Prevention of Known Emergency Attack in Medical Service Provider Selection for Body Area Network User in Real-Time Healthcare

Anupam Pattanayak (anupam.pk@gmail.com)
Indian Institute of Information Technology Guwahati
Subhasish Dhal (subhasis@iiitg.ac.in)
Indian Institute of Information Technology Guwahati

Research Article

Keywords: Smart Healthcare, Security and Privacy, Medical Emergency, Body Area Network, Blockchain

DOI: https://doi.org/10.21203/rs.3.rs-2086485/v2

Additional Declarations: The authors declare no competing interests.

Noname manuscript No.
(will be inserted by the editor)

Prevention of Known Emergency Attack in Medical Service Provider Selection for Body Area Network User in Real-Time Healthcare

the date of receipt and acceptance should be inserted later

Abstract In cloud-assisted body area network (BAN), medical user (MU) is equipped with sensor nodes to measure different physiological health information (PHI). Whenever the PHI reading is found to be abnormal, BAN notifies this medical emergency event to cloud. Cloud broadcasts this medical emergency event to all medical service providers (MSPs) attached to cloud while preserving privacy of MU. There can be multiple MSPs competing for admitting the MU in need of medical treatment. For this, MSP uploads its infrastructural parameters to cloud. However, some MSPs can upload false data to cloud for attracting MU. This is possible when malicious MSP is aware of some medical emergency event and ready to exploit the system to get more business. We classify this attack as Known Emergency Attack. This attack can severely deteriorate the appropriate MSP selection process during emergency. However, as per our knowledge, there is no prior work addressing this problem. We propose a blockchain based framework, which can prevent an MSP from uploading false data and thereby protect from known emergency attack. In response to MU emergency, MSPs send its encrypted attributes to miners in the blockchain network. Miners compute encrypted difference between registered MSP attribute and currently received MSP attribute. If the difference is within a threshold, only then this MSP is considered as a candidate for MSP selection. Security and privacy of proposed framework is verified using Random Oracle Model and AVISPA tool. Computational requirement of proposed framework has also been evaluated.

Keywords Smart Healthcare, Security and Privacy, Medical Emergency, Body Area Network, Blockchain

Address(es) of author(s) should be given

1 Introduction

One desirable feature of a smart society is to provide instant notification of any medical problem of its inhabitants. Moreover, this instant notification must precede by automatic detection of the medical emergency, and be followed by automatic selection of competent medical service provider. This is a very natural expectation amid the present age technological advancements. On the other hand, the MSPs are extending their service horizons to increase revenue. With the enormous technological progress in wearable and implantable wireless sensors, both these visions of smart healthcare and MSPs are being realized through the use of BAN [11], [16]. A very fundamental aim of BAN is to respond against any medical emergency very quickly [21]. However, selection of the most suitable MSP during medical emergency is a vital task as it can ensure proper and effective medical treatment. Failing to select appropriate MSP can lead to inadequate medical treatment and as a consequence, life of the patient is put in a risk. In general, the cloud server (CS) does the automatic selection of MSP among several MSPs attached to it based on the latest infrastructural data uploaded by MSPs. Whenever one or more physiological reading of users are analyzed by the cloud to be abnormal and an emergency is triggered [22], [28]. However, there are some major security and privacy issues that can restrict this technology to be adopted by smart society in real life. A malicious MSP may be aware of some medical emergencies in its surroundings and can try to exploit this information to get more business. It may try to upload false data to get preference from CS in getting MUs to attend them in medical emergency. We have identified this attack and classified this as known emergency attack. Known emergency attack is a very serious issue and no existing work has addressed this issue till date. In this paper, we propose a framework, where a blockchain based solution has been provided to prevent the MSPs from

uploading false data. The proposed solution does not use CS. Blockchain takes the responsibility of maintaining the MSP attributes and selecting the competent MSP when MU experiences medical emergency. There are three phases in the proposed framework: initialization phase, MSP resource registration and update phase, and MSP selection with fake response detection phase. We have used a trusted authority (TA). However, it is used only during the initialization phase and MSP resource registration phase. In response to an MU emergency from BAN, every MSP sends its attributes to all miners (Mi) in the blockchain network. Mi computes the difference between registered MSP attributes and the received MSP attributes in encrypted domain. If the deviation is within a threshold, only then the MSP is taken into consideration. Due to immutable nature of blockchain, data once stored in blockchain network, can never be altered. Use of distributed blockchain network solves the single-point failure problem associated with centralized trusted authority. Further, the blockchain technology ensures that MSP cannot alter the MSP resource status falsely on the blockchain database. Following is the summary of contributions made in this work:

i. It introduces a new kind of attack, in cloud-enabled BAN, called known emergency attack.

ii. It is established that it is safe to use TA for resource registration of MSP, but use of TA for detecting fake status update by MSP is not safe for uninterrupted service of very crucial real-time MSP selection.

iii. Blockchain based real-time MSP selection framework has been proposed to detect false update of MSP resources real-time and thereby prevent known emergency attack.

iv. MU can tune, as per desired security, the threshold δ that determines the maximum allowed difference between registered and broadcast attributes of MSP.

v. Security and privacy analysis of the proposed system has been performed using Random Oracle Model and AVISPA tool.

vi. Computational requirements of different components in the proposed system has been evaluated.

2 Related Works

Several articles such as [2], [6], [7], [11], [12], [16], [18] and [19] have discussed different spheres of BAN. In [29], K. Zhang et al. proposed a privacy-preserved priority-based health data aggregation scheme for cloud-assisted BAN. Q. Huang et al. proposed a secure collaboration and health data sharing scheme in [9]. In [25], J. Sun et al. have introduced a privacy-preserved emergency response system, where the assumption is that, a pre-decided primary physician will be accessing the health records. Chun-Ta Li et al. have discussed a secure cloud-assisted architecture, where predefined medical care providers can access and process medical data of MU in [13]. W. Yu et al. uses multi-dimensional range query (MDRQ) tree to represent PHI range in [28]. In [3], the authors proposed an approach to alert the closest medical staffs through a server for quickly attending the MU in emergency. A cloud-assisted privacy preserved mobile health monitoring system has been proposed by H. Lin et al. in [14]. In [27], the authors have discussed classification of the PHI data, and then filtering false data. Blockchain has been used to record the emergency related medical data as patient moves from one MSP to another in [8]. In [24], the authors have proposed a framework for adopting the blockchain technology to manage emergency response following a design science approach. In [5], the authors used private blockchain to propose a system where the smart contract supports real-time monitoring of MU and medical interventions by sending messages to MU and MSPs. In [23], the authors proposed an access control management of PHI during emergency using smart contracts based on permissioned blockchain hyperledger fabric. In [22], [28], authors have discussed automatic privacy preserving MSP selection issues. However, none of the existing works has discussed how to prevent known emergency attack by malicious MSP.

3 Proposed Framework

Here, it is first stated that it is safe to do the MSP resource registration with TA, but the subsequent MSP resource status checking followed by MSP selection through TA is unsafe for the system. MSPs first register its infrastructural status in blockchain. An MSP can also update its infrastructural status. MU is equipped with necessary infrastructure to continuously monitor its PHI status. If any PHI reading is beyond the normal range, then that event is considered as medical emergency. MU notifies this medical emergency event to MSPs and to the blockchain network. The entire process has been divided into two tasks, as the following.
Task I: Resource registration and update.
Task II: Detection of the fake resource status of MSP.

TA is trusted to carry out task I. TA can be a government entity responsible for enlisting the willing MSPs into the system and register the resources of MSP after careful inspection. This initial registration process can include physical verification of MSP resources by TA. The underlying assumption made here is that, TA possesses infrastructure to physically inspect the resources of MSP and verify them honestly. Similarly, the assumption is that whenever there is any change in resource status of MSP, that is verified by TA

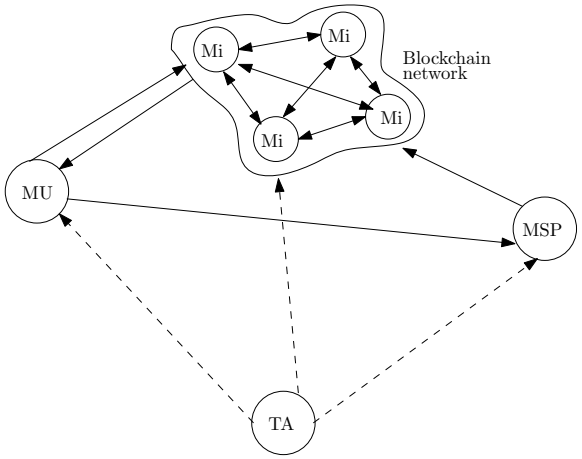

Fig. 1: Communication Model of Proposed System

and updated into blockchain network. TA is trusted but curious. TA will perform the resource verification and registration of these resources honestly. This is similar to the international standard organisation (ISO) certificates awarded by different trusted certification agencies that verify the infrastructure of the client and provide the certification. TA cannot be assigned to carry out the task II. This is because, TA follows centralized architecture. This centralized architecture puts serious vulnerability on the system since it can suffer from single-point failure problem. If TA collapses, then the entire system depending on it will fail. Hence, the task II cannot be carried out by TA. Decentralized blockchain system has been used to carry out the task of detecting fake resource update by malicious MSP to provide uninterrupted MSP selection to attend MU in emergency in real-time healthcare.

The proposed system architecture and its communication model is described in Fig. 1. In this communication model, the secured communication channels are depicted in dotted lines, and communication through insecure channels are depicted in solid lines. In the communication model, there are four different entities: medical user, medical service provider, trusted authority, and miners. There are three phases: initialization phase, MSP resource registration and update phase, and MSP selection with fake response detection phase. The notations used in this work is depicted in Table 1.

Table 1: Notations

Symbol	Meaning
r	Random number
N, N_1	Nonce
$\mathbb{G}$	Additive Group
g	Generator of $\mathbb{G}$
H_i	Hash function
P, Q, R	Invertible matrices
MS, MS'	Private and Public Master key of MUs
β, PK_{MU}	Private and Public key of MU
PR_{TA}, PK_{TA}	Private and Public key of TA
PR_{MSP}, PK_{MSP}	Private and Public key of MSP
$E_K()$	Encryption under key K
$D_K()$	Decryption under key K
m	MU attribute vector
x	MSP attribute during registration
y	MSP attribute sent against emergency
z	MSP location
z_1	MSP location sent against emergency
δ	Threshold value
d_1	Encrypted distance between x, y
d_2	Encrypted distance between m, y
d_3	Encrypted distance between z, z_1

3.1 Brief Overview

TA initializes the system by generating the keys and distributing these keys to other entities through a secure communication channel. When MSP joins into the system, its medical infrastructure resources is physically inspected by TA and this infrastructural detail is uploaded by TA into the blockchain network. This infrastructural detail is written into the blockchain after consensus among the miners. Whenever there is any change in MSP resources, that needs to be physically verified by TA, and the same is reflected in blockchain network. MU is equipped with BAN to monitor its PHI status. MU detects medical emergency whenever there is any PHI data outside the normal range. When MU detects any such medical emergency, it broadcasts this event to MSPs and blockchain. In response to this emergency event, MSPs send its current infrastructure details to the blockchain network. Miners in the blockchain network check if the individual infrastructural parameter of MSP, matches with the record already available in blockchain. If its parameters differ by at most δ, then miners will consider this MSP response as a valid candidate for MSP selection to attend the MU in medical emergency. Value of the parameter δ is preset by MU. MU can determine how much deviation in MSP attribute from what is registered with blockchain, is acceptable to her.

In the initialization phase, system setup is done. This algorithm is presented in algorithm 1. When MU is in medical emergency then the MSP selection and detection of fake response from MSP, if any, is performed through algorithm 3. MSP resource registration with TA is done by algorithm 2.

3.2 Initialization Phase

Algorithm 1 does the system initialization. TA, MU, MSP and Mi all take part in initialization phase. TA initializes the whole system. TA computes its private and public key pair PR_{TA} and PK_{TA}. TA then sends PK_{TA} to MU, MSP and Mi

securely. Next, TA generates master secret key pair: private key MS for MU, public key MS' for MSP and Mi.

Algorithm 1 for Initialization Phase

1. TA computes private and public key pair PR_{TA}, PK_{TA}, and sends PK_{TA} to MU, MSP and Mi securely
2. TA generates master secret key pair: private key MS, public key MS' and shares MS with MU and MS' with MSP and Mi

3.3 MSP Resource Registration and Update Phase

When MSP joins the system for the first time, it needs to register into the system. Also, whenever there is any change in MSP infrastructure, MSP requests for resource registration. However, the time to serve this request may not be instant as TA needs to verify physically the resources before the status gets reflected in blockchain network. The Algorithm 2 takes care of MSP resource registration process. Blockchain miners Mi must be informed about MSP resources in such a way that Mi does not get to know the MSP resource information. For this, MSP requests TA for resource registration, and TA will physically inspect the MSP resources and then encrypts its parameters x using public key of TA. Here, x is MSP resource attribute vector including its location. MSP location is also registered separately as z to detect fabricated location z_1 sent by malicious MSP. Next, TA sends the encrypted MSP attributes to Mi.

Algorithm 2 for MSP Resource Registration and Update Phase

1. MSP requests TA for resource registration or resource update
2. TA physically verifies MSP infrastructure
3. MSP generates invertible matrices Q of order $|x|$ and R of order $|z|$ where x is MSP resource, z is MSP location
4. MSP sends its resource and location information x^2, $Q^{-1}x$, and z^2, $R^{-1}z$ to TA
5. TA requests Mi to add $E_{PK_{TA}}\left(x^2\right)$, $E_{PK_{TA}}\left(Q^{-1}x\right)$, $E_{PK_{TA}}\left(z^2\right)$, and $E_{PK_{TA}}\left(R^{-1}z\right)$ as a new block
6. Successful Mi creates a new block containing MSP attributes through an appropriate consensus

3.4 MSP Selection with Fake Response Detection Phase

Algorithm 3 is used to send current attributes of MSP to blockchain in response to emergency event raised by MU for MSP selection process. Thereafter, Mi checks whether the attributes sent by any MSP in response to emergency event is fake or not. This is done by checking if the attribute sent by MSP differ by more than a pre-defined threshold value δ from the attribute data already available to the miners in blockchain network. If these two attributes differ by more than the threshold value δ, then the system considers the received MSP attribute status as fake and discards it. However, the threshold value δ can be tuned as per the security level that MU wants to incorporate. Thus, all the valid responses from MSPs are considered for selecting the most appropriate MSP whose attributes are best matched against the attributes of MU in emergency.

Algorithm 3 for MSP Selection with Fake Response Detection Phase

1. MU detects emergency when PHI status is not normal
2. MU randomly generates β and computes public key $PK_{MU} = g^{\beta}$
3. MU generates an invertible matrix P of order same as its attribute vector m
4. MU generates encrypted threshold range set $\{c_0, \cdots, c_k\}$ where $c_i = E_{PK_{MU}}(\delta_i)\, \forall i \in \{0, \cdots, k\}$
5. MU generates signature $H = h_{MS}\Big(E_{PK_{MU}}(m^2), E_{PK_{MU}}(P^{-1}), (P \cdot m), PK_{MU}\Big)$
6. MU broadcasts the emergency notification along with $E_{PK_{MU}}(m^2)$, $E_{PK_{MU}}\left(P^{-1}\right)$, PK_{MU}, $P \cdot m$, nonce $E_{PK_{MU}}(N_1)$ and H to MSP, Mi and the values $\{c_0, c_1, \cdots, c_k\}$ only to Mi
7. MSP receives emergency notification from MU and checks integrity using equation 3
8. MSP generates signature $H_1 = h_{MS'}\Big(E_{PK_{TA}}(y^2), E_{PK_{MU}}\left(y^2\right), P \cdot m \cdot y, Q \cdot y, N_2\Big)$
9. MSP broadcasts $E_{PK_{TA}}(y^2)$, $E_{PK_{MU}}\left(y^2\right)$, $P \cdot m \cdot y$, $Q \cdot y$, $E_{PK_{TA}}(z_1^2)$, $R \cdot z_1$, $E_{PK_{MU}}(N_1)$, H_1 to Mi in response to the emergency event
10. Mi computes encrypted Euclidean distance d_1 between x, y using equation 6
11. Mi detects y as fake MSP status if $d_1 \notin \{c_0, c_1, \cdots, c_k\}$
12. If $d_1 \in \{c_0, c_1, \cdots, c_k\}$ then Mi considers this MSP response for MSP selection process.
13. Mi computes encrypted Euclidean distance d_3 between the MSP location used during registration and the MSP location broadcasted using equation 7
14. Mi detects z_1 as fabricated location in comparison to the stored MSP location if encrypted $d_3 \neq c_0$
15. If encrypted $d_1 \in \{c_0, c_1, \cdots, c_k\}$ and encrypted $d_3 \neq c_0$, then Mi accepts this MSP response as valid
16. Mi finds MSP with minimum distance d_2 using equation 8 and equation $E_{PK_{MU}}(F_s) = E_{PK_{MU}}(Min\{\left(d_2^2\right)_{MSP_1}, \cdots, \left(d_2^2\right)_{MSP_k}\})$
17. MU receives $E_{PK_{MU}}(\sum (x-y)^2)$, $E_{PK_{MU}}(P \cdot y)$, and $E_{PK_{MU}}(N_1)$ from Mi
18. MU compares nonce N_1 sent by it and the nonce it received from Mi. If same, then MU is confirmed about Mi response against its medical emergency
19. MU decrypts $\sum (x-y)$, $P \cdot y$ to know selected MSP

$$\{c_i\} = \{E_{PK_{MU}}(\delta_i)\}\, \forall i \in \{0, 1, \cdots, k\} \tag{1}$$

$$H = h_{MS}\Big(E_{PK_{MU}}(m^2), E_{PK_{MU}}(P^{-1}), P \cdot m, PK_{MU}\Big) \tag{2}$$

$$h_{MS'}(H) = h_{MS'}\Big(h_{MS}(E_{PK_{MU}}(m^2), E_{PK_{MU}}(P^{-1}), P \cdot m, PK_{MU})\Big) \tag{3}$$

$$= E_{PK_{MU}}(m^2), E_{PK_{MU}}(P^{-1}), (P \cdot m), PK_{MU}$$

$$H_1 = h_{MS'}\Big(E_{PK_{TA}}(y^2), E_{PK_{MU}}\left(y^2\right), P \cdot m \cdot y, Q \cdot y, N_2\Big) \tag{4}$$

$$E_{PK_{MU}}(F_s) = E_{PK_{MU}}(Min\{\left(d_2^2\right)_{MSP_1}, \cdots, \left(d_2^2\right)_{MSP_k}\}) \tag{5}$$

$$E_{PK_{d_1^2}} = \sum E_{PK_{TA}}\left(y^2\right) \cdot E_{PK_{TA}}\left(x^2\right) \cdot Q \cdot y \cdot E_{PK_{TA}}^{-2}\left(Q^{-1} \cdot x\right) \tag{6}$$

$$= \sum E_{PK_{TA}}\left(y^2\right) \cdot E_{PK_{TA}}\left(x^2\right) \cdot E_{PK_{TA}}^{-2}\left(Q \cdot y \cdot Q^{-1} \cdot x\right)$$
$$= \sum E_{PK_{TA}}(y - x)^2$$

$$E_{PK_{d_3^2}} = \sum E_{PK_{TA}}\left(z^2\right) \cdot E_{PK_{TA}}\left(z_1^2\right) \cdot R \cdot z_1 \cdot E_{PK_{TA}}^{-2}\left(R^{-1} \cdot z\right) \tag{7}$$

$$= \sum E_{PK_{TA}} z^2 \cdot E_{PK_{TA}} z_1^2 \cdot E_{PK_{TA}}^{-2}\left(R \cdot z_1 \cdot R^{-1} \cdot z\right)$$
$$= \sum E_{PK_{TA}}\left(z^2\right) \cdot E_{PK_{TA}}\left(z_1^2\right) \cdot E_{PK_{TA}}^{-2}(z_1 \cdot z)$$
$$= \sum E_{PK_{TA}}(z - z_1)^2$$

$$E_{PK_{d_2^2}} = \sum E_{PK_{MU}} y^2 \cdot E_{PK_{MU}}\left(m^2\right) \cdot P \cdot m \cdot y \cdot E_{PK_{MU}}^{-2}\left(P^{-1}\right) \tag{8}$$

$$= \sum E_{PK_{MU}} y^2 \cdot E_{PK_{MU}} m^2 \cdot E_{PK_{MU}}^{-2}\left(P \cdot m \cdot y \cdot P^{-1}\right)$$
$$= \sum E_{PK_{MU}}(y - m)^2$$

It computes the distance between MSP encrypted attributes x stored in blockchain and the updated MSP attributes y using homomorphic encryption [17]. If the Euclidean distance is less than or equal to the threshold value δ, then updated MSP attribute y is accepted as not fake. If $|x_i - y_i| > \delta$ then updated MSP attribute vector is detected as fake MSP attribute and discarded by miners. Among the genuine MSP responses being considered for MU in emergency MSP_1, $\cdots$, MSP_m, the MSP with minimum attribute distance from MU attribute is computed using equation 5 similar to [28], and this MSP is selected for addressing MU emergency.

3.5 Protocol Diagram

The proposed MSP selection with fake response detection protocol is depicted in Fig. 2. The proposed MSP Resource Registration protocol is depicted in Fig. 3. Secure communication channel is depicted in dotted lines.

4 Security and Privacy Analysis

Robustness of the proposed scheme have been analyzed in this section in terms of privacy and security. For this, the proposed scheme has been formally verified using appropriate random oracle model for its privacy. Apart from this, the proposed scheme has been simulated using AVISPA [1], an widely accepted formal verification tool. Moreover, the proposed scheme has been scrutinized by different known attacks and checked the stability of proposed scheme against these attacks. Prior to the actual security analysis, the underlying threat model is introduced next, which have been considered for the formal privacy and security verification of the proposed scheme.

4.1 Threat Model

In the threat model, TA has been considered to be honest but curious. All Mi are curious. In particular, majority of miners in blockchain are honest. Therefore, their adversarial objective is to reveal the medical status and location of the MU. TA and Mi follow the protocol rules. However, an adversary other than TA or Mi may attempt to perform both passive and active attacks. Since TA is honest, the threat model considers that there is no collusion between TA and malicious MSP. Following are the functionalities of an adversary in attempt to fulfill its objectives:

i. Eavesdrops the communication channel between different entities.

ii. Intercepts messages over communication links, and try to reveal any sensitive information.

iii. Captures a message from the communication channel during a particular session, and try to mount replay attack by replaying the captured message in a separate session.

iv. Attempts to mount man-in-the-middle attack by changing message passing through insecure channel.

4.2 Privacy Analysis using Real or Random Oracle Model

Here, a real or random (RoR) oracle model has been employed to scrutinize the privacy of proposed framework. As mentioned, the proposed framework consists of medical user MU, medical service provider MSP, miner Mi of blockchain network, trusted authority TA, and different communication channels. In this analysis, a RoR oracle O is defined which can execute following functions:

i. ExtractSecret(i, m): It models O to access a protocol instance i, and thereafter extracts the message m =

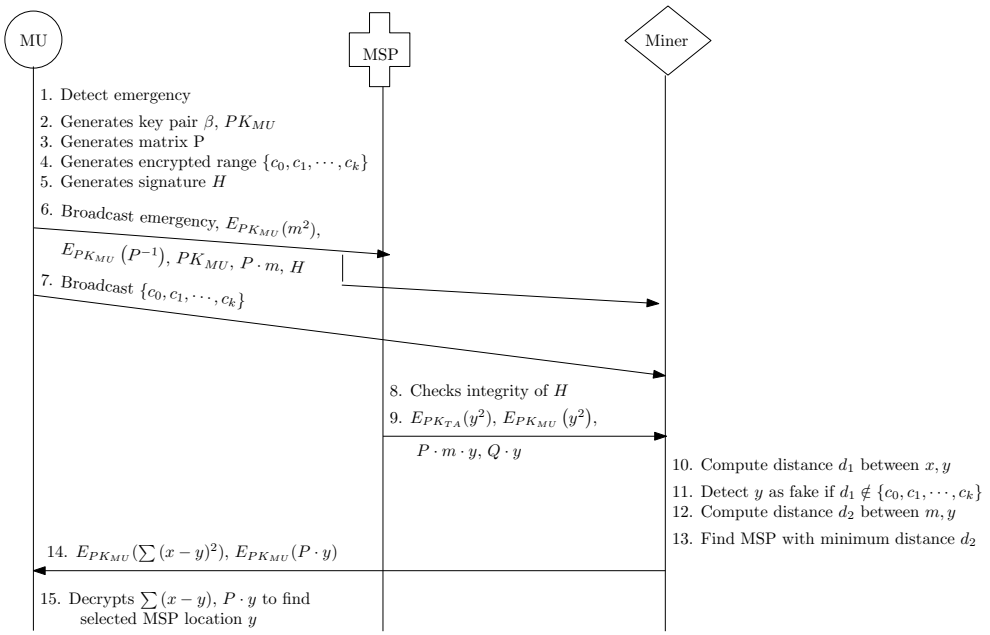

Fig. 2: MSP Selection with Fake Response Detection Protocol

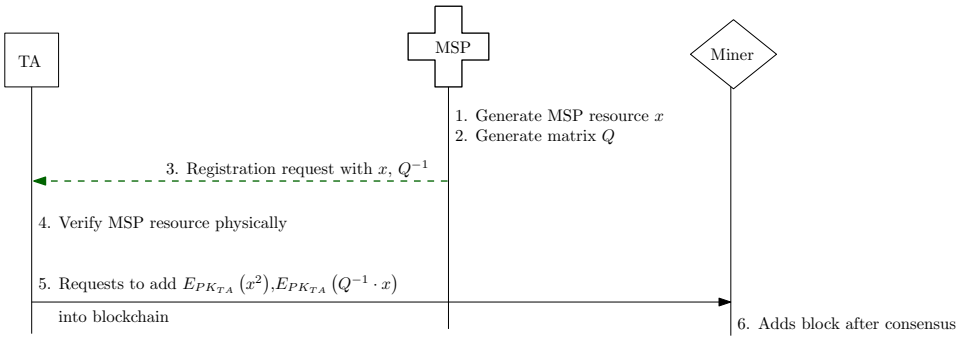

Fig. 3: MSP Resource Registration Protocol

$(m_1||m_2||m_3||m_4||m_5)$. Here, m_1, m_3 represent the encrypted attributes of MU and MSP respectively. The attributes of MU consists of the PHIs and location in the protocol instance i. The m_2 denotes the encrypted matrices P, Q. The m_4 consists of encrypted distance information between MU and MSP in addition to the encrypted $P \cdot y$, which are encrypted using the public key of MU. The m_5 contains encrypted distance threshold range sent by MU to Mi during i.

ii. GetAttribute(m_1, m_6): This function models O to obtain any one attribute value m_6 for the protocol instance i using the ciphertext m_1.

iii. GetMatrixP(m_3, m_6): The O executes this function to compute the matrix P generated in the protocol instance i in m_6 using the ciphertext m_3.

iv. ExtractSecretParam(j, m'): This function models O to access a protocol instance j, and thereafter obtains the message $m' = (m_7||m_8||m_9||m_{10}||m_{11})$, where m_7 consists of the encrypted attribute values of MSP. The attributes of MSP consists of the resources of MSP in the protocol instance j. The m_8, m_9 consist of encrypted location of MSP and encrypted matrices Q, R respectively. The m_{10}, m_{11} denote encrypted distances between registered MSP attributes, locations and broadcasted attributes, locations of MSP.

v. FindDistance(m_{11}, m_{12}, m_{13}): It enables O to use encrypted distance m_{11} between registered attribute and broadcast attribute of MSP, and encrypted distance m_{12} between registered location and broadcast location of MSP employed in the protocol instance j and obtains the distances in m_{13}.

It will be first proved that the proposed framework is capable to defend against known emergency attack. Then, it is proved that the proposed framework preserves privacy of MU. The assumption here is that, adversary $\mathcal{A}$ is very intelligent and it is having unbounded computational resources. Given a protocol instance i, the RoR oracle O is assumed to perform any computational task efficiently. Adversary $\mathcal{A}$ can use O to obtain the attributes of MU that includes PHI readings and/or the location information of MU used in the protocol instance i. $\mathcal{A}$ knows the MSP registration details, i.e. the attributes of MSP during registration. $\mathcal{A}$ also knows about the emergency event. However, $\mathcal{A}$ does not know the decrypted threshold value c set by MU. $\mathcal{A}$ needs to know this threshold value c for its success in influencing the decision in its favor to get selected to treat the MU in medical emergency. Now the following lemma is stated.

Lemma 1: The encrypted threshold value c generated by MU is secure.

Proof: To prove the Lemma 1, it is assumed that adversary $\mathcal{A}$ is able to decrypt the c. Since the c has been encrypted using private key of MU and this l-bit private key has been chosen randomly from the set of keyspace, the success probability to decrypt the encrypted threshold c is 2^{-l}. As in public key cryptography, the keystring length is fairly large, this value 2^{-l} is negligible. Hence, the encrypted threshold value c generated by MU is secure from attackers.

This lemma has been used in proof of the following theorem.

Theorem 1: Proposed framework is able to defend against known emergency attack.

Proof: To prove this theorem, a game $\mathcal{G}_1$ is designed for adversary agent $\mathcal{A}$. In this game, $\mathcal{A}$ is supplied with an incomplete and ongoing protocol instance j that is currently in progress and for which encrypted threshold value is available to $\mathcal{A}$. The game $\mathcal{G}_1$ execution time duration has an upper bound t_1, during which $\mathcal{A}$ has to complete the game and return the results to judge $\mathcal{J}$.

Here, $\mathcal{A}$ can be MSP or TA or any other third party adversary. Aim of adversary is to let false attribute of MSP be accepted for consideration in MSP selection. $\mathcal{A}$ is successful to launch known emergency attack if false attribute(s) of MSP are taken into consideration for MSP selection process. This may lead to, in the background of known emergency, the selected malicious MSP to attend the MU in medical emergency. To win in the game, $\mathcal{A}$ must be able to know the threshold value of MSP attribute deviation allowed from registered attributes. Outcome of the game is success if the $\mathcal{A}$ is able to inject false MSP attribute(s) into the system that passes the threshold value test set by MU. Outcome of the game is failure if the probability of $\mathcal{A}$'s ability to inject false attribute(s) of MSP is very much negligible.

The O gets the encrypted threshold value c sent by MU from the incomplete protocol instance j. O then tries to decrypt this c as per the Lemma 1. O does not know the threshold value set by MU. However, O knows from $\mathcal{A}$ that c is encrypted value of a member from the set $\{0, 1, 2, \cdots, k\}$. But, O does not know the key of MU. So, O needs to randomly guess a key of MU. If the length of key is l then the probability of correct guessing key is $\frac{1}{2^l} = 2^{-l}$. Then, O needs to randomly guess a threshold value set by MU. If the length of the threshold value is a member of the set $\{0, 1, 2, \cdots, k\}$, then probability of correctly guessing it randomly at once is $\frac{1}{k}$. Hence, the successfully obtaining the plaintext value of encrypted threshold value is $k^{-1} \times 2^{-l}$. This success probability is very much negligible when l is sufficiently large. Therefore, the probability of $\mathcal{A}$ winning the game in injecting false attributes in the context of known emergency is negligible and the proposed framework does not compromise with known emergency attack.

Hence, the proposed framework is able to sustain known emergency attack.

Theorem 2: Proposed framework preserves privacy of medical user MU.

Proof: To prove this, Lemma 1 is used. $\mathcal{A}$ wants malicious MSP be selected to attend MU in emergency. Which MSP attributes are to be changed and how much changes are to be calibrated in these attributes values are to be known by $\mathcal{A}$ so that these changes enable MSP to be selected for treatment of MU in emergency. A second game $\mathcal{G}_2$ is designed for adversary agent $\mathcal{A}$. In this game, $\mathcal{A}$ will be provided with a set of completed protocol instances $\mathcal{I}$. The game $\mathcal{G}_2$ run time duration has an upper bound t, during which $\mathcal{A}$ has to complete the game and return the parameters to judges $\mathcal{J}$.

Here, $\mathcal{A}$ is considered as successful if $\mathcal{A}$ computes any correct attribute, that is, PHI value and/or the location information of MU used in $i \in \mathcal{I}$ successfully. $\mathcal{A}$ uses the skills of O, and randomly picks a protocol instance $i \in \mathcal{I}$. The oracle O is invoked and given with protocol instance i. O thereafter uses its functionalities using Algorithm 4. In Algorithm 4, O accesses the protocol instance i, and at first executes ExtractSecret(i, m) to obtain different encrypted values used in the protocol instance i. O then separates the values accumulated in m using the parameters m_1, m_2, m_3, m_4 and m_5. Next, O executes the function GetAttribute(m_1, m_6) that uses the parameter m_1, and obtains the attribute values m_6. It randomly selects a PHI value or location information from the set of attribute values just computed, and assigns the selected attribute value in m_6. Thereafter, the oracle O executes GetMatrixP(m_3, m_4) to obtain the matrix P and its inverse P', which are then kept in the parameter m_4. Adversary $\mathcal{A}$ then returns the computed values m_2, m_4 along with the id of protocol instance i to judge $\mathcal{J}$. Upon receiving these pa-

Algorithm 4 executed by O

Input: Protocol instance $i \in \mathcal{I}$
Output: Probable attribute values in parameters m_6 and matrices in m_7, m_8

1. Randomly choose a protocol instance $i \in \mathcal{I}$
2. Execute ExtractSecret(i, m)
3. Execute GetAttribute(m_1, m_6)
4. Execute GetMatrixP(m_3, m_4)

rameters, $\mathcal{J}$ executes Algorithm 5 to decide the fate of $\mathcal{A}$. Algorithm 5 uses the following function.

1. Retrieve(i, m_9): This function is applied to retrieve the original attribute information of MU in m_9 for the protocol instance i. $\mathcal{J}$ retrieves these parameters from the database, which were recorded in the history of $\mathcal{I}$.

Algorithm 5 executed by the judges of $\mathcal{G}$

Input: Parameters m_6, m_9, i returned by $\mathcal{A}, t_1, \mathcal{I}$
Output: *Win* or *Defeat*

1. **If** time consumed by $\mathcal{A} > t_1$
2. **Then** Return *Defeat*
3. **Else**
4. Execute Retrieve(i, m_9)
5. **If** $m_6 \in m_9$
6. **Then** Return *Win*
7. **Else** Return *Defeat*
8. **End of inner If**
9. **End of outer If**

Algorithm 5 receives the input parameters m_6, m_9, i from $\mathcal{A}$ along with the completed protocol instance set $\mathcal{I}$, and the maximum duration t_1 of $\mathcal{G}_2$. In Algorithm 5, $\mathcal{J}$ delivers the verdict *Defeat* to $\mathcal{A}$ when duration of $\mathcal{G}_2$ played by $\mathcal{A}$ is $> t_1$. Otherwise, $\mathcal{J}$ executes Retrieve(i, m_9). In this function, $\mathcal{J}$ extracts the attribute m_9 from the database using the value of i. $\mathcal{J}$ announces the verdict *Win* $\mathcal{A}$ if it finds m_6 returned by $\mathcal{A}$ is in m_9. Therefore, $\mathcal{A}$ can be declared as winner if O successfully obtains the attribute value m_1, i.e., the PHI values and location of MU correctly.

The probability that O can correctly compute at least one correct attribute value in m_1 depends on the probability of obtaining correct attribute values from the given ciphertexts $E_{PK_{MU}}m^2$, which requires O to solve hash function or determining m_1 from $P \cdot m_1$ which requires to solve integer factorization problem. According to Birthday Attack assumption, an attribute value can be computed from a given ciphertext with the probability of $\left(\frac{1}{2}\right)^{\frac{k}{2}}$ where length of hash value is k bits. If there is the requirement of obtaining l number of attribute values to compute correct PHI and/or location, then the probability, p_1, of computing one correct attribute is $\left(\frac{1}{2}\right)^{\frac{lk}{2}}$. For any sufficient large value of k, this probability p_1 is very negligible.

Now, consider that the probability to retrieve m_1 from $P \cdot m_1$ is p_2. Retrieving m_1 from $P \cdot m_1$ involves solving integer factorization problem and then solving the system of linear equations $A \cdot x = b$ where both A, x are unknown. Given a fairly large integer b, no efficient algorithm exist in classical computing paradigm to find integers A, x such that $A \cdot x = b$. So, the probability p_2 to factorize $P \cdot m_1$ within time t_1 is very negligible. Therefore, the probability, p, to obtain m_1 is $< p_1 + p_2$. Since both p_1, p_2 are very much negligible, the probability of successful attack by $\mathcal{A}$ is very much negligible. Hence, the proposed scheme preserves privacy of MU.

4.3 Informal Security Analysis of the Proposed System

Here, security of the proposed framework is examined through informal security analysis. The primary security objective of the proposed framework is to safeguard privacy of MU. Neither attacker nor TA nor Mi nor MSP should be able to know attributes of MU. Even, the MSP would not be able to extract PHI or location information of MU prior to MSP selection for attending the MU in emergency.

Claim 1: The proposed framework preserves privacy of MU.

Proof: Privacy objective of the proposed system is that, when an MU encounters medical emergency, the system must not disclose either PHI or location of MU during MSP selection with false emergency response detection phase. It is also desired that distance between MU and MSP attributes must not be revealed until the MSP is scheduled to attend the MU in emergency. These are very sensitive and private information of MU. All the attributes of MU are represented by m. We first verify if this information, m can be extracted by TA or Mi or MSP or third party attacker prior to MSP selection. MU embeds m into $P \cdot m$ and encrypts m as $E_{PK_{MU}}m^2$. Due to difficulty of integer factorization problem, TA or MSP or Mi or attacker cannot extract m from $P \cdot m$. Similarly, due to the hardness of cracking a public key cryptosystem, it is very difficult to extract m from $E_{PK_{MU}}m^2$ without knowing PK_{MU}. Hence, privacy of MU attributes remain preserved.

Claim 2: The proposed framework provides confidentiality.

Proof: MU attribute m is confidential and cannot be accessed by either TA or MSP or Mi. Upon emergency, MU broadcasts $E_{PK_{MU}}m^2$ to MSP and Mi. Due to the hardness of cryptanalyzing a public key cryptosystem, MSP or Mi or adversary cannot obtain private key and hence cannot extract MU attributes m from $E_{PK_{MU}}m^2$.

Claim 3: The proposed framework has sufficient safeguard for integrity check.

Proof: In the MSP selection and fake response detection phase, MU generates signature $H \leftarrow h_{MS}\Big(E_{PK_{MU}}(m^2), E_{PK_{MU}}(P^{-1}), (P \cdot m), PK_{MU}\Big)$. MU then broadcasts the emergency event along with $E_{PK_{MU}}(m^2)$, $E_{PK_{MU}}\left(P^{-1}\right)$, PK_{MU}, $P \cdot m$, and H to MSP and Mi. Integrity of the emergency information received from MU is checked by MSP as $h_{MS'}\Big(h_{MS}(E_{PK_{MU}}(m^2), E_{PK_{MU}}(P^{-1}), (P \cdot m), PK_{MU})\Big)$. So, the third party adversary cannot be successful in altering the emergency information broadcast by MU without being detected by MSP through integrity check.

Claim 4: The proposed framework is secure against replay attack.

Proof: In replay attack, the attacker replays a message from a past session in the current session. An adversary can replay an old medical emergency message alert to MU to mislead MU in the present session. The nonce N_1 is used by MU prevents adversary to replay message of an old session in current session. Similarly, as MSP uses the nonce N_1 received from MU to broadcast its current status, adversary will not be successful in replaying an old MSP broadcast message in current session as that will be detected by MU.

Claim 5: The proposed framework is secure against man-in-the-middle attack.

Proof: In man-in-the-middle (MITM) attack, communication between two users is relayed through attacker and the message can also be altered. Since MU broadcasts its emergency event in encrypted form using its public key and MSP broadcasts its response to MU emergency event using public key of MU and public key of TA, and the private keys of MU and TA are not known to the adversary, adversary will not be able to successfully alter the messages being sent by MU and MSP. Hence, MITM attack is not feasible.

Claim 6: The proposed framework is secure against single-point failure problem.

Proof: TA is not involved in the proposed system beyond initialization and MSP resource registration phase. TA is not at all involved in the emergency detection and fake response detection phase. As a result, even if TA fails during medical emergency of MU, the process of MSP selection does not get hampered. So, the proposed system does not suffer from single-point failure problem.

Claim 7: The proposed framework is capable of detecting fake MSP status.

Proof: In response to MU emergency broadcast message, a malicious MSP can send a fake MSP attribute y to Mi. However, MU has determined, in prior, the maximum permissible difference δ between MSP attribute x registered by TA in blockchain network, and the current MSP attribute y. If the Euclidean distance between x and y, computed by Mi in encrypted domain, does not belong to set of encrypted threshold range $\{c_0, c_1, \cdots c_k\}$, Mi detects the MSP attribute y as fake. So, the proposed framework is capable to detect fake MSP status.

Claim 8: The proposed framework is capable of detecting known emergency attack.

Proof: A malicious MSP may be aware of some medical emergencies in its vicinity and accordingly wishes to get patients admitted in its facility although it may not be the most suitable facility to treat those patients in medical emergency. For this, it may successfully upload false data to system and extract preference in MSP selection process. The proposed model does not suffer from known emergency attack. To resist known emergency attack, MSP attributes are registered by TA in blockchain network. MU in emergency decides the threshold of deviation it allows for MSPs. In response to MU emergency event, every MSP sends its attributes to all Mi. Mi computes the encrypted difference between registered MSP attribute and currently received MSP attribute. If this difference is within the allowed threshold, only then the MSP is considered for MSP selection. As a result, the proposed system does not suffer from known emergency attack.

4.4 Formal Security Verification using AVISPA

To prove that a proposed protocol is correct, the symbolic model called Dolev-Yao model (DY model) [4] is often used. This DY model is an abstract model that helps to construct automatic verification tools. There are different tools such as AVISPA [26], and Tamarin [15] etc. which are examples of such symbolic models. AVISPA tool has been used here to prove that the protocols in proposed framework are secure. Three roles have been defined in the AVISPA code for three agents: MU, MSP, and Mi. Two more roles have been defined for session and environment. The goals that we set in the AVISPA code are the secrecy of MU and MSP attributes and authentication of MSP. The experimental result is depicted in Fig. 4, which concludes that the proposed scheme satisfies the desirable security properties.

5 Performance Analysis of Proposed System

Four different type of entities are present in the proposed system - MU, MSP, TA, and Mi. Out of these four kinds of entities, MU is resource constrained. TA, Mi, and MSPs have abundant resources. Performance analysis of MU and

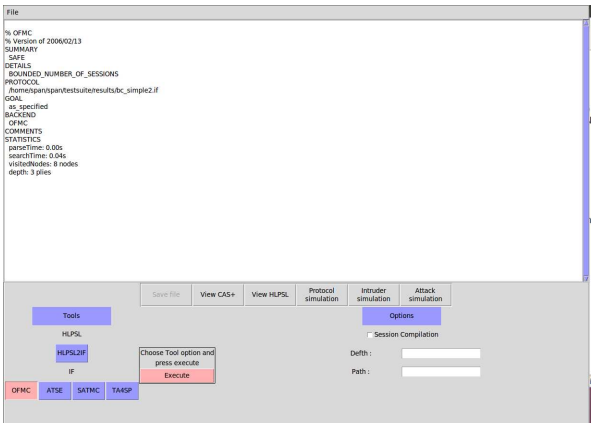

Fig. 4: AVISPA Code Execution Summary

Table 2: Primitive operations of MU, MSP, Mi, TA

Operation	MU	MSP	Mi	TA
+	$m^2 - m$			
×	m^2			
^	1			
compare	$m - 2$		$4 + 3t$	
random no. generation	2		3	
matrix generation	$m \times m$	$n \times n$		
		3×3		
matrix inverse	$m \times m$	3×3		
key-pair generation				2
hash	1	2	9	
$E_k()$	k			
$D_k()$	1			
homomorphic $E_k()$	2			
homomorphic $D_k()$	1			
homomorphic ×			4	
homomorphic scalar ×			2	

MSP, TA, and Mi, in terms of computation time, has been made in this section.

5.1 Computation Time

The computation time of MU, MSP, Mi, and TA in the proposed framework has been evaluated in terms of number of primitive operations used. Example of these primitive operations are addition, multiplication, exponentiation, compare, generation of matrices, hash value generation, encryption, decryption etc. The complete list of all such primitive operations and number of times they have been used by different entities are given in the Table 2. Here, m is the number of MU attributes, n is the number of MSP attributes, and t is the number of miner nodes in the network. The computation time taken by MU and MSP as obtained in our simulation results are depicted in Fig. 5 and Fig. 6. We have performed the simulation in Intel Core 2 Duo 1.83 GHz system. We have used SEAL library [10] for simulation. We have written smart contract using Remix IDE for deployment in Ethereum [20]. For easy of comparison of simulation results, it has been considered that size of MU attributes and MSP attributes are same.

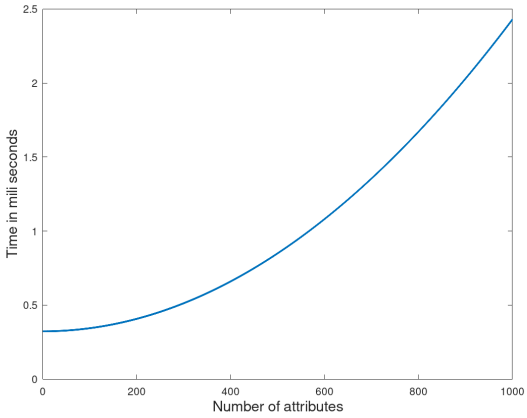

Fig. 5: Time taken by MU

Time taken by MU grows as the number of MU attributes increases. The algebraic expression of MU time is a polynomial of degree two. In other words, MU time is a function of m, size of MU attributes.

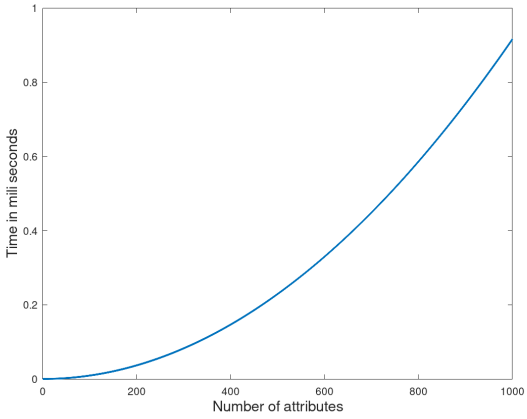

Fig. 6: Time taken by MSP

Time taken by an MSP increases as the number of MSP attribute increases. MSP time is a function of n, size of the MSP attributes. The MSP time function is a degree two polynomial. Time taken by a Mi is mostly influenced by time consuming homomorphic multiplications and hash value computations which have been used only a constant number of times by Mi. As per our simulation, Mi takes 45 seconds of time in a blockchain network with one thousand

miner nodes and TA takes a constant amount of time, 0.0751 milliseconds since it does not depend on the attribute size of the MU or MSP or number of miners in the blockchain network.

6 Conclusion

BAN is an important artifact in smart healthcare. To respond quickly to a medical emergency of MU, appropriate selection of the most suitable MSP, while preserving privacy of MU, is a vital task. Failure in selection of appropriate MSP can result in poor medical treatment of MU. A malicious MSP can be aware of some medical emergencies in its surroundings and can try to exploit this information to get more business. It can try to upload false data to get preference in getting MUs to attend them in medical emergency. This has been classified as known emergency attack. This work proposes a framework, where a blockchain based solution of MSP selection has been proposed to prevent the malicious MSPs from launching known emergency attack. Security and privacy of the proposed framework has been evaluated using the Real or Random Oracle Model and the AVISPA tool. Evaluation of computational requirement of different components of the proposed system has been also performed, which confirms the eligibility of the proposed scheme in real life.

7 Statements and Declarations

Authors declare the following.

7.1 Ethics Approval

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

7.2 Conflict of Interest

Both the authors declare that they have no conflict of interest.

7.3 Data Availability

Authors declare that the supporting data and/or program used for the work are available with the authors and will be made available to reviewer(s) and/or editor as and when required so.

7.4 Author Contribution

Anupam Pattanayak has contributed in problem statement formulation, developing the algorithms and protocols, performing security analysis, writing programs and performing simulations, drawing the figures, preparing and editing the drafts.

Subhasish Dhal has contributed in problem statement formulation, developing the algorithms and protocols, supervising security analysis and simulations, proofreading and editing the drafts.

7.5 Funding

No explicit funding information is available for this work.

7.6 Consent to Publish

Authors Anupam Pattanayak and Subhasish Dhal give their consent to publish the submitted manuscript if it is accepted after completion of review process.

References

1. Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., et al., (2005). “The AVISPA tool for the automated validation of internet security protocols and applications”, In *Proceedings of International conference on computer aided verification*, (pp 281–285). https://doi.org/10.1007/11513988_27

2. Chen, M., Gonzalez, S., Vasilakos, A., Athanasios, C., Huasong, L. & Victor, C., (2011). “Body Area Networks: A Survey,” *Mobile Networks and Applications*, 16(2), 171–193. https://doi.org/10.1007/s11036-010-0260-8

3. Chiou, S., & Liao, Z., (2018). “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*, 6, 35787–35800. https://doi.org/10.1109/ACCESS.2018.2847030

4. Dolev, D., & Yao, A., (1983). “On the security of public key protocols", In *IEEE Transactions on Information Theory*, 29(2), 198--208.https://doi.org/10.1109/TIT.1983.1056650

5. Griggs, K. N., Ossipova, O., Kohlios, C. P., Baccarini, A. N., Howson, E. A., & Hayajneh, T., (2018). “Healthcare blockchain system using smart contracts for secure

automated remote patient monitoring," *Journal of medical systems*, 42(7), 1–7. https://doi.org/10.1007/s10916-018-0982-x

6. Hajar, M. S., Al-Kadri, M. O., & Kalutarage, H. K., (2021). "A survey on wireless body area networks: Architecture, security challenges and research opportunities," *Computers & Security*, 104, 102211. https://doi.org/10.1016/j.cose.2021.102211

7. Hasan, K., Biswas, K., Ahmed, K., Nafi, N. S., & Islam, M. S., (2019). "A comprehensive review of wireless body area network," *Journal of Network and Computer Applications*, 143, 178–198. https://doi.org/10.1016/j.jnca.2019.06.016

8. Hasavari, S., & Song, Y. T., (2019). "A secure and scalable data source for emergency medical care using blockchain technology", In *Proceedings of IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp 71–75. https://doi.org/10.1109/SERA.2019.8886792

9. Huang, Q., Wang, L., & Yang, Y., (2017). "Secure and privacy-preserving data sharing and collaboration in mobile healthcare social networks of smart cities," *Security and Communication Networks*, 2017. https://doi.org/10.1155/2017/6426495

10. Laine, K., (2017). "Simple encrypted arithmetic library 2.3.1", In *Microsoft Research*. https://www.microsoft.com/en-us/research/uploads/prod/2017/11/sealmanual-2-3-1.pdf

11. Latré, B., Bart, B., Moerman, I., Blondia, C., & Demeester, P., (2011). "A Survey on Wireless Body Area Networks," *Wireless Networks*, 17(1), 1–18. https://doi.org/10.1007/s11276-010-0252-4

12. Lai, X., Liu, Q., Wei, X. Wang, W., Zhou, G., & Han, G., (2013). "A Survey of Body Sensor Networks," *Sensors*, 13(5), 5406–5447. https://doi.org/10.3390/s130505406

13. Li, C. T., Lee, C. C., & Weng, C. Y., (2016). "A secure cloud-assisted wireless body area network in mobile emergency medical care system," *Journal of Medical Systems*, 40(5), 1–15. https://doi.org/10.1007/s10916-016-0474-9

14. Lin, H., Shao,J., Zhang C., & Fang, Y. (2013). "CAM: Cloud-Assisted Privacy Preserving Mobile Health Monitoring," *IEEE Transactions on Information Forensics and Security*, 8(6), 985–997. https://doi.org/10.1109/TIFS.2013.2255593

15. Meier, S., Schmidt, B., Cremers, C., & Basin, D., (2013). "The TAMARIN Prover for the Symbolic Analysis of Security Protocols", In *Proceedings of International Conference on Computer Aided Verification*, pp 696–701. https://doi.org/10.1007/978-3-642-39799-8_48

16. Movassaghi, S., Abolhasan, M., Lipman, J., Smith, D., & Jamalipour, A.,(2014). "Wireless Body Area Networks: A Survey," *IEEE Communications Surveys Tutorials*, 16(3), 1658–1686. https://doi.org/10.1109/SURV.2013.121313.00064

17. Naehrig, M., Lauter, K., & Vaikuntanathan, V., (2011). "Can homomorphic encryption be practical?", In *Proceedings of the 3rd ACM workshop on Cloud Computing Security Workshop*, 113–124. https://doi.org/10.1145/2046660.2046682

18. Narwal, B., & Mohapatra, A. K., (2021). "A survey on security and authentication in wireless body area networks," *Journal of Systems Architecture*, 113, 101883. https://doi.org/10.1016/j.sysarc.2020.101883

19. Negra, R., Jemili, I., & Belghith, A., (2016). "Wireless body area networks: Applications and technologies," *Procedia Computer Science*, 83, 1274–1281. https://doi.org/10.1016/j.procs.2016.04.266

20. Nizamuddin, N., Salah, K., Azad, M. A., Arshad, J., & Rehman, M., (2019). "Decentralized document version control using ethereum blockchain and IPFS," *Computers & Electrical Engineering*, 76, 183–197. https://doi.org/10.1016/j.compeleceng.2019.03.014

21. Pattanayak, A., & Dhal, S., (2020). "Cloud Enabled Body Area Network," In Biswash, S.K., & Addya, S.K. (Eds.). *Cloud Network Management: An IoT Based Framework*, Chapman and Hall/CRC, (1st ed. 67–85). https://doi.org/10.1201/9780429288630

22. Pattanayak, A., Dutta, M., & Dhal, S., (2023). "Privacy Preserved Medical Service Provider Selection in Cloud-based Wireless Body Area Network," *Wireless Personal Communications*, 128(2), 1349–1371. https://doi.org/10.1007/s11277-022-10003-w

23. Rajput, A. R., Li, Q., Ahvanooey, M. T., & Masood, I., (2019). "AEACMS: Emergency access control management system for personal health record based on blockchain", In *IEEE Access*, 7, 84304–84317. https://doi.org/10.1109/ACCESS.2019.2917976

24. Siemon, C., Rueckel, D., & Krumay, B., (2020). "Blockchain technology for emergency response", In *Proceedings of the 53rd Hawaii International Conference on System Sciences*. http://hdl.handle.net/10125/63814

25. Sun, J., Fang, Y. & Zhu, X., (2010). "Privacy and emergency response in e-healthcare leveraging wireless body sensor networks," *IEEE Wireless Communications*, 17(1), 66–73. https://doi.org/10.1109/MWC.2010.5416352

26. Viganò, L., (2006). "Automated Security Protocol Analysis With the AVISPA Tool ", In *Electronic Notes in Theoretical Computer Science*, 155, 61–86. https://doi.org/10.1016/j.entcs.2005.11.052

27. Xu, C., Wang, N., Zhu, L., Sharif, K., & Zhang, C., (2019). "Achieving searchable and privacy-preserving data sharing for cloud-assisted E-healthcare system", In *IEEE Internet of Things Journal*, 6(5), 8345–8356. https://doi.org/10.1109/JIOT.2019.2917186

28. Yu, W., Liu, Z., Chen, C., Yang, B., & Guan, X., (2017). "Privacy-preserving design for emergency response scheduling system in medical social networks," *Peer-to-Peer Networking and Applications*, 10(2), 340–356. https://doi.org/10.1007/s12083-016-0429-4

29. Zhang, K., Liang, X., Barua, M., Lu, R., & Shen, X. S., (2014). "PHDA: A priority based health data aggregation with privacy preservation for cloud assisted WBANs," *Information Sciences*, 284, 130–141. https://doi.org/10.1016/j.ins.2014.06.011