_s to CS
12. CS decrypts C_s to obtain $C_{L_s}, E_{L_s}, C_{A_s}, E_{A_s}, N, T$
13. CS compares N received from MU with the N received from MSP and if they differ then discards the MSP response.
14. CS obtains $E_{PK_{MU}}(d_1^2)$ using equations 7 and 8, and obtains $E_{PK_{MU}}(d_2^2)$ using equations 9 and 10
15. CS computes $E_{PK_{MU}}(d_1^2 + d_2^2)$ using equation 11, and $E_{PK_{MU}}(F_s)$ using equation 12
16. CS sends tuple $\{E_{PK_{MU}}(d_1^2 + d_2^2), E_{PK_{MU}}(d_2^2), T\}$ separately for each MSP, and $E_{PK_{MU}}(F_s)$ to MU
17. MU decrypts F_s
18. MU selects one unverified tuple i.e., $\{E_{PK_{MU}}(d_1^2 + d_2^2), E_{PK_{MU}}(d_2^2), T\}$ received in Step 16, and decrypts $d_1^2 + d_2^2$
If $F_s = d_1^2 + d_2^2$ **Then**
 MU decrypts $d_2^2, PK_{MSP} \parallel \sum_{i=1}^3 y_i^2 \parallel \sum_{i=1}^3 y_i$
 If $(F_s - d_2^2) = \left(\sum_{i=1}^3 x_i^2 - \sum_{i=1}^3 2x_i y_i + \sum_{i=1}^3 y_i^2 \right)$ **Then**
 MU computes $E_{PK_{MSP}}(x_1, x_2, x_3, N)$, and
 sends this to selected MSP
19. **If** MU did not send $E_{PK_{MSP}}(x_1, x_2, x_3, N)$ to any MSP and there is at least one unverified tuple **Then**
 Repeat from Step 18.

$$C_{A_s} = E_{PK_{MU}} \begin{pmatrix} q_{11} \cdot v_1 & q_{12} \cdot v_2 \cdot v_3 \cdots & q_{1n} \cdot v_n \\ q_{21} \cdot v_1 & q_{22} \cdot v_2 \cdot v_3 \cdots & q_{2n} \cdot v_n \\ \vdots & & \\ q_{n1} \cdot v_1 & q_{n2} \cdot v_2 \cdot v_3 \cdots & q_{nn} \cdot v_n \end{pmatrix} \quad (6)$$

MSP then encrypts its location and attributes, and uploads these along with N to CS. CS decrypts these and compares the decrypted value of N with the value of N received from MU in current session. If this verification is unsuccessful, CS discards this particular MSP response. For every valid MSP response, CS separately computes the encrypted square distance between MU and MSP locations, and squared attribute distance between MU and MSP using equations 7, 8, 9 and 10.

$$\begin{aligned} C_{L_u} \cdot C_{L_s} &= P^{-1} \cdot L_u \cdot E_{PK_{MU}}(P \cdot L_s) \\ &= E_{PK_{MU}}(P^{-1} \cdot L_u \cdot P \cdot L_s) \\ &= E_{PK_{MU}}(L_u \cdot P^{-1} \cdot P \cdot L_s) \\ &= E_{PK_{MU}}(L_u \cdot L_s) \\ &= E_{PK_{MU}} \left(\sum_{k=1}^3 (x_k \cdot y_k) \right) \end{aligned} \quad (7)$$

$$\begin{aligned} &E_{L_u} \cdot E_{L_s} \cdot E_{PK_{MU}}^{-2}(L_u \cdot L_s) \\ &= E_{PK_{MU}} \left(\sum_{i=1}^3 x_i^2 \right) \cdot E_{PK_{MU}} \left(\sum_{i=1}^3 y_i^2 \right) \cdot E_{PK_{MU}}^{-2} \left(\sum_{k=1}^3 (x_k y_k) \right) \\ &= E_{PK_{MU}} \left(\sum_{i=1}^3 (x_i - y_i)^2 \right) \\ &= E_{PK_{MU}}(|L_u - L_s|^2) = E_{PK_{MU}}(d_1^2) \end{aligned} \quad (8)$$

$$\begin{aligned} &C_{A_u} \cdot C_{A_s} \\ &= Q^{-1} \cdot A_u \cdot E_{PK_{MU}}(Q \cdot A_s) \\ &= E_{PK_{MU}}(Q^{-1} \cdot A_u \cdot Q \cdot A_s) \\ &= E_{PK_{MU}}(A_u Q^{-1} \cdot Q \cdot A_s) \\ &= E_{PK_{MU}}(A_u \cdot A_s) \\ &= E_{PK_{MU}} \left(\sum_{k=1}^n (u_k \cdot v_k) \right) \end{aligned} \quad (9)$$

$$\begin{aligned}
& E_{A_u} \cdot E_{A_s} \cdot E_{PK_{MU}}^{-2}(A_u \cdot A_s) \\
&= E_{PK_{MU}}\left(\sum_{i=1}^n u_i^2\right) \cdot E_{PK_{MU}}\left(\sum_{i=1}^n v_i^2\right) \cdot E_{PK_{MU}}^{-2}\left(\sum_{k=1}^n (u_k v_k)\right) \\
&= E_{PK_{MU}}\left(\sum_{i=1}^n (u_i - v_i)^2\right) \\
&= E_{PK_{MU}}(|A_u - A_s|^2) = E_{PK_{MU}}(d_2^2)
\end{aligned} \tag{10}$$

Then, for every MSP response, $E_{PK_{MU}}(d_1^2)$ and $E_{PK_{MU}}(d_2^2)$ are separately combined by CS to form encrypted summation of d_1^2 and d_2^2 using equation 11. CS also computes the encrypted MSP selection information $E_{PK_{MU}}(F_s)$ for the minimum $d_1^2 + d_2^2$ value using appropriate homomorphic operation through the process similar to [4] illustrated in equation 12.

$$E_{PK_{MU}}(d_1^2) \cdot E_{PK_{MU}}(d_2^2) = E_{PK_{MU}}(d_1^2 + d_2^2) \tag{11}$$

$$\begin{aligned}
& E_{PK_{MU}}(F_s) = \\
& E_{PK_{MU}}(\text{Min}\{(d_1^2 + d_2^2)_{MSP_1}, \dots, (d_1^2 + d_2^2)_{MSP_m}\})
\end{aligned} \tag{12}$$

Next, CS sends the encrypted distance information for each MSP and the encrypted MSP selection information to MU. Upon receiving these, MU selects one unverified tuple $\{E_{PK_{MU}}(d_1^2 + d_2^2), E_{PK_{MU}}(d_2^2), T\}$ and decrypts $d_1^2 + d_2^2$. If the decrypted $d_1^2 + d_2^2$ value matches with the value of the decrypted F_s , MU further verifies that the response from CS is not forged, by decrypting d_2^2 , $PK_{MSP}||\sum_{i=1}^3 y_i^2||\sum_{i=1}^3 y_i$, and comparing $(F_s - d_2^2)$ with $(\sum_{i=1}^3 x_i^2 - \sum_{i=1}^3 2x_i y_i + \sum_{i=1}^3 y_i^2)$. If it matches, MU encrypts and sends its location and nonce value through $E_{PK_{MSP}}(x_1, x_2, x_3, N)$ to the selected MSP. The selected MSP decrypts the MU location (x_1, x_2, x_3) , and nonce value N . MSP then attends the MU after comparing the currently received N with the nonce N received previously. If the verification of the tuple selected is unsuccessful and there are still one or more unverified tuples available, MU repeats this for another unverified tuple. Finally, MU stops if either it discovered a correct tuple and sends its encrypted location information to the corresponding MSP or there is no unverified tuple available. Fig. 3 gives sketch of emergency detection and MSP selection.

4 Security and Privacy Analysis

The proposed framework has been formally verified through appropriate random oracle model for its privacy. Also, we have analyzed the robustness of the proposed framework in terms of security and privacy by simulating the

Fig. 3 Medical Emergency Detection and MSP Selection

proposed framework using the well accepted formal verification Tool ProVerif [23], [24]. Moreover, We informally verify various other known attacks and checks the stability of the proposed framework against these attacks. Prior to the analysis, we introduce the threat model, which we have considered for the formal security and privacy verification of the proposed framework.

4.1 Threat Model

In the threat model, TA and CS are assumed to be honest but curious. Therefore, their adversarial objective is to know the medical status and/or location of the user. However, both TA and CS will follow the protocol rules. Adversary is capable of performing the following:

- Can eavesdrop the communication link between different entities.

- Can intercept one or more messages over one or more communication links, and try to obtain any sensitive information.

4.2 Privacy Analysis through Random Oracle Model

In this section, we use a random oracle model [20], [21] to scrutinize the privacy of the proposed framework. In this analysis, a random oracle $\mathcal{O}$ has been defined that can execute the following functionalities:

- $\text{Extract}(i, m)$: This function models $\mathcal{O}$ to access a protocol instance i , and subsequently obtains the message $m = (m_1 \| m_3 \| m_6 \| m_9)$, where m_1 consists of the encrypted id values that are created after reading the PHIs using MDRQ tree in the protocol instance i . m_3 consists of encrypted matrices P, Q . m_6 consists of the concealed PHI and location information of MU. m_9 consists of the ciphertext for location of MU, which is encrypted using the public key of the scheduled MSP.
- $\text{GetPHI}(m_1, m_2)$: This function models $\mathcal{O}$ to compute any one PHI value in m_2 for the protocol instance i using the ciphertext m_1 and the MDRQ tree used in i .
- $\text{GetMatrixP}(m_3, m_4)$: This function models $\mathcal{O}$ to compute the matrix P generated in the protocol instance i in m_4 using the ciphertext m_3 .
- $\text{GetMatrixQ}(m_3, m_5)$: This function models $\mathcal{O}$ to compute the matrix Q generated in the protocol instance i in m_5 using the ciphertext m_3 .
- $\text{GETLOCP}(m_6, m_4, m_7)$: This function models $\mathcal{O}$ to compute the location information of MU in m_7 for the protocol instance i using m_6, m_4 through matrix multiplication operation.
- $\text{GetPHIQ}(m_6, m_5, m_8)$: This function models $\mathcal{O}$ to compute any one PHI value of MU in m_8 for the protocol instance i using m_6, m_5 through matrix multiplication operation.
- $\text{GetLOC}(m_9, m_{10})$: This function models $\mathcal{O}$ to compute the location information of MU in m_{10} using the ciphertext m_9 .

The random oracle $\mathcal{O}$ is assumed to efficiently accomplish any computational task given a protocol instance i . Adversary $\mathcal{A}$ can use $\mathcal{O}$ to obtain the PHI readings and/or the location information of MU used in the protocol instance i .

Theorem 1: Proposed framework preserved privacy of MU.

Proof: We design a game $\mathcal{G}$ for the adversary $\mathcal{A}$. In this game, $\mathcal{A}$ will be given a set of completed protocol instances $\mathcal{I}$, and the MDRQ tree used in $\mathcal{I}$. There is an upper bound t for the duration of $\mathcal{G}$, during which $\mathcal{A}$ has to finish the game and return the values of some parameters to the judges $\mathcal{J}$. $\mathcal{A}$ can be assumed as winner if it can successfully compute any correct PHI value and/or the location information of the MU used in i . $\mathcal{A}$ exploits the functionalities of $\mathcal{O}$, and randomly selects a protocol instance $i \in \mathcal{I}$. The oracle $\mathcal{O}$ is invoked and given the protocol instance i and the MDRQ tree used in i . $\mathcal{O}$ executes its functionalities using Algorithm 4. In Algorithm 4, accessing the protocol instance i , $\mathcal{O}$ first executes $\text{Extract}(i, m)$ to obtain various

encrypted values used in the protocol instance i . It then separates the values collected in m using the parameters m_1, m_3, m_6 and m_9 . $\mathcal{O}$ then executes the function $\text{GetPHI}(m_1, m_2)$ using the parameter m_1 , and obtains the id values. It then computes the PHI values using the id values and the MDRQ tree. After this, it randomly selects a PHI value from the set of PHI values it just computed, and keeps the selected PHI value in m_2 . The oracle $\mathcal{O}$ executes $\text{GetMatrixP}(m_3, m_4)$ to obtain the matrix P' , which is then kept in the parameter m_4 , and similarly, it keeps the computed matrix Q' in the parameter m_5 after executing $\text{GetMatrixQ}(m_3, m_5)$.

Algorithm 4 executed by $\mathcal{O}$

Input: A protocol instance $i \in \mathcal{I}$ and the MDRQ tree

Output: Probable PHI values in parameters m_5, m_8 and location in parameters m_7, m_9

1. Selects a protocol instance $i \in \mathcal{I}$ randomly
 2. Executes $\text{Extract}(i, m)$
 3. Executes $\text{GetPHI}(m_1, m_2)$
 4. Executes $\text{GetMatrixP}(m_3, m_4)$
 5. Executes $\text{GetMatrixQ}(m_3, m_5)$
 6. Executes $\text{GETLOCP}(m_6, m_4, m_7)$
 7. Executes $\text{GetPHIQ}(m_6, m_5, m_8)$
 8. Executes $\text{GetLOC}(m_9, m_{10})$
-

It then executes $\text{GETLOCP}(m_6, m_4, m_7)$ to obtain the location information x'_1, x'_2, x'_3 . In this function, $\mathcal{O}$ first extracts the concealed location from m_6 , and then obtains x'_1, x'_2, x'_3 in m_7 after multiplying the concealed location and matrix P' in m_4 . In similar fashion, $\mathcal{O}$ executes $\text{GetPHIQ}(m_6, m_5, m_8)$. In this function, $\mathcal{O}$ extracts concealed PHIs from m_6 , and then obtains the PHIs used in i after multiplying the concealed PHIs and the matrix Q' in m_5 . It then randomly selects any one PHI value and keeps this in m_8 . After this, $\mathcal{O}$ executes $\text{GetLOC}(m_9, m_{10})$, and obtains the location information x''_1, x''_2, x''_3 of MU, which is then kept in m_{10} . The computed values m_2, m_7, m_8, m_{10} along with the id of the protocol instance i are returned by the adversary $\mathcal{A}$. Receiving these parameters, $\mathcal{J}$ execute Algorithm 5 to decide the fate of $\mathcal{A}$. Algorithm 5 uses the following function.

1. $\text{Retrieve}(i, m_{11}, m_{12})$: This function is used to retrieve the original location information x_1, x_2, x_3 in m_{11} , and the original PHI readings of MU in m_{12} for the protocol instance i . $\mathcal{J}$ retrieve these parameters from the database, which recorded the history of $\mathcal{I}$.

Algorithm 5 executed by the judges of $\mathcal{G}$ **Input:** Values of the parameters m_2, m_7, m_8, m_{10}, i returned by $\mathcal{A}, t, \mathcal{I}$ **Output:** *Winner* or *Not a Winner*

-
1. **If** Time consumed by $\mathcal{A}$ is more than t **Then**
 2. Return *Not a Winner*
 3. **Else** execute $\text{Retrieve}(i, m_{11}, m_{12})$
 4. **If** $m_5 \in m_{12}$ **OR** $m_8 \in m_{12}$ **OR** $m_7 \in m_{11}$ **OR** $m_{10} \in m_{11}$ **Then**
 5. Return *Winner*
 6. **Else**
 7. Return *Not a Winner*
-

Algorithm 5 receives the parameters m_2, m_7, m_8, m_{10}, i returned by $\mathcal{A}$ along with the set $\mathcal{I}$ of the completed protocol instances and the maximum duration of $\mathcal{G}$, i.e., t . In Algorithm 5, $\mathcal{J}$ declares that $\mathcal{A}$ is *NOT a Winner* if duration of $\mathcal{G}$ played by $\mathcal{A}$ exceeds t . Otherwise, $\mathcal{J}$ executes $\text{Retrieve}(i, m_{11}, m_{12})$. In this function, $\mathcal{J}$ extracts the location m_{12} and the set of PHIs m_{11} from the database using the value of i . $\mathcal{J}$ declares $\mathcal{A}$ as the *Winner* if it finds any of values m_2, m_7, m_8, m_{10} returned by $\mathcal{A}$ in either in m_{11} or in m_{12} . Therefore $\mathcal{A}$ may be declared as the Winner if $\mathcal{O}$ correctly compute the value of the parameters m_2, m_7, m_8, m_{10} , i.e., the location or PHI values of MU. Probability that $\mathcal{O}$ can correctly compute at least one correct PHI value in m_2 depends on the probability of determining correct id values from the given ciphertexts C'_1 , which requires $\mathcal{O}$ to solve hash function. According to Birthday Attack assumption, an id value can be computed from a given C'_1 with probability $(\frac{1}{2})^{\frac{k}{2}}$, where k is the length of hash value in bits. And since there is the need to find l number of id values to compute correct PHI using MDRQ tree, the probability of computing one correct PHI is $(\frac{1}{2})^{\frac{lk}{2}}$. For sufficiently large value of k , this quantity is negligible. For computing correct PHI in m_8 , $\mathcal{O}$ has to correctly compute the matrix Q from m_3 , which is encrypted using the public key PK_{MU} . Hence, the success depends on the hardness assumption of the encryption scheme used for encrypting matrix Q . The similar argument can be used in case of m_7 and m_{10} . Assuming the hardness property of the encryption schemes, the probability that none of these values cannot be guessed is $(1 - 2^{-d_1}) \times (1 - 2^{-d_2}) \times (1 - 2^{-d_3})$, where d_1 is the total length of all the values in matrix Q , d_2 is total length of all the values in matrix P , and d_3 is the total length of the location parameters x_1, x_2, x_3 . Therefore, the probability that $\mathcal{O}$ can successfully compute any one parameter among m_2, m_7, m_8, m_{10} correctly is $P(S) = 1 - [(1 - 2^{-\frac{lk}{2}}) \times (1 - 2^{-d_1}) \times (1 - 2^{-d_2}) \times (1 - 2^{-d_3})]$. For sufficiently large value of k, d_1, d_2, d_3 , this quantity $P(S) \leq \epsilon$ will be negligible for sufficiently small value of ϵ . Hence, the proposed scheme preserves the privacy of MU.

4.3 Formal Security Verification using ProVerif

In order to prove that a protocol is correct, the symbolic model, also known as Dolev-Yao model [22] is usually used. This symbolic model is an abstract model, which helps to build automatic verification tools. Many such tools are used in practice. ProVerif [25], AVISPA [26], and Tamarin [27], etc. are few such examples of symbolic models. We have used ProVerif to prove that our proposed framework is secure. The verification summary of Initialization phase and Emergency Detection phase is shown in the following rectangular box:

```
(* Initialization and Emergency Detection phases *)
(*secure and private channels*)

-----
Verification summary:

Query not attacker(N[]) is true.
The nonce hasn't been revealed to the attacker
-----
*)
```

The verification summary of MSP selection phase is shown in the following rectangular box:

```
(* MSP Selection Phase *)

-----
Verification summary:

Query not attacker(x[]) is true. // X IS THE LOCATION ATTRIBUTE
Query not attacker(u[]) is true. // U IS THE PHYSICAL ATTRIBUTE
-----
*)
```

The results, shown in the above boxes, conclude that the proposed framework satisfy the security properties that we aim to achieve.

4.4 Informal Security Analysis of the Proposed Framework

Here, we informally analyze the security and privacy of the proposed framework with the assumption that TA and CS are curious. We prove that our proposed framework fulfills the security and privacy requirements through the following claims and their proofs.

Claim 1: Proposed framework preserves privacy of MU

Proof: In medical emergency, the framework should not reveal any private information of MU. However, the location of MU can be revealed to the selected MSP only after completion of the MSP selection task. MU embeds its current PHI values in C'_1 through the id values generated using MDRQ tree. C'_1 uses hash function to hide these id values. Due to the pre-image resistance property of hash functions, any adversary $\mathcal{A}$ including CS and TA cannot retrieve the id values. Thus, PHI information of MU is safe. PHI and location information of MU are concealed in matrix P, Q , and also they are encrypted using the public key of MU, $PK_{MU} = g^\beta$ (Equations 3, 4, 5, 6) to obtain $C_{L_u}, C_{A_u}, C_{L_s}, C_{A_s}$. We see that, $\mathcal{A}$ cannot obtain the location of MU from C_{L_u} due to the hardness of solving the integer factorization problem. Similarly, $\mathcal{A}$ cannot obtain the PHI values of MU from C_{A_u} . $\mathcal{A}$ may also try to obtain the location and/or PHI values from the ciphertexts C_{L_s} and/or C_{A_s} . However, based on the hardness assumption of cryptanalyzing the encryption algorithm, there is negligible probability that $\mathcal{A}$ can obtain the location and PHI values exploiting the ciphertexts C_{L_s} and/or C_{A_s} . Similarly, $\mathcal{A}$ cannot obtain the attribute and location information of MSPs. TA may try to correctly generate d_1^2 to find the distance between MU and MSPs by guessing the location of MU, (x_1, x_2, x_3) . The probability of correct guess is $2^{-b_1} \cdot 2^{-b_2} \cdot 2^{-b_3} = 2^{-(b_1+b_2+b_3)}$, where b_1, b_2, b_3 are sizes of x_1, x_2, x_3 respectively. Therefore, the probability of guessing the MU location is negligible. Thus, the proposed framework preserves privacy of MU.

Claim 2: The proposed framework preserves integrity requirement.

Proof: In the emergency detection phase, CS verifies the integrity of the received information from MU using equation 1. Suppose, the adversary $\mathcal{A}$ is trying to change C'_1 into C''_1 to wrongly predict the emergency of MU. Therefore, it has to generate correct C'_2 value for which the integrity verification would be correct. Since $\mathcal{A}$ does not have correct secret key β , it can try to generate $C''_2 \leftarrow g^{\frac{1}{C''_1 + \beta' + N + I_i}}$ using some guessed secret key β' . Moreover, $\mathcal{A}$ has to use the same public key PK_{MU} . Otherwise, the emergency prediction will not be for the actual MU, and which in turn, will not fulfill the objective of $\mathcal{A}$. However, if $\beta \neq \beta'$, for same public key PK_{MU} , $g^{\frac{1}{C''_1 + \beta' + N + I_i}} \neq g^{\frac{1}{C'_1 + \beta + N + I_i}}$. Therefore, the integrity verification will be failed. $\mathcal{A}$ may try to guess correct β . However, the probability of correct guess is negligible considering β is sufficiently large. Thus, $\mathcal{A}$ cannot be successful to break the integrity. Similarly, in MSP selection phase, the MSP and MU verify integrity of the messages communicated between them through CS. In all these cases, $\mathcal{A}$ without having the secret key β , cannot break the integrity. Therefore, the proposed framework preserves the integrity requirement.

Claim 3: The proposed framework is secure against replay attack

Proof: The adversary $\mathcal{A}$ can try to replay a message. For example, an MSP can attempt to replay the value of an old C_s for which the MSP was selected in an old session for the same user. However, the CS will reject this message

as the value of N used in this message is not same as the value used in the current session. In emergency detection phase, old alert can be replayed by an outside attacker, which will be rejected by MU as the value of N used in the replayed message will not be the same with the value of N used in the current session. Similarly, any replayed message will be rejected by the respective entities responsible for verification of the messages before their acceptance. Thus, the proposed framework is secure against replay attack.

Claim 4: The proposed framework is secure against message forgery attack by the adversaries including CS

Proof: In message forgery attack, the intended recipient receives a fabricated message instead of original message. As an example, the adversaries including CS may try to create a forged message $E_{PK_{MU}}(P)$, $E_{PK_{MU}}(Q)$, PK_{MU} , N , H , H_1 and broadcast this message to all MSPs. However, this message will not be accepted. Because the outside adversaries neither have the master secret key MS nor the secret key of CS. Therefore, the advantage in generating the signatures H, H_1 are negligible. Having the secret key, CS can generate the valid signature H_1 . However, it cannot generate the valid signature H as it does not have the secret key MS . Therefore, the proposed framework provides protection against forgery attack.

Claim 5: The proposed framework is secure against power leakage attack

Proof: In this type of attacks, attacker attempts to make MU unnecessarily engage in computations that drains the limited power of MU. As the battery power of MUs is limited, malicious agents may try to send fake Emergency alerts in the Emergency Detection Phase to drain the power by involving MU in unnecessary computations. In the proposed framework, MU uses nonce value N to detect any such attempt. MU sends this value to CS after encryption using the secret key of CS. Therefore, only the CS can decrypt this value and use this value to send the alert message. Since the outside adversaries do not have the secret key, they cannot decrypt N . These adversaries can guess a value and send that value as the alert message. However, the probability of guessing a value such that the corresponding value of N will be correct is negligible. Therefore, MU will reject the fake alerts, and will not perform any further computations. This can prevent the power leakage due to fake alerts.

5 Performance Analysis of Proposed Framework

In this section, we analyze the performance of the proposed framework in terms of security, computation and communication requirement. We then compare the performance of our proposed framework with the framework proposed in [4]. We use only [4] for the comparison because this is the lone existing work, which handles the MSP selection problem.

5.1 Security and Privacy

A summary of comparative security features present in the current work and in [4] is given in Table 2.

Table 2 Security and Privacy Comparison

	a	b	c	d	e
Proposed framework	Yes	Yes	Yes	Yes	Yes
Protocol in [4]	No	Yes	Yes	No	No

a: Confidentiality, b: Integrity, c: Replay attack, d: Power Leakage, e: Message forgery.

From this comparison, we can conclude that the proposed framework in this paper has outperformed the work in [4].

5.2 Computation

We first compute the number of operations used by the entities present in the proposed framework in this paper. Table 3 illustrates this computation.

Table 3 Number of various operations used in the proposed framework

Operation	MU	MSP	CS	TA
$\leftarrow$	$21n + 5$	$n^2 + 8n$	1	1
+	$32n + 1$	$n^2 + n$	—	n
$\times$	$n^2 + n + 12$	$n^3 + n^2 + 13n$	$20n$	$10n$
$\div$	$10n$	—	—	—
$\oplus$	—	—	$10n$	—
$\wedge$	$20n$	—	—	$20n + 2$
compare	$\frac{n}{2} + 2$	—	$25n$	—
$\parallel$	—	$2n$	—	—
random no.	1	—	—	3
matrix gen.	$n^2 + 3^2$	—	—	—
matrix inv.	$n^2 + 3^2$	—	—	—
key-pair gen.	1	n	1	—
hash	$10n + 1$	n	$10n$	$20n$
$E_k()$	$10n + 3$	$n^2 + 5n$	2	—
$D_k()$	4	1	1	—
pairing func.	—	—	$20n$	$10n$
homo. $E_k()$	2	$2n$	—	—
homo. $\times$	2	—	2	—
homo. scalar $\times$	1	—	1	—
MDRQ tree	1	—	—	1

Table 3 shows that MU needs to perform many computations in case of an emergency. However, this computations have ensured the security and privacy of the proposed framework.

Fig. 4 Time taken during Emergency Detection Phase

5.2.1 Phase wise Time Comparison

Based on the number of operations performed by every component, we have computed the execution time of the proposed framework using Intel Core 2, 1.83 GHz CPU. We plot the time taken by different phases of our proposed framework and the scheme proposed in Yu’s framework [4]. The time comparison for emergency detection phase, and MSP selection phase are shown in Fig. 4 and 5 respectively. To ensure more security and privacy, computation in MSP selection phase in the proposed framework is higher compared to the scheme proposed in [4], as it is illustrated through the graph in Fig. 5. However, in the emergency detection phase, our framework performs much better than the scheme in [4], as illustrated in Fig. 4.

5.3 Communication

In this section, comparison of number of communications that take place for MU has been provided. Assumptions on number of bits of different elements, that are used for communication, are as shown in the Table 4. The number of communications for MU in our proposed framework and the scheme in [4] are shown in Fig. 6. We observe that our proposed framework always takes much lower communication overhead for MU than its counterpart in Yu’s framework [4].

Fig. 5 Time taken during MSP selection Phase

Table 4 Assumptions on size of different entities used in communication

Element	Size in bits
PHI id	32
MDRQ tree node	32
PUID[4]	128
Random numbers	1024
I_i	32
δ_i	1024
$Normal$	128
$\$$	128
C_1, C_2, C'_1, C'_1	128
Matrix elements	32
Group elements	32
Key	1024
Ciphertext	1024
Plaintext	1024
Hash output	128

6 Conclusion

In this work, we have identified that a trusted third party is being utilized to select an appropriate MSP for a medical user in medical emergency. However, the internal agents of the third party may be curious and hence the privacy of MU can be leaked through this component. Moreover, if this component is failed, the entire system will suffer one point failure problem. Therefore, we have proposed a privacy preserved MSP selection framework, which can avoid these problems. Analysis in this paper reveals that the proposed framework

Fig. 6 Number of Communications done by MU

outperforms the existing work in terms of security and privacy. However, in our proposed framework, the devices implanted in MU needs to perform more computations. As the future work, we will try to reduce this computation overhead.

7 Compliance with Ethical Standards

Funding: This study was funded by SERB, DST, Govt. of India (project grant number EEQ/2020/000039).

Conflict of Interest: Author Anupam Pattanayak declares that he has no conflict of interest. Author Mou Dutta declares that she has no conflict of interest. Author Subhasish Dhal declares that he has no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith and A. Jamalipour, "Wireless Body Area Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp 1658–1686, Mar 2014.
2. B. Latré, B. Bart, I. Moerman, C. Blondia, and P. Demeester, "A Survey on Wireless Body Area Networks," *Wirel. Netw.*, vol. 17, no. 1, pp 1–18, Jan 2011.
3. A. Pattanayak, and S. Dhal, "Cloud Enabled Body Area Network," *Cloud Network Management*, Chapman and Hall/CRC, pp 67–85, 2020.

4. W. Yu, Z. Liu, C. Chen, B. Yang, and X. Guan, "Privacy-preserving design for emergency response scheduling system in medical social networks," *Peer-to-Peer Networking and Applications*, vol. 10, no. 2, pp 340–356, 2017.
5. X. Lai, Q. Liu, X. Wei, W. Wang, G. Zhou, and G. Han, "A Survey of Body Sensor Networks," *Sensors*, vol. 13, no. 5, pp 5406–5447, 2013.
6. M. Chen, S. Gonzalez, A. Vasilakos, C. Athanasios, L. Huasong, and C. Victor, "Body Area Networks: A Survey," *Mob. Netw. Appl.*, vol. 16, no. 2, pp 171–193, Apr 2011.
7. T. Maitra, and S. Roy, "Research Challenges in BAN Due to the Mixed WSN Features: Some Perspectives and Future Directions," *IEEE Sensors Journal*, vol. 17, no. 17, pp 5759–5766, Sep 2017.
8. X. Liang, R. Lu, L. Chen, X. Lin, and X. Shen, "PEC: A privacy-preserving emergency call scheme for mobile healthcare social networks," *Journal of Communications and Networks*, vol. 13, no. 2, pp 102–112, 2011.
9. K. Zhang, X. Liang, M. Barua, R. Lu, and X. S. Shen, "PHDA: A priority based health data aggregation with privacy preservation for cloud assisted WBANs," *Information Sciences*, vol. 284, pp. 130–141, 2014.
10. Q. Huang, L. Wang, and Y. Yang, "Secure and privacy-preserving data sharing and collaboration in mobile healthcare social networks of smart cities," *Security and Communication Networks*, vol. 2017, 2017.
11. J. Sun, Y. Fang, and X. Zhu, "Privacy and emergency response in e-healthcare leveraging wireless body sensor networks," *IEEE Wireless Communications*, vol. 17, no. 1, pp 66–73, 2010.
12. Chun-Ta Li, Cheng-Chi Lee, and Chi-Yao Weng, "A secure cloud-assisted wireless body area network in mobile emergency medical care system," *Journal of Medical Systems*, vol. 40, no. 5, pp 117, 2016.
13. S. Chiou, and Z. Liao, "A Real-Time, Automated and Privacy-Preserving Mobile Emergency-Medical-Service Network for Informing the Closest Rescuer to Rapidly Support Mobile-Emergency-Call Victims," *IEEE Access*, vol. 6, pp 35787–35800, 2018.
14. H. Lin, J. Shao, C. Zhang and Y. Fang, "CAM: Cloud-Assisted Privacy Preserving Mobile Health Monitoring," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp 985–97, Jun 2013.
15. E. Shi, J. Bethencourt, T. Chan, D. Song, and A. Perrig, "Multi-dimensional range query over encrypted data", In *Proceedings of IEEE Symposium on Security and Privacy (SP'07)*, pp 350–364, 2007.
16. B. Hore, S. Mehrotra, M. Canim, M. Kantarcioglu, "Secure multidimensional range queries over outsourced data", In *The VLDB Journal*, Springer, vol. 21, no. 3, pp 333–358, 2012.
17. M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers", In *Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, pp 24–43, 2010.
18. C. Gentry, and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme", In *Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, pp 129–148, 2011.
19. CA. Dhote, "Homomorphic Encryption for Security of Cloud Data", In *Procedia Computer Science*, Elsevier, vol. 79, no. 3, pp 175–181, 2016.
20. R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited", In *Journal of the ACM (JACM)*, ACM, vol. 51, no. 4, pp 557–594, 2004.
21. N. Kobitz, and A J Menezes, "The random oracle model: a twenty-year retrospective", In *Designs, Codes and Cryptography*, Springer, vol. 77, no. 2, pp 587–610, 2015.
22. D. Dolev and A. Yao, "On the security of public key protocols", In *IEEE Transactions on Information Theory*, pp 198–208, 1983.
23. B. Blanchet, "An Efficient Cryptographic Protocol Verifier Based on Prolog Rules", In *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW-14)*, pp 82–96, 2001.
24. B. Blanchet, M. Abadi, and C. Fournet, "Automated verification of selected equivalences for security protocols", In *Journal of Logic and Algebraic Programming 75(1)*, pp 3–51, 2008.

-
25. B. Blanchet, "Automatic verification of security protocols in the symbolic model: The verifier proverif", In *Foundations of security analysis and design VII*, pp 54–87, 2013, Springer.
 26. L. Viganò, "Automated Security Protocol Analysis With the AVISPA Tool ", In *Electronic Notes in Theoretical Computer Science 155*, pp 61–86, 2006.
 27. S. Meier, B. Schmidt, C Cremers, and D. Basin, "The TAMARIN Prover for the Symbolic Analysis of Security Protocols", In *Proceedings of International Conference on Computer Aided Verification*, pp 696–701, 2013.