IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 6, 2018, accepted May 26, 2018, date of publication June 13, 2018, date of current version July 6, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2846779

Continuous Patient Monitoring With a Patient Centric Agent: A Block Architecture

MD. ASHRAF UDDIN, ANDREW STRANIERI, IQBAL GONDAL, AND VENKI BALASUBRAMANIAN

Internet Commerce Security Laboratory, Centre for Informatics and Applied Optimisation, Federation University Australia, Ballarat, VIC 3350, Australia

Corresponding author: Md. Ashraf Uddin (mdashrafuddin@students.federation.edu.au)

This work was supported by the Internet Commerce Security Laboratory, Centre for Informatics and Applied Optimisation, Federation University Australia

ABSTRACT The Internet of Things (IoT) has facilitated services without human intervention for a wide range of applications, including continuous remote patient monitoring (RPM). However, the complexity of RPM architectures, the size of data sets generated and limited power capacity of devices make RPM challenging. In this paper, we propose a tier-based End to End architecture for continuous patient monitoring that has a patient centric agent (PCA) as its center piece. The PCA manages a blockchain component to preserve privacy when data streaming from body area sensors needs to be stored securely. The PCA based architecture includes a lightweight communication protocol to enforce security of data through different segments of a continuous, real time patient monitoring architecture. The architecture includes the insertion of data into a personal blockchain to facilitate data sharing amongst healthcare professionals and integration into electronic health records while ensuring privacy is maintained. The blockchain is customized for RPM with modifications that include having the PCA select a Miner to reduce computational effort, enabling the PCA to manage multiple blockchains for the same patient, and the modification of each block with a prefix tree to minimize energy consumption and incorporate secure transaction payments. Simulation results demonstrate that security and privacy can be enhanced in RPM with the PCA based End to End architecture.

INDEX TERMS Blockchain, body area sensor network, healthcare, remote patient monitoring, patient centric agent, proof of work, streamed data, Internet of Things, dynamically generated session key, patient record encryption key.

I. INTRODUCTION

Internet of Things(IoT) [1] applications in the modern healthcare system include devices, services and wireless sensors that detect physiological signs with wearable or ingestible sensors [2] that stream data to remote, and often Cloud based servers. Secure continuous monitoring of patient's physiological signs [3] has the potential to augment traditional medical practice, particularly in developing countries that have a shortage of healthcare professionals.

Remote Patient Monitoring(RPM) [4],[5] involves the integration of physiological data collected with Wearable or Implantable Medical Devices(IMDs), with other data including demographic, health record and geographic location data.

The challenges of designing effective, efficient and secure remote patient monitoring systems include the aggregation and indexing of huge streams of continuous data while maintaining patient privacy. Privacy [6] refers to one's personal space that also includes the capacity to have control over data and determine access levels to be granted to others.

In 2013 [7], an alarming 44 percent of all registered data in targeted medical companies was breached. Data breaches reportedly increased by 60 percent from 2013 to 2014 which led to financial losses that increased by a remarkable 282 percent. The vulnerability of wireless communications and relatively weaker cryptographic techniques compared to wired communications make RPM communications an easy target [8].

The threats to the confidentiality, integrity, and availability of healthcare information come from insiders, and outsiders in addition to operational environments [9]. Insiders such as healthcare professionals and support staff, service providers, and outsiders such as hackers threaten health information security by gaining unauthorized access to confidential data.

Actions by unauthorized persons can result in alteration of patient's information and can even cause death. Breaches of privacy can erode trust that patients and health care professionals place in the system.

Software disruption caused by viruses, worms, and malware, in addition to resource misuse such as personal use of systems can also threaten health information systems. Communication infiltration, interception, embedded malicious code and repudiation of patient's data pose threats to the confidentiality and integrity of patient's data. Accidental misrouting, technical infrastructure failure, and operational errors can also jeopardize the security of health information.

As outlined in the next section, existing RPM architectures are yet to address these threats. Therefore, there is a need for architectures that afford greater protection of RPM devices and software against attacks.

An eHealth framework [10] for RPM requires that privacy should be preserved while enabling access by authorized healthcare professionals.

Efforts to ensure privacy in RPM have been made in recent years however most approaches focus on a single link in the architecture that chains data from patient sensors to health care professionals through intermediary devices and servers. The archetypal RPM architecture involves sensors near, on or in the patient transmitting data wirelessly using Bluetooth, ZigBee or customized protocols, through a Body Area Wireless Sensor Network to a Base station that initially process data and transmits it to remote servers for further processing.

An effective and efficient RPM needs to address issues of rapid storage at appropriate security levels, user authentication, access control, mobility management and sustainability of patient's health data.

In this article, an End to End eHealthcare architecture is advanced that addresses RPM healthcare data management issues to ensure that appropriate levels of trade-off between effectiveness and privacy can be established for rapid, secure data storage and access, user authentication, role based access, and sustainability. Key features of the architecture include a Patient Centric Agent to co-ordinate End to End data streams and a Blockchain component for distributed storage of parts of the data.

A. RAPID STORAGE AND ACCESS

Large volumes of data streams can be generated by body area sensors. Some conditions such as a sudden arrhythmia may demand a quick response [11]. However, the analysis of a patient's data to determine appropriate actions as data rapidly streams in from sensors challenges computational processing and storage resources. Further, not all of the data generated from patient's body sensors can be assumed to be sensitive or required to be stored [12]. Some raw data streams may even be replaced with simple descriptors over aggregated data such as ''normal heart rate pattern'' for 24 hours or the result of real time analyses such as those proposed [13], [14]–[16]. Chuah and Fu [13] defined a threshold of separating normal and abnormal ECG data. The ECG signals which width is less than 100 ms and the R-to-R interval is between 0.8s and 0.9s or width less than 60ms, and the R-to-R interval goes beyond 1.1s are classified as Normal ECG waveform. The signals that do not fall in this range are classified as abnormal ECG data. Other streamed data can be presumed to need to be stored but without strong encryption. This means that a process is required to rapidly determine and execute a Data Storage Security Level required for a stream of data. A Patient Centric Agent, described below is proposed to perform the role of determining the data storage security level required. The PCA performs three main functions:-

B. USER AUTHENTICATION

Only authenticated nodes should participate in the transfer of data to prevent attackers intercepting data flows. Although asymmetric key cryptography algorithms ensure data cannot practically be decrypted without the key, they cannot guarantee the owner of a public key is the legitimate owner without the involvement of a trusted authority to issue public/private key pairs. But reliance on trusted authorities for key management results in security, fault tolerance and bottleneck problems in IoT settings [17]. In RPM, a compromised trusted authority might stop the real-time monitoring of all of the patient's physiological data. Further, asymmetric key cryptography [8] is computationally expensive and places great demands on power constrained battery operated devices essential in RPM architectures. The key management is performed by the PCA as a Trusted center at patient's end. The PCA uses symmetric key cryptography for BSN and asymmetric cryptography for SDP and customized Blockchain.

C. ROLE BASED ACCESS

Role Based Access Control(RBA) [18] refers to restricted privileges of healthcare professionals on patient data according to their expertise or roles. A patient's physiological data may contain sensitive information where a patient prefers access to be restricted to selected physicians. Inappropriate interpretation of health data may lead to incorrect treatment that might eventually result in long term consequences for a patient's health. The PCA ensures Role Based Access using Access Grant Transaction of the Blockchain.

D. SUSTAINABILITY

A financial model is required to ensure that all participants in the electronic health data storage service receive appropriate incentives to ensure the eHealth systems remain sustainable, and affordable. Financial transactions between healthcare professionals, patients, insurers, government and others are commonplace. So a secured fund transaction process is required to be included in RPM architecture.

The framework advanced in this article, includes Blockchain technology embedded into an End to End architecture. Blockchain for cryptocurrencies is a shared, authenticated, auditable and tamper-proof distributed database [19].

The anonymous properties of a transaction in digital currency address some challenges inherent in patient privacy. But existing Blockchain technology in digital currency cannot be applied as is, to IoT based RPM data because of high computational costs and long transaction processing times. RPM data can stream from sensors so rapidly that it cannot be feasibly processed and added to a Blockchain in real time resulting in delays might discourage patients from using Blockchain. Volunteer miners might also be reluctant to join Blockchain networks owing to the large storage and processing requirements.

We propose that RPM challenges can be reduced with the inclusion of a Patient Centric Agent(PCA). The Patient Centric Agent(PCA) has oversight over the End to End flow of data. The PCA determines the storage, security and access level required at any point in time. The PCA coordinates different segments of RPM such as patient sensors and devices, Blockchain nodes, and healthcare service provider devices. The PCA determines whether a stream of data should be stored in a Blockchain and manages the process, if so. The PCA executes on a machine with mass memory capacity and high processing power. No studies to date have advanced the notion of embedding Patient Centric Agent with customized Blockchain for RPM.

The PCA based End to End RPM architecture securely connects a patient's BSN to healthcare providers through different intermediate devices. It has the following design elements, explained in more detail throughout the article;

1) Two tiers-One tier deals with the stream and storage of data. The second tier, called the Healthcare Control Unit, deals with auditing and key management.
2) A secure communication protocol from BSN to patient's smartphone and smartphone to the Patient Centric Agent(PCA). This involves a lightweight authentication algorithm that includes dynamically generated sessional symmetric keys to confirm an End to End security as well as consumption of less power.
3) A Blockchain customized for RPM. Modifications include:
    - The task of selecting a Miner is left to the PCA so that computational effort is reduced and multiple Blockchains can be accommodated.
    - The modification of a block with prefix tree to minimize energy consumption and incorporate secure transaction payments.

The architecture advanced here envisages the PCA playing a central role enforcing security, mediating access to relevant electronic health records, storing particularly sensitive RPM data in a distributed manner, and enabling secure payments.

In the paper, we review related papers in Section IIand describe our proposed framework in SectionIII. The performance of key algorithms in the architecture is demonstrated in Section IV before concluding remarks.

II. RELATED WORK

We review the state-of-the-art works in three categories: conventional RPM solutions, attribute based RPM solutions and Blockchain based RPM solutions.

A. CONVENTIONAL RPM SOLUTIONS

Codeblue [20] is one of the earliest healthcare architectures developed based on BSN worn by the patient. Medical sensors wirelessly transmit the sensing data to end users such as PDA(Personal Digital Assistance), laptop, and personal computer. Healthcare professionals issue queries for the analysis of patient's data in a publish-subscribe manner. Although authors in the Codeblue project highlighted the need for security and privacy with medical data, they did not include privacy and security protection in their architecture.

Alarm-net [21] is a heterogeneous network architecture consisting of body sensor networks and environmental sensors for patient health monitoring in the assisted living and home environment. The circadian activity rhythms module in Alarm-net help to adjust context-aware power management and privacy policies. Alarm-net which is connected to BSN, back-end server and IP network imposed some network and data security policies for physiological, environment, behavioral parameters about residents [22]. However, Kumar and Lee *et al.* [23] and Ng *et al.* [24] showed that Alarm-net could not guard against the leakage of resident's location. Further, Kumar and Lee [23] pointed that hardware built in cryptosystem in Alarm-net makes the application highly platform dependent. Although Alarm-net performs some initial analysis on sensor data for power management, it did not focus on storage management, patient mobility management, security level of streamed physiological data as our proposal extend these techniques.

UbiMon [24] proposed by Ng *et al.* is BSN based healthcare system and addresses the issues of wearable and implantable sensors for distributed monitoring. Although UbiMon is an ubiquitous healthcare architecture consisting of LUP(Local Processing Unit) that can detect patient's abnormalities and issue instant warning to healhcare service provider, central server and work station for physician, but Ng did not consider security and privacy in their ubiquitous healthcare monitoring architecture. We extended this architecture by placing a smart agent at the patient's end and replace the central server with Blockchain technology. We also embed proper security at each segment of our architecture. Intelligent analysis of huge streamed physiological data requires a dedicated patient agent at the patient's end.

Chakravorty designed a wide-area mobile patient monitoring called MobiCare [25] to collect physiological conditions of patients continuously. MobiCare improves the quality-of-patient care and Mobicare server gives access to physiological data off line to medical staff. Although Chakravorty addressed the security and privacy for real-time applications, secure localization, and anonymity are not yet implemented.

Sensor Network for Assessment of Patients(SNAP) [26] has been proposed to address security concerning wireless health monitoring but it does not authenticate users while providing medical data. Furthermore, adversaries can intercept or modify physiological data because text to the controller is not encrypted in the architecture.

MEDiSN [27] consists of multiple physiological motes powered by a battery for in-hospital patient monitoring. The MEDiSN architecture addresses the issue of reliable communication, routing, data rate, and QoS. In the design, authors expressed the necessity of encrypting physiological monitoring data, but their study did not report the encryption technique that ensures confidentiality and integrity.

Moosavi *et al.* [28] proposed an End to End security scheme for mobility enabled healthcare IoT. The End to End Security Scheme architecture consists of three-layers, the device layer(BSN), fog layer(gateways, network router), and Cloud layer. Moosavi proposed a secure and efficient end-user authentication and authorization architecture based on the certificate DTLS(Datagram Transport Layer Security) handshake, secure End to End communication based on session resumption, and robust mobility based on interconnected smart gateways. DTLS depends on a trust center and involves a higher number of flights to complete the authentication process. Storage of physiological data in the Cloud causes higher latency and consumes bandwidth for continually monitoring healthcare system. Furthermore, Moosavi et al. did not focus on access control on patient's data in the Cloud. We advance our End to End e-healthcare architecture incorporating Blockchain technology, which includes a secure payment system and employ more lightweight authentication at the patient's end. A trusted authority in our architecture has a role to play in certifying healthcare professionals through Blockchain.

Gope and Hwang [10] proposed a modern healthcare system called BSN-Care for RPM. BSN-Care architecture includes conventional devices such as BSN, Local Processing Unit(LPU) and BSN-Care Server in healthcare monitoring system. In BSN-Care, BSN-Care Servers analyze patient physiological data transferred by LPU using heuristic approach. BSN-Care Server alerts healthcare givers if physiological data exceed a preset thresholds. Gope proposed a one-way hash based lightweight authentication method that uses shadow identity to preserve patient privacy and security. Data processing may bottleneck because the architecture depends on a single server. Yeh [29] also used the architecture of BSN-Care. He proposed hash based public/private key authentication LPU to BSN-Care Server and a hash based lightweight authentication integrating GPS for BSN to LPU.

In this work, we extend the BSN-Care lightweight authentication by including proximity and HMAC [30](keyed-Hash Message Authentication Code) for Body Area Sensor Network to smartphone communication channel. Proximity helps BSN devices in RPM to detect attacker's devices because of their physical location. In proximity authentication, two entities estimate distances between each other by exchanging some signals such as radio or voice. Therefore, device cannot claim incorrect physical location during authentication. We include GPS and options for using different kinds of encryption algorithms for smartphone to PCA communication channel. Spoof attack can be prevented by GPS. Option on using different encryption algorithm along a communication channel delays the attacker's effort to break data confidentiality.

Central Server based architecture [20]–[29] serves to store and analyze health records of limited number of patients. Therefore, we propose a scalable healthcare architecture integrating customized Blockchain, which is scalable and robust against attacks.

B. ATTRIBUTE BASED RPM SOLUTIONS

Attribute based authentication [31] refers to validating entities/persons on conditions that they possesses a certain number of attributes such as a person must be a doctor, heart specialist and service experience of 10 years to access a patient's record with heart diseases. A trust party ensures that a person owns the required properties in attribute based authentication and encryption. Two attribute-oriented authentication and transmission schemes for secure and privacy-preserving health information sharing in health social networks (HSNs) that is a social platform like Twitter for patients and healthcare service providers to share medical records and their views are proposed in where the access policy is defined by a target set of attributes such as patient identity, diseases history, and social status. Only users who satisfy the access policy are able to decrypt the cipher text. Privacy is preserved in this approach because it does not require the identity of the entity. They demonstrate that the proposed schemes can effectively resist various attacks including forgery attack, attribute-trace attack, eavesdropping attack, and collusion attack. However, authors didn't focus on revocation that refers to remove the access capabilities of authorized users any time and write operation on attribute-based encrypted medical data. Lounis *et al.* [32] pointed that medical data encrypted on attributes in the Cloud needs to be downloaded and stored again if the access policy associated with attributes changes. The computational cost of attribute based cryptosystems increases linearly with the number of attributes. The papers used symmetric key to encrypt data files in Cloud. The symmetric key is encrypted with access policies associated with attributes. Therefore, symmetric key requires to be encrypted if access policy is changed.

Liang *et al.* [33] also used patient's attributes to ensure authentication in smart home based pervasive healthcare system while communicating to online healthcare provider. To preserve a patient's location privacy, a receiver chain is formed where the source node requests a neighbor node to be a proxy source to enable vendor-to-patient communication. However, the source node can still be traced along the chain. Li *et al.* proposed a scalable personal health records to be stored in the Cloud using attribute based encryption so that

patient's record can be securely shared with multi-authority by maintaining appropriate privacy. The scheme also supports the dynamic modification of attribute policies, on-demand user/attribute revocation and break-glass access at the time of emergencies. Although attribute based encryption schemes provides security and privacy of patient's record, the scheme is not lightweight enough to implement in wearable medical sensors and smartphone [46].

C. BLOCKCHAIN BASED RPM SOLUTIONS

In the End to End frameworks proposed by [20]–[33], the patient must depend on Trusted Centers for key management. The devices that streamed real-time data in RPM experienced higher latency and communication overhead to obtain keys from Trusted Centers. Further, these frameworks don't ensure the availability of patient's data if the traditional server or Cloud server is compromised. Single point health applications described by [34],[35] have some drawbacks such as when a user goes to another hospital, the previous hospital may be reluctant to share data. Further, healthcare professionals can violate privacy by providing patient's data to a third party. Health data might suffer from a single point of failure.

In addition, traditional remote patient monitoring system requires authorization between remote end user/healthcare center and medical devices at the patient's end. The authorization causes communication overhead (required bandwidth, computational overhead, the number of transmitted message).

Blockchain healthcare architecture reduces the communication overhead to eliminate the requirements of running authorization algorithm for the remote end user to access data from the Blockchain [36], [37]. Further, Blockchain healthcare provides the facilities of peer-to-peer record's transmission without the involvement of third party trust, interpretability of longitudinal healthcare records, transparency with pseudonymity and irreversibility of records. Patients in Blockchain have options to hide their identity with alphanumeric address or show proof of their identity to others [36].

On the other hand, Blockchain technology applied in RPM involves a high-cost consensus process, IoT data can plausibly be generated faster than Blockchain consensus approach can validate the proof of concept [38]. An optimized Blockchain is required in order to preserve privacy in IoT based remote patient monitoring.

Zhang *et al.* [39] introduced a modified IEEE 802.15.6 authentication association protocol by considering the limited processing power of sensor nodes for pervasive social network based healthcare between BSN and smartphone. IEEE 802.15.6 requires two scalar multiplications at the BSN end and two scalar multiplications at the smartphone. In modified 802.15.6, sensor device performs one scalar multiplication. Secondly, they use a Blockchain oriented architecture that consists of the body area wireless sensor network and PSN(powerful computer, laptop, and smartphone build this network). They included a coordinator node(smartphone) placed at user end, which broadcasts transaction among PSN to verify its signature(using the master key exchanged through the modified protocol described in the first part of their works) of the sensor and the node itself. The transaction of Blockchain in this proposal doesn't contain physiological health record. The transaction includes patient or healthcare providers' meta data such as identity, address, and diseases. Their modified IEEE 802.15.6 still involves high computation for BSN because scalar multiplication is computationally expensive operation.

Bowhead [40] is a Blockchain based healthcare application. The application introduces some medical devices such as test cartridge, test reader and dispensing devices to capture patient data. They have also designed an App that prompts patients about the dosage and time of taking medication. The user can provide his information to Bowhead's application through different body area medical sensors and the application stores that information to a Blockchain based database. Although white paper describes the procedure for collecting patients data, it does not describe how the stream of medical data produced from patients body fit to the existing Blockchain.

MeDShare [41] is also a medical data sharing system among Cloud service provider via Blockchain contract. The contract refers to a program written by user defining terms and conditions of an agreement. The Blockchain nodes only justify the rules of the agreement. The authors propose an architecture for sharing documents among requestors. In design approach, they discussed the system setup, requested file, package delivery, auditing and provenance in detail where the function of each layer of their architecture and smart contact technology in Blockchain are integrated to share data securely. However, MedShare constitutes only a sub-system in RPM architecture.

Health Care Data Gateway(HDG) [42] is a smartphone based App that integrates traditional database and Blockchain distributed database to manage patient health data. They proposed a multiparty computation(MPC) approach where third parties can access data but not alter data. HDG consists of three layers called Storage Layer, Data Management Layer, and Data Usage Layer. Cloud is the platform for Storage Layer in Blockchain fashion. The Data Management Layer comprises individual devices. All kinds of request either incoming or outgoing will pass through this layer. The Data Management Layer helps in indexing and making queries to retrieve data from the Cloud. Data Usage Layer includes physicians, electronic medical record system, and data analytical algorithms. The diagnostic center will directly send data to patients and patients transfer authority for the data to the doctor for further analysis. The third party might continue analyzing data without accessing user's data through MPC in the Blockchain. They proposed a unified scheme to store medical data. The paper does not mention how consensus mechanism and auditing processing work in Cloud based Blockchain.

The proposal might be integrated with the body area sensor network. Yue *et al.* [42] do not show how patient record can be encrypted or accessed at granular level and how Blockchain tackles stream of medical sensor data. In IoT healthcare, one key for all the users in the Blockchain raises privacy risk and one key for every individual involved in the Block chain is also not feasible because user might be still identified through inspection and analysis of available open information on the Blockchain. Moreover, the key should vary for every chunk of data block in the Blockchain.

Zhao *et al.* [35] proposed a fuzzy vault based key management in Blockchain health architecture. The architecture of the Blockchain based health framework includes wearable sensor nodes on the patient's body, some implanted nodes and gateway nodes for the body area sensor networks. The Gateway collects physiological data from wearable sensor nodes and sends aggregated data to some pointed hospital which individually make a block in Blockchain and message generated from the Gateway is considered as a single block. Wearable sensor nodes produce a key before sending physiological data to the Gateway node and encrypt the data with the key generated from the signal of patient's body. Blockchain communities or healthcare professionals cannot leak the patient information. The patient can only recover the key from her physiological data to decrypt data. However, this approach causes significant burden for power constrained medical sensors because these sensors require to construct the key from patients' physiological data during decryption.

Linn and Koo [43] planned an off Blockchain approach for health data storage called Data Lake and a Blockchain containing all authorization transactions. Data in an encrypted format is stored as key-value pairs in the data lake. The user, health data provider, and doctor use mobile apps to access data through Blockchain from the data lake. After completing analyses such as different types of medical tests, the provider inserts a signature and authenticates a user to access that result through Blockchain. In the same way, the user can offer authentication and authorization of its data to a doctor through Blockchain. They do not show the design elements of Blockchain and off Blockchain does not provide the access to the medical record that healthcare providers need.

In eHealth frameworks, the mechanism to provide the patient's data to health professionals is not highlighted in central server based architecture. On the other hand, streamed data is generated from the medical sensors in continuous patient monitoring system, the current architecture of Blockchain based healthcare also overlooks the efficient processing of huge stream of patient's data so that patients get a rapid response from healthcare professionals. There is still a need to design an End to End eHealthcare framework merging Blockchain with legacy healthcare architecture.

III. PROPOSED SECURE PATIENT MONITORING ARCHITECTURE

The proposed architecture comprises two tiers; the lower tier provides the data streaming and storage solution whereas the upper tier manages keys healthcare provider and is called Healthcare Control Unit(HCU). The lower tier includes six systems illustrated in Fig. 1. Body Area Sensor Network(BSN), Sensor Data Provider(SDP) such as smartphone, Patient Centric Agent(PCA), Blockchain, Healthcare Provider Agent(HPA) and Healthcare Provider's Wallet(HPW). In Fig. 1, BSN is connected to Patient Centric Agent(PCA) through Sensor Data Provider such as a smartphone. PCA is connected to Blockchain network, Cloud and the Healthcare Control Unit. Healthcare Provider Agent connects Blockchain, Healthcare Control Unit and Healthcare Wallet at healthcare provider end. The architecture is explained in accordance to the communication links between different segments below and the functional view of the architecture is depicted in Fig. 2. The architecture is designed to scale to large numbers of patients.

A. BODY AREA SENOSR TO SENSOR DATA PROVIDER

In this section, we discuss BSN and SDP, and mutual authentication process between these two segments.

1) BODY AREA SENSOR NETWORK(BSN)

Different types of wearable sensor devices such as motion tracker, biophysiological sign measurement devices(EEG, ECG, BSC etc.) [44] form the Body Area Sensor Network. In our architecture, we also consider n number of wireless wearable sensor devices measuring physiological signs in the Body Area Sensor Network(BSN). Devices in BSN are extremely power constrained and have very limited processing power [45]. Typically, these devices send patient's data to a nearby smartphone using the Bluetooth or ZigBee protocol [46].

2) SENSOR DATA PROVIDER(SDP)

The Sensor Data Provider(SDP) is software that executes on a mobile device, cellphone or modem. We assume that a patient has a dedicated smartphone to receive health data from medical sensors in BSN and wirelessly provides the data to the Patient Centric Agent (PCA) described in III-B. The data stream generated from a sensor is partitioned by the SDP in variable size data frame windows [47]. For instance, the size of the window is set by the SDP for each variable using simple heuristics and varies according to the fluctuations in data rates. The SDP does not perform any aggregation of data such as packaging heart rate and breath rate streams together. Further, the SDP does not pre-process data to remove outliers or abnormal values.

The SDP includes two components. A Security Service Module(SSM) performs cryptographic operations such as key generation, authentication, encryption and decryption and a

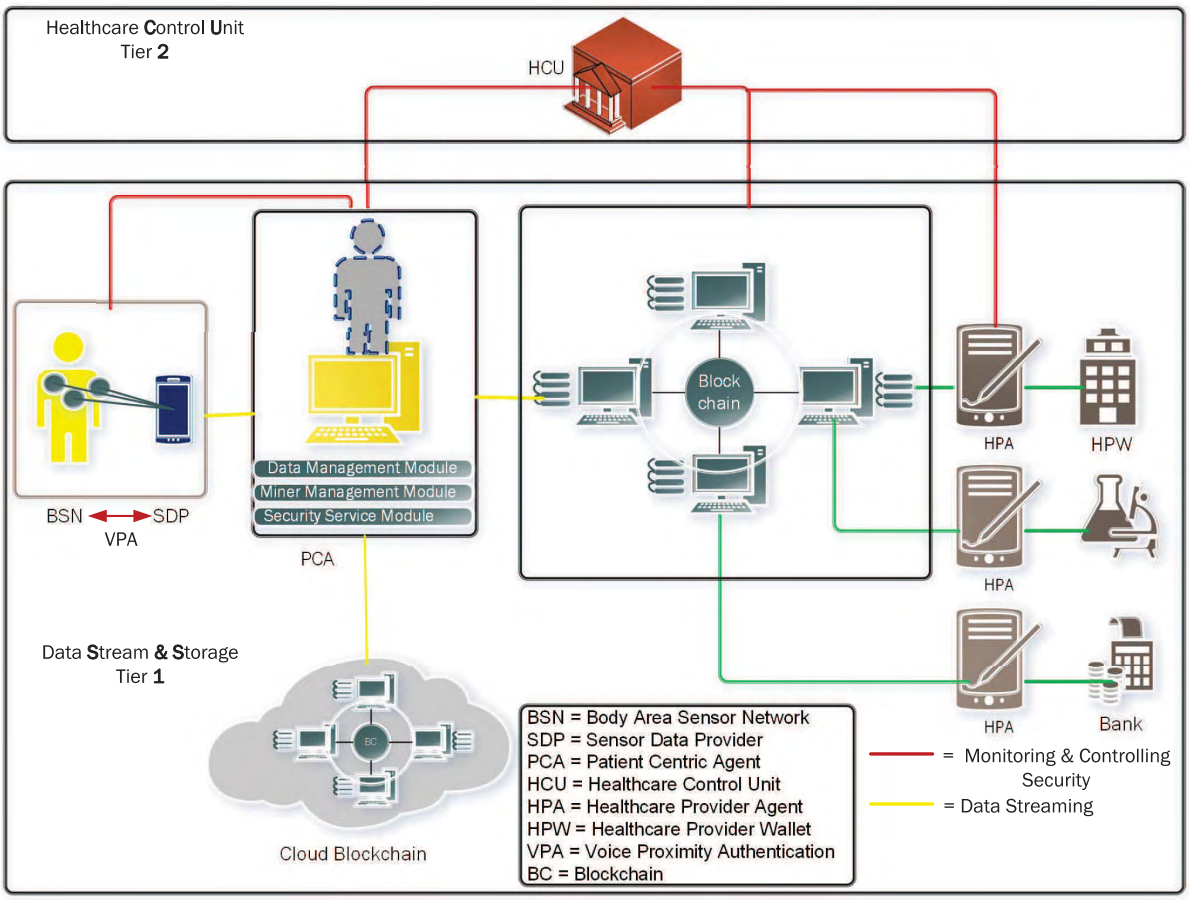

FIGURE 1. The tier based Remote Patient Monitoring architecture.

Patient Mobility Management Module continues transmitting patient's physiological data to PCA using long range communication standards such as NB-IoT [46] while patient moves to a new place.

Public/Private encryption requires devices having considerable processing power [48]. Wireless sensors in BSN lack the processing power required for public key encryption. Furthermore, confirmation of ownership's legitimacy of public/private key is not feasible without a third party trust center. Although symmetric key authentication is a promising solution for the BSN to SDP segment, key sharing is vulnerable to man in middle attack. According to Malina *et al.* [8], Proximity based User Authentication(PUA) introduced by [49] can address the key exchange challenges of IoT devices with power and memory limitations.

3) AUTHENTICATION BSN TO SDP

We design a mutual authentication approach by integrating Proximity User Authentication(PUA) and HMAC [30](keyed-Hash Message Authentication Code) for BSN to SDP channel. The mutual authentication is a two way authentication where both entities in the communication link prove their identities to one another. Entities in PUA perform their authentication based on their physical distance. In RPM healthcare system, legitimate medical sensors attached to the patient's body and smartphone in SDP are usually closer than attacker's devices. Therefore, intrusion to patient's medical sensors will not be successful thanks to attacker's position even if it can discover the legitimate device's session key that is used to produce HMAC.

Radio signal based proximity measurement [50], [51] estimates distance between entities in a communication link by exchanging radio signals. Two entities in close proximity to each other can be assumed to transmit stronger radio signals however radio signals can be easily manipulated by attackers [49]. However, voice signal introduced by [50] requires less power than radio signal to measure the proximity between two entities.

Therefore, voice signal is used in the proposed authentication process to estimate the distance between two entities and to prove their legitimate identities. Voice based Proximity(VP) has some limitations. An attacker might play a recording of the voice while person is asleep and the processing of VP in BSN device adds much computational burden on extremely power constraint medical sensor.

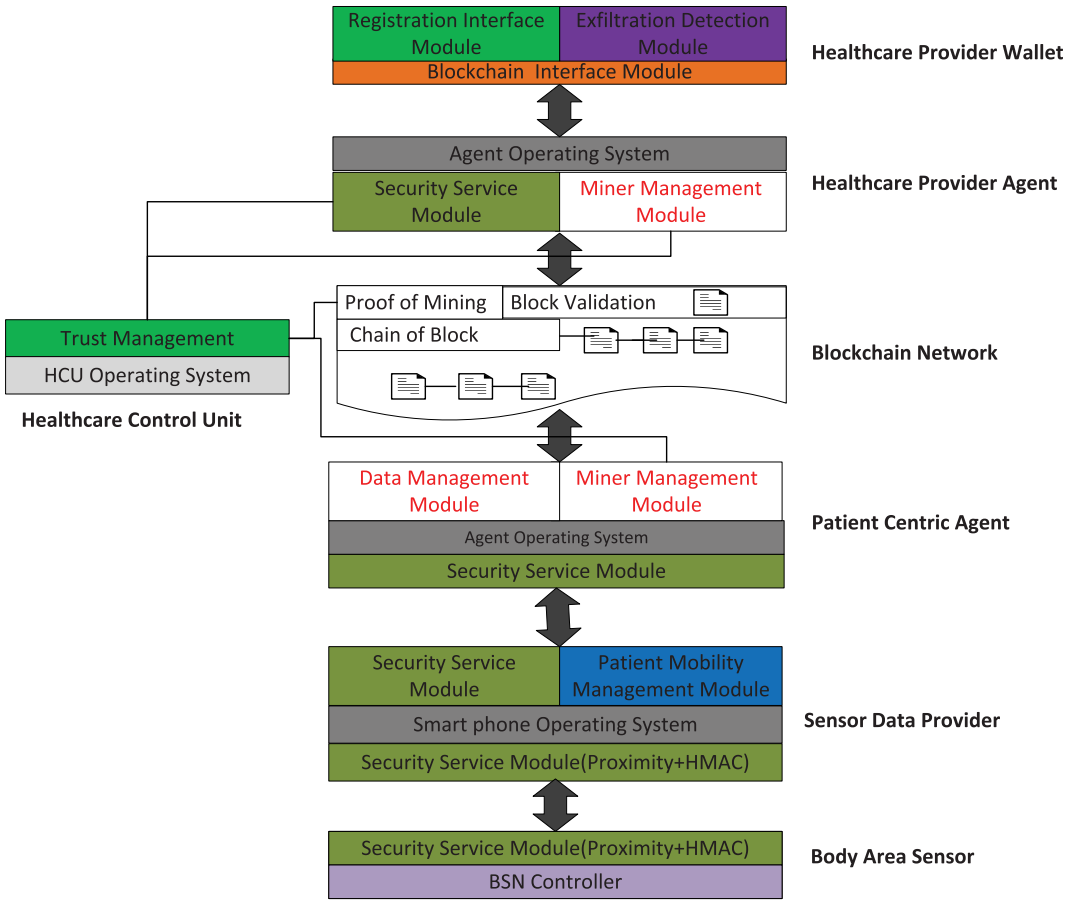

FIGURE 2. Conceptual view of the tier based health monitoring architecture.

Therefore, we presume that BSN and SDP device store the legitimate user's voice and have also a voice or audio processing unit.

a: MUTUAL AUTHENTICATION PROTOCOL

The lightweight authentication protocol confirms SDP receives physiological data from the legitimate patient wearing medical sensors. We assume that BSN and SDP devices produce a sessional symmetric key(K_i) for authentication through a dynamic key generation mechanism described in III-B3. The mutual authentication BSN to SDP is depicted in Fig. 3. Here, the authentication process starts with the BSN speaking to the SDP. H() represents Blake2 message digestion code and HMAC represents Marvin message authentication code. Pereira *et al.* [52] showed that Blake2 Hash and Marvin MAC outperform other approaches in IoT devices in terms of speed and energy. HMAC is faster and requires less computational cost in terms of processing power than public/private key pairs. The cryptographic notions and meaning are illustrated in Table 1. The mutual authentication process is described as follows:

- Firstly, BSN initiates authentication by sending the SDP device a transmission that consists of two messages: An information message consisting of time, nonce, and user voice signal; an HMAC of time, nonce, and voice signal to ensure the integrity of the flight. Here, time, nonce, and user voice signal are encrypted with a one-time pad to hide them from the attacker.

BSN $\rightarrow$ *SDP* The BSN device randomly chooses a nonce(N_s) and uses system time($Time_s$) to confirm the freshness of the authentication message. $Time_s$ guards against reply attack and nonce is also used as dynamic identities of BSN and SDP devices during transmission of health data. BSN device performs XOR($\oplus$) operation on the nonce and H() of symmetric key(K_i) to produce $X_s \leftarrow H(K_i) \oplus N_s$ that hides the nonce from attackers. Likewise, Y_s represents one time pad of voice signal and it is produced as $Y_s \leftarrow H(N_s) \oplus V_s$ where V_s represents voice stored in BSN. BSN produces an information message $I_s \leftarrow Time_s \| X_s \| Y_s$. The attacker might intercept and change information message(I_s). Therefore, BSN computes $T_s \leftarrow HMAC(K_i, I_s)$ where T_s represents the HMAC produced from I_s using a sessional symmetric key(K_i). BSN device sends I_s and T_s to SDP device. Here, T_s ensures that I_s are produced by BSN device.

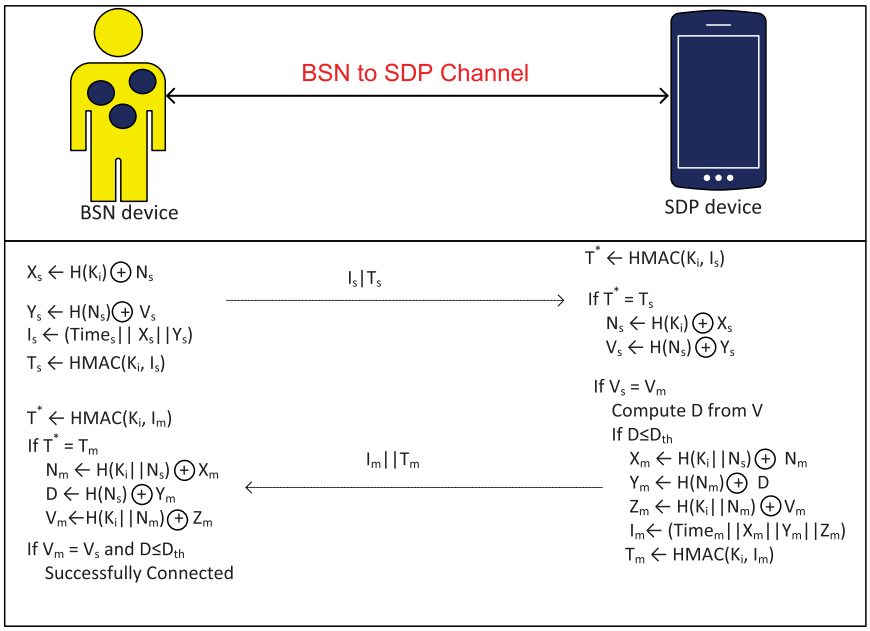

FIGURE 3. Mutual authentication BSN to SDP.

TABLE 1. The cryptography notions and meanings.

$(\oplus)$	XOR operation
H()	Blake2 Hash operation
HMAC()/MMAC()	Marvin Hash Authentication Code
K_i	Dynamically generated sessional symmetric key
$Time_s$	BSN device's system time
N_s	Nonce generated by BSN device
V_s	User voice/audio stored in BSN device
X_s	One time pad produced from BSN device's Nonce and $H(K_i)$
Y_s	One time pad produced from BSN's voice/audio and $H(N_s)$
I_s	Information message consisting of $Time_s$,X_s,Y_s from BSN
$Time_m$	BSN device's system time
N_m	Nonce generated by SDP device(mobilephone)
V_m	User voice/audio stored in SDP device
X_m	One time pad produced from SDP device's Nonce and $H(K_i\|\|N_s)$
Y_m	One time pad produced from SDP's voice/audio and $H(N_m)$
I_m	Information message consisting of $Time_m$,X_m,Y_m from SDP
$pubKey_m$	Public key of SDP device
$privateKey_m$	Private key of SDP device
$pubKey_a$	Public key of PCA
$privateKey_a$	Private key of PCA
I_a	Encrypted Information Message from PCA

- Secondly, the SDP device receives the flight from BSN device over an insecure channel and decrypts the information message provided that the verification of HMAC is successful. Next, SDP estimates the distance between the origin of the voice(BSN) and itself if the voice from BSN is identical to SDP's stored voice. After that the SDP also prepares a flight consisting of its information- Time, Nonce, Distance and its Voice, and HMAC of these information using a dynamically generated symmetric key at BSN and SDP end. The SDP also sends its flight to BSN for mutual authentication.
SDP $\rightarrow$ *BSN* SDP produces $T^* \leftarrow HMAC(K_i, I_s)$ upon receiving I_s and T_s from BSN using the same session symmetric key(K_i). SDP extracts nonce($N_s \leftarrow H(K_i) \oplus X_s$) and voice($V \leftarrow H(N_s) \oplus Y_s$) if $T* = T_s$. SDP measures distance(D)between the origin of the

voice and the SDP from voice signal(V_s) provided that $V_s = V_m$(voice stored in SDP device). If distance(D) between BSN and SDP doesn't exceed the threshold distance(D_{th})set by the PCA, then SDP also randomly chooses a nonce(N_m) and computes $X_m \leftarrow H(K_i||N_s) \oplus N_m$, $Y_m \leftarrow H(N_m) \oplus D$, and $Z_m \leftarrow H(K_i \| N_m) \oplus V_m$. SDP forms an information message $I_m \leftarrow Time_m \| X_m \| Y_m \| Z_m$ and produces $T_m \leftarrow HMAC(K_i, I_m)$ where HMAC's result of I_m using the symmetric key. SDP device sends I_m and T_m to BSN device.

- Thirdly, the BSN verifies the flight from SDP in a similar fashion.
  $BSN \rightarrow SDP$ BSN verifies T_m and extracts N_m, D and V_m to check if $V_s = V_m$.

B. SENSOR DATA PROVIDER TO PATIENT CENTRIC AGENT

The Patient Centric Agent that embeds Blockchain with SDP and BSN at the patient's end is discussed in this section. Following that, the mutual authentication process between SDP and PCA, sessional symmetric key generation and the communication protocol for BSN, SDP and PCA are discussed.

1) PATIENT CENTRIC AGENT(PCA)

The Patient Centric Agent is software that executes on a patient's laptop, desktop or a dedicated server. The patient agent node contains three modules: medical Data Management Module(DMM), Security Service Module(SSM), and Miner Management Module(MMM). The agent node plays a critical role in our architecture and the functionality of each component is described as follows:

- **Data Management Module(DMM):** Continuous patient monitoring system generates huge volumes of data [11]. However, some data is not required to be stored at all, whereas other data requires storage with strong encryption. For instance, unusual heart patterns in cardiovascular patients is likely to be clinically useful and would normally be stored. An intelligent module is needed to determine the level of storage required for each data stream [53]. The module classifies patient's data as Normal, Eventful and Uneventful following [58] and [59]. The module stores uneventful data locally and also sends uneventful patient's data to Cloud in case healthcare professionals requires the data. We do not discard any physiological signal of the patient, because even uneventful data may be useful for some future purpose such as research. Further, physiological data compression [56]–[58] that requires high computational cost is performed in DMM instead of BSN.
- **Miner Management Module(MMM):** The PCA participates in storing streamed data in a Block. The PCA, through its MMM might also act as a Miner in case no other Miners respond within a certain time. The module runs the Miner Selection Algorithm(MSA). The MMM and Cloud also form a Blockchain containing uneventful data. The module also collects network information such as availability, CPU resources about Miners in Blockchain from Healthcare Control Unit(HCU).
- **Security Service Module(SSM):** The SSM of PCA continuously analyzes the susceptibility of communication channels such as BSN to SDP, SDP to PCA and PCA to BSN to network security attack. The module excludes devices compromised by attacker in BSN and SDP. Further, the SMM periodically sends updated key generation information in BSN and SDP devices. In Blockchain, public/private key is used to hide user identity. Patient's agent might use a set of private/public key. SSM maps a person's public key into one of a few symmetric keys linked to that key and randomly uses one of the linked keys in place of the public key for indexing in Blockchain. Otherwise any Miner can follow a person's public key down the chain to discover all transactions.

2) AUTHENTICATION SDP TO PA

Proximity authentication is not appropriate for SDP device to PCA because of patient's movement. Here, we include GPS(Global Positioning System) [29] that protects data from spoofing attack.[1] A channel is considered more secure if the channel changes data encryption algorithm for a new session, because attackers do not have knowledge about channel's encryption/decryption mode. So, SDP device and PCA agree on an encryption approach from a predefined algorithm set(Triple DES, RSA, Blowfish, Twofish, AES(CBC, CTR, OCB, CCM, GCM) [60]) through authentication. But usage of different kinds of encryption algorithm at BSN to SDP channel is not feasible because of resource constraints on IoT devices in BSN. According to [52], AES-CTR is the most suitable encryption mode among AES, Curupira and Trivium for power constrained IoT devices in terms of speed and energy. BSN to SDP channel uses AES-CTR encryption mode to preserve health data confidentiality.

PCA might use several public/private key pairs. SDP device and PCA are required to validate new public/private key. The session key(K_i) is used to validate new public/private key in the authentication process so that devices don't require third party trusted center to verify public/private key.

a: MUTUAL AUTHENTICATION PROTOCOL

- Firstly, SDP device and the PCA require validating new public/private key pair using their symmetric key as this involves no third party trusted center to certify public/private key of patient's end's device. SDP initiates the authentication protocol by sending a flight formed by the encrypted public key of SDP using the symmetric key and encrypted HMAC of the encrypted public key using SDP's private key to ensure that

[1] A spoofing attack is performed by the attacker or malicious program by successfully impersonating health data on behalf of patients. ARP, DNS and IP spoofing are some example of spoofing attack.

attackers have not changed the encrypted text and the SDP has produced the HMAC.

SDP $\rightarrow$ *PCA*, we suppose that SDP or PCA has a new public/private key pair. SDP makes two encrypted text: $X_m \leftarrow Enc(K_i, pubKey_m)$ and $P_m \leftarrow Enc(privateKey_m, HMAC(K_i, X_m))$ where $pubKey_m$ and $privateKey_m$ are public and private key of a SDP device(mobile phone). X_m is the encrypted text of SDP device's public key using session key(K_i) and P_m is encrypted text of $HMAC(K_i, X_m)$ using SDP device's private key. SDP device sends the flight($X_m \| P_m$) to PCA over an insecure channel. $HMAC(K_i, X_m)$ and P_m ensure that the owner of the public key/private key is legitimate and $X_m and P_m$ have not been changed by attackers.

- The PCA receives the flight from SDP and decrypts the public key of SDP using the symmetric key(K_i), and the HMAC using SDP's public key which has been encrypted using SDP's private key. Similarly, the PCA makes a flight packing its encrypted public key using the symmetric key and an encrypted text of HMAC produced from the encrypted public key.

  PCA $\rightarrow$ *SDP* PCA decrypts X_m to obtain public key of SDP device $pubKey_m \leftarrow Dec(K_i, X_m)$). Next, PCA decrypts P_m to get $P^* \leftarrow Dec(pubKey_m, P_m)$ and verify if $P^* = HMAC(K_i, X_m)$. After that, PCA also produces two encrypted text: $X_a \leftarrow Enc(K_i, pubKey_a)$ and $P_a \leftarrow Enc(privateKey_a, HMAC(K_i, X_a))$ where $pubKey_a$ and $privateKey_a$ are public and private key of the PCA. X_a is encrypted text of PCA's public key using a session key(K_i) and P_a is encrypted text of $HMAC(K_i, X_a)$ PCA's private key. PCA sends the flight($X_a \| P_a$) to SDP device over an insecure channel. $HMAC(K_i, X_a)$ and P_a ensure that the owner of the public key/private key is legitimate and X_a and P_a has not been changed by attackers as well.
- Secondly, SDP prepares an information message that contains Time, Nonce, Data Encryption Algorithm, Location. SDP first encrypts the information message using PCA's public key, and then ciphertext is again encrypted by using the symmetric key. Two-time encryption ensure that only the legitimate PCA can decrypt the final ciphertext. The SDP produces HMAC of the ciphertext of information message and encrypts HMAC twice; first uses its private key and then public key of the PCA. The SDP transfers PCA the flight having encrypted information message and HMAC. Encryption of authentication message using both symmetric key and public/private key makes sure that an attacker cannot break the security of the authentication process without knowing both types of key.

  SDP $\rightarrow$ *PCA* The SDP(mobile phone) randomly chooses a nonce(N_m). SDP produces two encrypted text: $I_m \leftarrow Enc(K_i, Enc(pubKey_a, Time_m \| N_m \| EA \| L_m))$ that contains information of time($Time_m$), nonce(N_m), data encryption algorithm (EA), GPS location(L_m) of the SDP device and $T_m \leftarrow Enc(pubKey_a, Enc(privateKey_m, HMAC(K_i, I_m)))$ where $PubKey_m$ denotes the public key of the SDP device. First encryption in T_m using private key of SDP private key ensures that encryption is done by SDP device and second encryption in T_m using public key of PCA ensures that only PCA can decrypt and verify the encrypted text.
- The PCA decrypts an information message using its symmetric key and private key respectively. Next, it verifies the HMAC of ciphertext of information message by decrypting it using the private key of PCA and the public key of SDP as shown in flight 3 of Fig. 4. The PCA also prepares a flight with it's information message and HMAC of ciphertext of the information message.

  PCA $\rightarrow$ *SDP* PCA decrypts T_m to get $T^* \leftarrow Dec(pubKey_m, Dec(privateKey_a, T_m))$ upon receiving the flight($I_m \| T_m$) from SDP device. If $T^* = HMAC(K_i, I_m)$, the PCA decrypts information $I_m (I \leftarrow Dec(privateKey_a, Dec(K_i, I_m)))$ using a session key(K_i) and its private key($privateKey_a$) respectively. Here, attackers can only decrypt information if both session key and private key of public/private key pair are known to attackers. After that, the PCA randomly chooses a nonce(N_a) and then PCA produces two encrypted text: $I_a \leftarrow Enc(K_i, Enc(pubKey_m, Time_a \| N_a \| EA \| L_a))$ that contains information of time($Time_a$), nonce(N_a), data encryption algorithm (EA), GPS location(L_a) of the PCA and $T_a \leftarrow Enc(pubKey_m, Enc(privateKey_a, HMAC(K_i, I_a)))$ where $PubKey_a$ denotes the public key of the PCA. First encryption in T_a using private key of PCA's private key ensures that encryption is done by PCA and second encryption in T_a using public key of SDP device ensures that only legitimate SDP device can decrypt and verify the encrypted text.
- SDP verifies T_a and decrypts I_a to obtain PCA's information.

In Fig. 4, first two flights(1 & 2) between SDP and PCA occur to validate new public/private key and second two flights(3 & 4) represents sharing authentication information.

3) SESSIONAL SYMMETRIC KEY GENERATION

The exchange of symmetric keys for a session is vulnerable to man in middle attack and also causes higher communication overhead. BSN, SDP and Patient Centric Agent in our RPM architecture generate the same session key to reduce communication overhead and security vulnerability before performing authentication based on some pre-shared information by PCA. This means that devices at the patient's end do not require a key exchange mechanism for every new session.

Session Key Generation Method: The approach advanced includes a primary secret key(PSK), a linear feedback shift register(LFSR) used by [61] sequence generator with feedback polynomial $f(x_1, x_2, \ldots, x_m)$, and a hash table(T) that

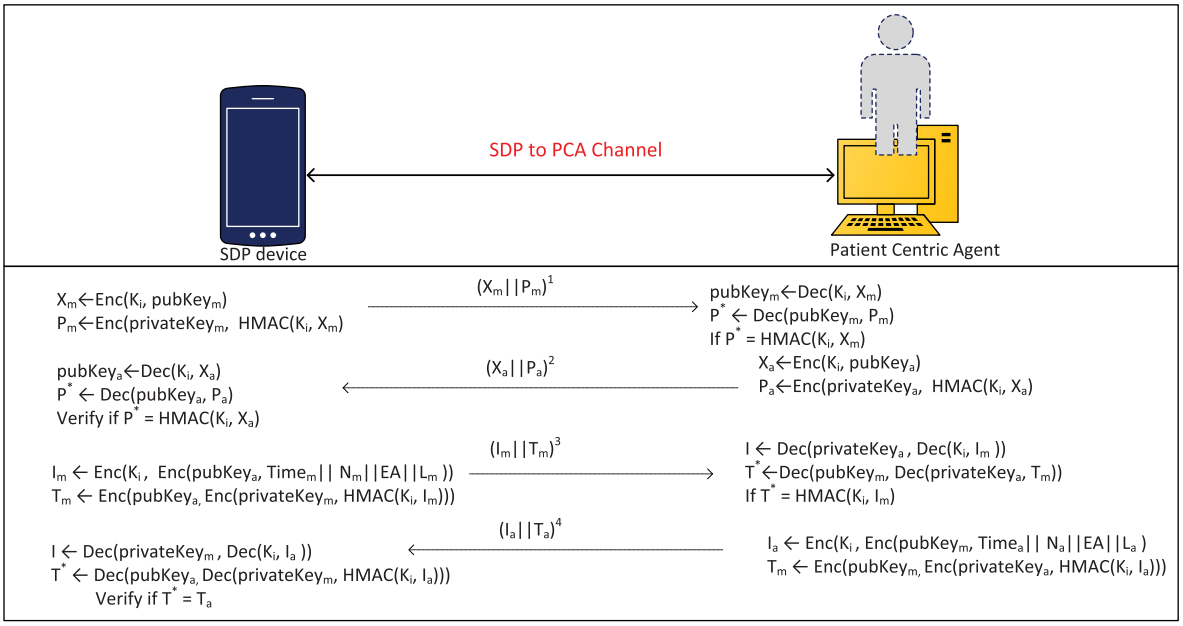

FIGURE 4. The mutual authentication SDP to PCA.

holds random numbers to generate the session key. The session key generation process is explained as follows:

- *Step 1:* Device at patient's end performs MMAC(Marvin Message Authentication Code) operation on the XOR(⊕) operation of linear feedback shift register, and previously used session key using the primary secret key(PSK). Marvin Message Authentication Code(MMAC) that has the best performance in terms of energy and speed in IoT devices is used as MAC operation [52].
- *Step 2:* Another MMAC operation is done on a random number taken from a preshared random number table using the primary secret key(PSK). Later, a sessional symmetric key is generated by performing XOR(⊕) operation on the two MMAC results; the previous MMAC from **Step 1** and this MMAC. $K_i = MMAC(PSK, f(x_1, x_2, x_3, \ldots, x_n) \oplus K_{i-1}) \oplus MMAC(PSK, r_i)$
  Where PSK is the primary secret key, K_{i-1} and K_i represents the current and previous session key respectively and r_i is a random value from a table(T) and $r_i = T[i \bmod n]$ where n is the total number of random number in table(T) and i represents the i^{th} session. Here, the random number creates immunity against rainbow table attack[2] that refers to 2^n input and output pairs pre-computed and stored in a table [62].
- *Step 3:* The hash table(T) containing random numbers is updated by applying the H() on the value of the table repeatedly if all of the random numbers have been used up.

[2]A rainbow table attack makes use of a large database that holds a large number of a hash function's input and corresponding outputs. Attacker stores plaintext and the corresponding hash of plaintext in a table to avoid generation of the hash again during looking up the hash next time.

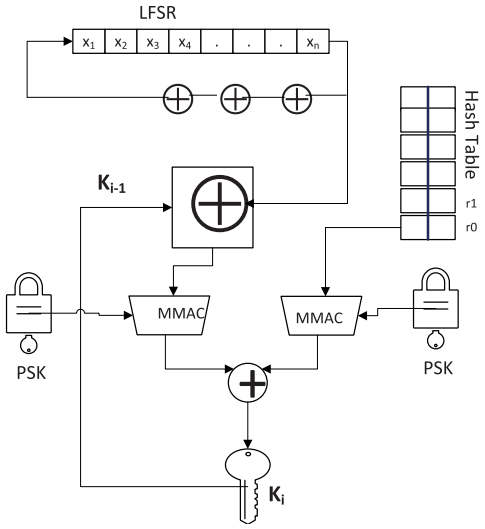

FIGURE 5. The session key generation.

The session key generation algorithm illustrated in Algorithm 1 and in Fig. 5 where the XOR operation is performed on the output of LFSR which input bit is XOR of the previous state, and the previously used session key. Next, MMAC operation is performed on the XOR output using the primary secret key. Finally, session key is generated from the XOR operation of this MMAC result and MMAC result from random value from the Table(T).

The output of LFSR was encrypted by a pre-shared key to generate one time password in [61] to monitor IoT devices

Algorithm 1 Session Key Generation Algorithm

Data: primary secret key(PSK), linear feedback shift register(LFSR), hash table(T)
Result: Dynamic Key

Calculate $K_i \leftarrow MMAC(PSK, f(x_1, x_2, x_3, \ldots, x_m) \oplus K_{i-1}) \oplus MMAC(PSK, r_i)$
if *all random numbers in T used up* **then**
  for $i = 1$ *to* n **do**
    $T[i-1] \leftarrow H(T[i] \oplus T[i-1])$
  end
end

in smarthome. But the random output from LFSR depends on the number of bits and it produces a limited random number. Therefore, we presented our approach by including a predefined random number and previously used key introduced in [63]. The dynamic session key generation presented is a lightweight process as it involves only LFSR and two MMAC operations. The MAC operation is faster and requires less processing power and memory than AES encryption [52].

4) SECURE COMMUNICATION PROTOCOL

Generation of a symmetric key during a session ensures higher security for BSN device but a fresh key for every session is computationally expensive for IoT devices. BSN device and SDP device normally assign Key Validation Time(KVT) to a new symmetric key of a medical sensor that continuously streams physiological data such as ECG. The communication protocol is depicted in Fig. 6 where the source device checks the KVT of the old symmetric key when it needs to transfer data. If the KVT is already expired, the source and destination device execute the session key generation algorithm and execute the authentication process using the new symmetric key. Otherwise, the source device executes the authentication process using the old symmetric key. If the authentication is successful, the source device prepares the data packet as shown in Table 2 and sends the packet to destination device. The source and destination device use H() of the nonce exchanged during authentication as their identification so that attackers cannot correlate session data to a BSN or SDP device.

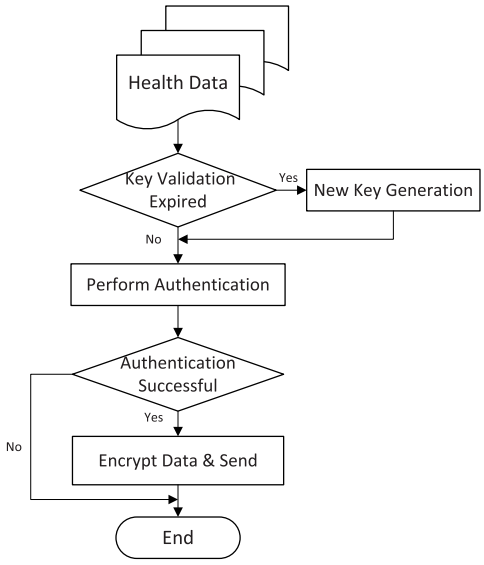

FIGURE 6. The communication protocol.

TABLE 2. The data packet format.

H(Source Nonce)	H(Destination Nonce)
Sequence Number	HMAC(K_i,H(Data|| Sequence Number))
Enc(K,Data)	

C. PATIENT CENTRIC AGENT TO CUSTOMIZED BLOCKCHAIN

The PCA connects the patient's BSN with the customized Blockchain network. The PCA decides what data is to put into the Blockchain and, which Miner is to be selected. Blockchain is not only a tamper proof distributed database for patient's record but also an authentic platform verified by all the nodes in the Blockchain. Nodes of Blockchain might be provided by healthcare providers, other organizations or individuals. The nodes in our customized Blockchain are normally classified as half nodes, general nodes, benign nodes and miner nodes like Bitcoin Blockchain. Half nodes normally indicate an individual user or a healthcare provider that would like to use data stored in the Blockchain. General nodes are responsible for storing the chain of blocks and broadcast blocks for validation. The Miners that are powerful nodes in terms of CPU processing mine a block. Miner and general nodes ensure that a data packet originates from a legitimate node using the verification process. In our RPM architecture, benign nodes can be distinguished from other nodes by the Trust Management of Healthcare Control Unit(HCU).

The Blockchain in Bitcoin demands a lot of processing power to mine a block. Further, the transaction processing time of the Bitcoin is longer to handle stream data in continuous RPM in real time. Healthcare professionals normally need to quickly retrieve streamed data from the Blockchain. These challenges motivated us to design a customized Blockchain with Patient Centric Agent to process the patient's stream data in real time. In the customized Blockchain, the patient has full control of his or her record. The following sections contain the basic components of Blockchain technology used in the proposed architecture. The components include miner selection for the proof of work, transaction and block. We first describe the Bitcoin Protocol before illustrating our customized version.

- Half nodes or general nodes make a transaction with the sender's signature and broadcast the transaction throughout the Blockchain network.
- Miner nodes gather certain amount of transactions and process transactions in a block. All miner nodes start

solving a difficult hashing problem called Proof of Work [64] by incrementing a variable field called the nonce of the block. The Miner that successfully generates the target hash containing pre-specified number of leading zeroes first broadcasts the block to Bitcoin network and receives a financial reward for doing so.
- All nodes in Blockchain verify the block and add the block to the current Blockchain.

1) MINER SELECTION IN CUSTOMIZED BLOCKCHAIN

The Proof of Work in digital cryptocurrencies consumes huge processing power because all of the miners compete to be first to generate the target hash of block to prevent the tampering of the record. Proof of Stake and Proof of Capacity or space are alternative consensus protocols used in some cryptocurrencies.

The Proof of Stake [65] does not depend on the processing power of the miners. The Miner that owns and locks the highest share of coin to the system has the higher chance to mine the next block. For example, if there are three miners namely m_1, m_2, and m_3 which own 25%, 10% and 15% share respectively, then, the first miner builds the next block.

With the Proof of Capacity or space [66] approach, the Miner with the greatest memory or disk space capacity is selected to add the block to the chain.

In our End to End healthcare architecture, we propose to select a group of trusted miners based on some characteristics and the miner selection is done by the PCA. The PCA collects resource information and ratings given by other PCAs about miner nodes from TM(Trust Management) module as shown in Fig. 7 in HCU and also send it's rating about the selected miner to TM. The PCA aggregates patient physiological data and builds a block, the block is transferred to a miner node listed in the group. The selected miner node runs Proof of Work as in Bitcoin. The process reduces the power consumption of Blockchain as one Miner produces the Target Hash.

The HCU explained in III-E executes Miner Selection Algorithm for healthcare professional's registration transaction. The traditional bank acts as a Miner on behalf of its customer for payment transactions that is discussed in Section III-C4. The Miner Selection Algorithm executed by the PCA is described in III-C1a.

a: CHARACTERISTICS BASED MINER SELECTION

We present our characteristics based Miner selection in Algorithm 2, the nonce generation for block's target hash in Algorithm 5 and block verification Algorithm in 4 respectively.

- In the proposed Miner Selection Algorithm(MSA), the PCA first obtains Miners' disk space, locked currency amount from it's NM(Network Manager) and the trust's estimation(T) from the TM of HCU. The trust's estimation(T) is discussed in III-C1.b.
- Secondly, the ratio(in percentage) of locked currency, and memory capacity of available Miners is calculated respectively.
- Thirdly, a linear equation involving locked currency and memory ratio is solved to maximize the total ratio of a miner by using linear programming subject to total ratio of the memory and currency is equal to 1 and individual share of memory or currency is equal or less than $\frac{3}{4}$.
- Fourthly, the result of the linear equation is normalized and is added with normalized trust's estimation(T) of a miner to measure the ultimate rating for the miner. Later, a set of the fittest miners are selected randomly or using hiring selection algorithm($\frac{1}{e}$ algorithm) after estimating the rating of all available miners.
- The PCA executes the algorithm to select only one miner from the set of the fittest Miners every time it has a block. In the RPM e-healthcare framework, data transaction processing rate is higher than any other Blockchain applications because of huge stream of real time data from BSN. This selection of a Miner from the list reduces the computational cost in Blockchain as well as the PCA.
- The PCA waits for a certain time after handling over the block to the selected Miner. If the PCA does not receive the block to verify as one of the validator within a pre-specified time,the PCA nominates another Miner from the fittest list. Here, the nominated Miner transmits the block to its neighbor nodes in the Blockchain network. The neighbor nodes continue to broadcast the block in the Blockchain network. In the meantime, the Miner comes up with target hash of the block and later just broadcasts the identifier, target hash and nonce of the block in the Blockchain network. The nodes in Blockchain already having the block verify the target hash and confirm the addition of the block to the patient Blockchain. The advantage of the approach is that: patient's block can be available to healthcare professionals in real-time. The disadvantage of this is that: it causes network overhead in the Blockchain. But the availability of a patient's record in the Blockchain network is a vital requirements in RPM.

b: TRUST MODEL

The PCA needs to discover reliable nodes among the available Miners as it needs to choose only one miner to perform Proof of Work. We propose a trust model as illustrated in Fig. 7 to discover the most reliable miners. The model executed by HCU ranks a miner on the basis of the rate given by those PCA(miner's client) that already selected the Miner, and summation of probability of some other trust parameters illustrated in Table 3. The PCA queries HCU to get trust's estimation (T) for a miner. The Miner's client provides the TM of HCU with feedback of the Miner regarding turn around time and mining charges information. The Miner's client that already experienced comparatively less turn around time gives a higher rating to the Miner. The Miner that voluntarily mines block is also given the highest rating.

In Fig. 7, let the miner m_1 is already picked up by some PCAs such as $A_1, A_2, A_3, \ldots, A_n$ and the normalized turn

Algorithm 2 Characteristics Based Miner Selection Algorithm

Data: currentCurrency, currrentCapacity, turst(T[]) of m numbers of available miners
Result: List of reliable Miners

Intialize $count \leftarrow 0$
if *there is no available miner* **then**
  $minerSelection[count++] \leftarrow patientAgent$
else
  for *each miner* $i = 1$ *to* m **do**
    if $currentCapacity[i] \geq Th$ **then**
      $capacity[i] \Leftarrow \frac{currentCapacity[i]}{\sum_{j=1}^{m} currentCapacity[j]} \times 100$
      $currency[i] \leftarrow \frac{currentCurrency[i]}{\sum_{j=1}^{m} currentCurrency[j]} \times 100$
    end
    Maximize $ratings[i] \leftarrow xcapacity_i + ycurrency_i$ subject to $x \leq \frac{4}{3}$, $y \leq \frac{4}{3}$ and
    $x + y = 1$
  end
  /* Normalize ratings and trust where A and B, C and D are the lowest and highest value of $ratings[i]$ and $T[i]$ respectively. $1 - R$ represents the scale of the normalization. */
  for *for each miner* $i = 0$ *to* m **do**
    $ratings[i] \leftarrow 1 + \frac{(ratings[i]-A)\times(R-1)}{(B-A)}$
    $T[i] \leftarrow 1 + \frac{(T[i]-C)\times(R-1)}{(D-C)}$
    $ratings[i] \leftarrow ratings[i] + T[i]$
  end
  $numSkip \leftarrow m \times \frac{1}{e}$
  for $t = 1$ *to numTrials* **do**
    Shuffle(ratings)
    $bestRating \leftarrow ratings[1]$
    $candidate \leftarrow -1$
    $readyToHire \leftarrow false$
    for *each miner* $i = 0$ *to* m **do**
      if $i \geq numSkip$ **then**
        $readyToHire \leftarrow true$
      end
      if $ratings[i] > bestRating$ **then**
        $candidate \leftarrow i$
        $bestRating \leftarrow ratings[i]$
        if *readyToHire=true* **then**
          break
        end
      end
    end
    if *candidate=-1* **then**
      continue;
    end
    if $ratings[candidate] \geq thresholdRating$ **then**
      $minerSelection[count++] \leftarrow candidate$
    end
  end
end

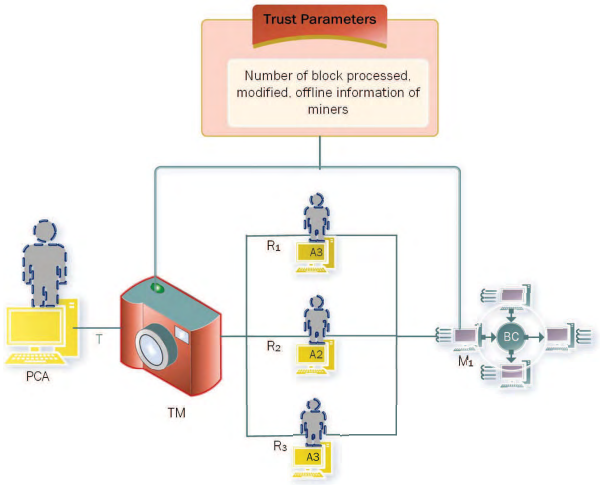

FIGURE 7. The trust model.

around time $TAT_1, TAT_2, TAT_3, \ldots, TAT_n$. The trust model is defined as in (1)

$$T_1 = d + \frac{TV_1 \times R(A_1) + TV_1 \times R(A_2) + \cdots + TV_1 \times R(A_n)}{(1-d)} \tag{1}$$

Where TV_1 is estimated by the summation of probability of the some parameters stated in the Table 3 as follows:

$TV = (1 - P(\frac{N_b^m}{N_b^p})) + (1 - P(\frac{T_offline}{24})) + P(\frac{N_b^v}{N_B})$

and $R(A_i) = WR_i + (1-W)R_0$ where $R(A_i)$ is the rating of i^{th} client PCA.

TABLE 3. The trust model parameters.

Symbol	Description
N_B	Total Number of Block within a Time Limit
N_b^p	Total Number of Block Processed by a Miner
N_b^m	Total Number of Block Modified by a Miner
$T_offline$	Total Offline Duration with 24 Hours
N_b^v	Total Number of Verified Block by a Miner
R_{TAT}	Rate from Turn Around Time
R_m	Mining Charge Rate

Here, R_i is the average rate given by i^{th} client PCA on two parameters(Turn Around Time and Mining Charge Rate) and $R_i = \frac{R_i^{TAT} + R_i^m}{2}$. The client PCA defines rating R_i^{TAT}, R_i^m from 1 to 5 according to Turn Around Time and Mining Charge Rate as shown in the Table 4.

R_0 is the average of previously obtained ratings from other client PCA and W is a weight in between $0 < W \leq 1$. If the current average is greater than the prior average then TM randomly assigns W: $\frac{3}{4} \leq W \leq 1$ and d is a probability factor and the value of d is $\frac{1}{N}$ where N is the number of available miners.

c: RANDOM MINER SELECTION

The PCA might avoid computational overhead of the characteristic based Miner selection process using Random Miner

TABLE 4. The ratings given by individual neighbor agent.

Mining Charge Rating	
Criteria	Ratings
Volunteer Mining	5
Low Mining Charge	4
Low Medium Mining Charge	3
Medium Mining Charge	2
High Mining Charge	1
Turn Around Time	
Criteria	Ratings
t_1 to t_2	5
t_3 to t_4	4
t_5 to t_6	3
t_7 to t_8	2
t_9 to t_{10}	1

Selection(RMS) policy. However, this introduces the risk that a malicious node may be nominated. The responsibility of mining a Block given to K number of Miners can make the system secure. The PCA can let a small set of Miners to compete to mine a Block unlike Bitcoin where all Miners compete. The PCA might execute RMS in case the information from HCU for characteristic based Miner Selection is not available.

Algorithm 3 Random Selection of Miner Node

Data: list of available miner node
Result: list of selected miner
for *each miner* $i = 1$ *to* k **do**
  $selectedMiner[i] \leftarrow minerList[i]$
end
srand(time(NULL))
for $i = k$ *to* m **do**
  $j \leftarrow rand()mod(i+1)$
  if $j < k$ **then**
    $selectedMiner[j] \leftarrow minerList[i]$
  end
end

d: VERIFICATION OF BLOCK

The Miner selected by PCA produces the target hash of block according to Algorithm 4. Target Hash is produced from Version (V), Type(T), Previous Block Hash(PBH), Timestamp(TS), Trie Tree Root(TTR), Target Difficulty(DT), Block Owner Address(BOA), and Transaction Time Frame(TTF) of the block. Next, the miner broadcasts the block for all other nodes in Blockchain network to verify the block according to Algorithm 5. Information of the next block verified by the nominated miner includes the previous block hash, difficulty level of target hash, legitimacy of all transactions and sender's payment. A Blockchain node adds the next block to the existing chain of Blocks if the verification process is successful.

Algorithm 4 Nonce Generation Algorithm

Data: Previous Block Hash, Difficulty Level(number of leading zero(n))
Result: Target nonce and Target tHash
Initialize $nonce \leftarrow 0$ and $target \leftarrow false$
Build Trie Tree of the Transactions
Run Transaction Fee Protocol
blockHeader$\leftarrow$ Hash$(V\|T\|PBH\|TS\|TTR\|TD\|BOA\|TTF)$
while $target = false$ **do**
  if $Hash(blockHeader\|nonce) =$ *hash with leading n number zeroes* **then**
    $target \leftarrow true$
    $tHash \leftarrow Hash(blockHeader\|nonce)$
  else
    $nonce++$
  end
end
return nonce|| tHash

Algorithm 5 Block Verification Algorithm

Data: nonce, tHash
Result: blockAcceptance
Initialize $sigStatus \leftarrow false$, $dStatus \leftarrow false$, $iStatus \leftarrow false$, $tStatus \leftarrow false$
dLevel$\leftarrow$ extractDifficulty(blockHeader(*Difficulty*))
tStaus $\leftarrow$ checkTime(*Timestamp*)
sigStatus $\leftarrow$ blockSignatureVerification()
dSatus $\leftarrow$ checkDifficulty(*dLevel*)
iStatus$\leftarrow$ checkTransactionIntegrity(*TrieTreeRoot*)
blockHash $\leftarrow$ Hash $(V\|T\|PBH\|TS\|TTR\|TD\|BOA\|TTF\|nonce)$
if $sigStatus = true \wedge blockHash = tHash \wedge dStatus = true \wedge tStatus = true \wedge iStatus = true$ **then**
  $blockAcceptance \leftarrow true$
else
  $blockAcceptance \leftarrow false$
end

2) DESCRIPTION OF TRANSACTIONS

In Bitcoin, a transaction is made when a sender wants to transfer cryptocurrencies to a recipient. Every transaction has two parts called input and output. Public/Private key pairs are used to hide the identity of the transaction owner. The sender and receiver are identified with their public key. A valid transaction has an authorized sender signature and a valid source of digital currency. The transaction format of Bitcoin is illustrated in Table 5.

Likewise, we introduce different transactions called Data Transaction(DT) for physiological data, Registration

TABLE 5. The general format of Bitcoin transaction.

Transaction Identifier			
Input		Output	
Bitcoin Source		Receiver Address	
Signature	Sender pubKey	Script	Receiver pubKey

Transaction(RT) that authorizes healthcare provider such as physician, and diagnostic center, Access Grant Transaction(AGT) for granting a healthcare provider's Role Based Access(RBA) to patient's record, and Payment Transaction(PT). The health providers send the PCA patient's other records such as prescriptions, and medical test results. Healthcare provider uses the PCA's public key for preserving the confidentiality of patient's record. The PCA has authorization only to make transactions of Blockchain for patient's records. Each type of transaction is discussed in this section.

a: DATA TRANSACTION

Data Transaction(DT) format is illustrated in Table 6. SDP device(smartphone) makes DT that consists of physiological data coming from BSN during a time interval (t_i to t_j).

TABLE 6. The format of data transaction.

TimeStamp	SensorId
SDP Address	PCA Address
MultiSignature	
Record Type	
Hash of Cipher Data	
Cipher of Data	
Transaction Fee	
CREDITLIMIT	CREDIT
CREDITPRICE	

The SDP device puts it signature in DT's Signature field and sends DT to Patient Centric Agent(PCA). The PCA signs the DT after the verification of SDP device's signature. The signature verification process is illustrated in Fig. 9

The multi-signature script introduced in [67] for the field **MultiSignature** in DT is defined as follows:

$rScript = OP\_1\ pubKey_m$
$||pubKey_a||OP\_2||OP\_CHECKMULTISIG$

Where, OP_1(n) indicates the required number of signatures among OP_2(m) numbers of signatures. $pubKey_m$ and $pubKey_a$ represent the public key of SDP device and PCA respectively.

The signature formation is illustrated in Fig. 8. A SDP device generates hash from the header of a transaction and then encrypts that hash using its private key. The signature built by a SDP device is as follows: $MultiSignature_m = Enc(privateKey_m, H(Timestamp||SensorId||SDP\ Address||Hash\ of\ Data||Transaction\ Fee))$.

The PCA checks the signature and again encrypts the signature with its private key to obtain $MultiSignature = Enc(privateKey_a, MultiSignature_m)$. The node that verifies

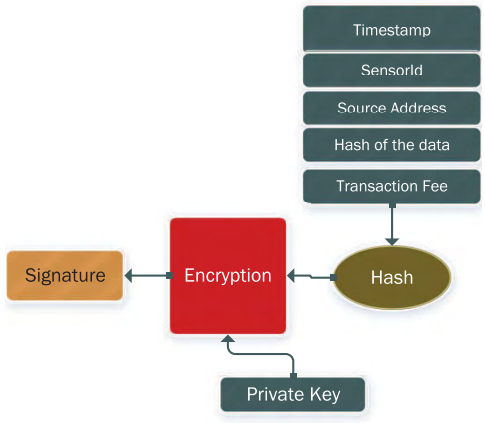

FIGURE 8. The signature formation process.

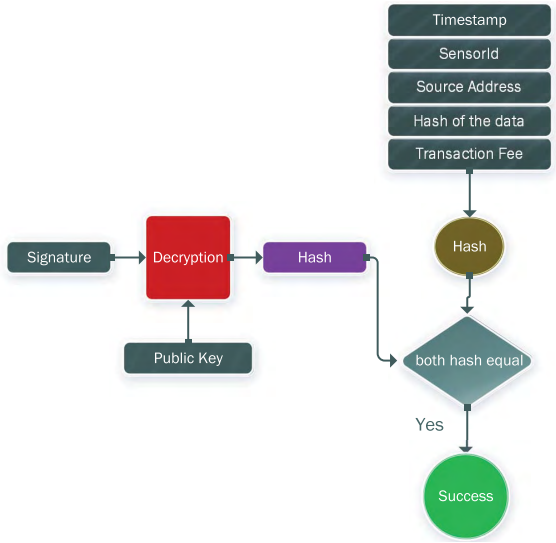

FIGURE 9. The signature verification process.

the signature first decrypts the signature using PCA's public key, next, it decrypts the hash by SDP device's public key.

The DT contains the encrypted health data called **Cipher of Data**, the hash code of the encrypted data(Hash of Cipher Data) to ensure data integrity. The **Transaction Fee** varies as every transaction doesn't carry the same amount of health data. Three fields called CREDITLIMITS, CREDIT, and CREDITRICE are used to estimate Transaction Fee. CREDITLIMITS represents the maximum amount of credits for a transaction such as Data Transaction, Registration Transaction. Further, the health data in a specific transaction such as Data Transaction might vary. So, CREDIT represent the required amount for processing of a particular transaction. CREDITPRICE represents the price per byte in a transaction. For instance, the PCA sets CREDITLIMITS(2000)and CREDITPRICE(100 Credit) for a DT with 10 bytes, the CREDIT required to process the transaction is $10 \times 100 = 200$ VCP(Virtual Credit Point).

TABLE 7. The format of registration transaction.

Record Type	Timestamp
HCU Address	
HCU Signature	
Healthcare Provider Signature	
Healthcare Provider Adress	
Healthcare Provider Profile	
Transaction Fee	
CREDITLIMIT	CREDIT
CREDITPRICE	

b: REGISTRATION TRANSACTION

The Registration Transaction(RT) format is illustrated in Table 7. RT represents the legitimate healthcare provider. The RT is issued by Healthcare Control Unit(HCU) and stored in Blockchain. The signature of HCU in RT ensures the legitimacy of healthcare provider.

c: ACCESS GRANT TRANSACTION

The Access Grant Transaction(AGT) is shown in Table 8. The PCA separates Input and Output in AGT like Bitcoin transaction. The PCA includes MultiSignature of data source in Input and a Cipher for healthcare provider to access data in Output. The Cipher includes dynamically constructed Patient Record Encryption Key(PREK) described in III-C2.d, DeviceMetaData such as Medical Sensor Id, Data Window Time Frame etc., and Time that indicates the validation period of Patient Record Encryption Key. The PCA produces the Cipher using the healthcare provider's public key so that only the legitimate healthcare provider obtains the access to health data. The AGT also contains an Access Granting Code(AGC) and Record Type explained in III-C2.d. The AGC varies based on the role of healthcare provider. For instance, the AGC for nurse is different from that of a physician.

TABLE 8. The format of grant access transaction.

Record Type	Timestamp
Access Granting Code	
Input	Output
Source Address	Destination Address
MultiSignature	Enc($pubKey_{dest}$, PREK || deviceMetadata||Time)
Transaction Fee	
CREDITLIMIT	CREDIT
CREDITPRICE	

d: PATIENT RECORD ENCRYPTION KEY GENERATION

The PCA makes Data Transaction(DT) by gathering data in predefined time frames. The challenge is to encrypt every transaction by using individual keys so that healthcare professionals can access only limited records that are assigned. One key assigned to a medical sensor might give healthcare provider access to huge records for a long time. Encryption of transaction according to its time frame window ensures fine granular access of patient's record. In addition, healthcare providers have different access levels based on their roles. The PCA is also required to construct Patient Record Encryption Key(PREK) based on Record Type(RT) and healthcare provider's role. The PCA needs to dynamically construct Patient Record Encryption Key(PREK) during processing a transaction as the storage of individual keys per transaction requires huge memory.

The PCA produces the PREK through Hash operation of its Secret Key, Sensor ID, and Time Frame that includes the date and window time frame($22-03-2018 : 10.30-10.45$) of a transaction.

$$PREK = H(PCA\ Secret\ Key||Sensor\ ID||Record\ Type||Time\ Frame)$$

Where *PREK* represents Patient Record Encryption Key for a pre-specified time frame. The PCA encrypts a transaction by using a dynamically constructed PREK. PREK can be regenerated by the PCA whenever the PCA grants a healthcare provider access to patient health records. The PCA can share its secret key with SDP and BSN so that BSN and SDP can encrypt physiological data generating PREK. As PREK involves only H() operation, generation of PREK is also feasible for BSN.

The Healthcare Provider's Wallet can only decrypt the patient's record and the HPW deletes the record after elapse of time in AGT illustrated in Table 8. In RPM, healthcare professionals deal with diverse genre of patient records such as raw records, prescription records, and diagnostic result. PCA assigns a code to a patient's record: raw record(00), prescription record(01), diagnostic result(10) etc.

The PCA defines the user of health data based on the healthcare provider'roles drawn from an eHealth standard such as openEHR [68]. This includes Healthcare Provider Organization such as Diagnostic Center(DC) or Hospital(H), Individual Healthcare Provider such as Physician(P) or Nurse(N), and Healthcare Consumer such as Relatives(R) or Others(O). Finally, the PCA merges patient's record code and selected healthcare user code to make Access Granting Code(AGC) of AGT shown in Table 8. For instance, if the PCA gives a patient's Physician, Nurse and Relatives access to medication prescription, the PCA produces the code(01-110010) illustrated in Table 9. The AGC and the HPW will reject access if someone's role does not satisfy the AGC.

TABLE 9. The access granting code.

RT	P	N	DC	H	R	O
01	1	1	0	0	1	0

3) DATA BLOCK STRUCTURE

Bitcoin Miners collect transactions from different users worldwide and build a block with 1024 transactions. The Miner creates a Merkle Tree that is a binary tree to pack the transactions in a block. The Merkle Tree ensures the integrity

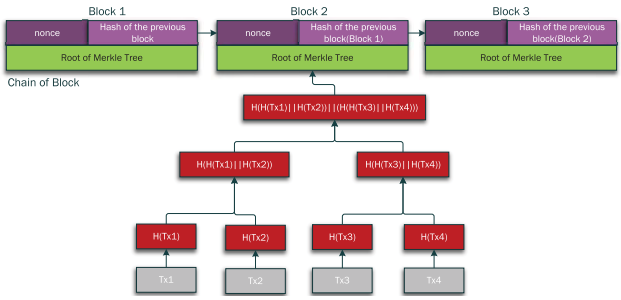

FIGURE 10. The Blockchain in Bitcoin.

of the transactions in a block. A Blockchain with Merkle Tree in Bitcoin is depicted in Fig. 10. Here, we assume that there are four transactions: $Tx1, \ldots, tx4$. To create Merkle Tree, first, a hash value for each transaction is produced. Secondly, hash function is applied on the concatenated hash value of two transactions and this process is continued until only one hash value is generated from all the transactions. The final hash value is the root of the Merkle tree. The Merkle Tree root is inserted into one field of the block. The nonce field that is also called counter is the only variable in the block. Nonce is incremented by the miner as one of the inputs of hash function until it produces a target hash of the block. The previous hash field contains the target hash value of the latest block of the blockchain. In this way, a chain of blocks is created, which protects an individual block to be tampered.

In our customized Blockchain for IoT healthcare framework, we propose a Trie Tree instead of a Merkle Tree to retrieve data quickly while maintaining the integrity of data. Data Transaction Block(DTB) consists of only physiological data transactions. Similarly, other Blocks such as Registration Transactions(RT), Access Grant Transactions(AGT) consist of respective kinds of transactions. Fields of a Data Block are depicted in Table 10. The Type in Block represents the kind of transactions (DT, RT, AGT). The PCA inserts a Trie Tree root in place of the Merkle Tree root in the Block. The Merkle Tree demands huge processing power and is not suitable for RPM because of huge volume of streamed data and takes longer time to retrieve data to preserve data integrity.

TABLE 10. The format of data block.

Block Header of Blockchain	
Field	Description
Version	Block Version Number
Type	Transaction type
Previous Block Hash	Hash of the previous block in the chain
Timestamp	Creation time of the block
Trie Tree Root	Root of the Trie tree containing transaction
Target Difficulty	The Proof-of-Work difficulty target
Nonce	A counter for the Proof-of-Work
Block Owner Address	
Transactions Timeframe	

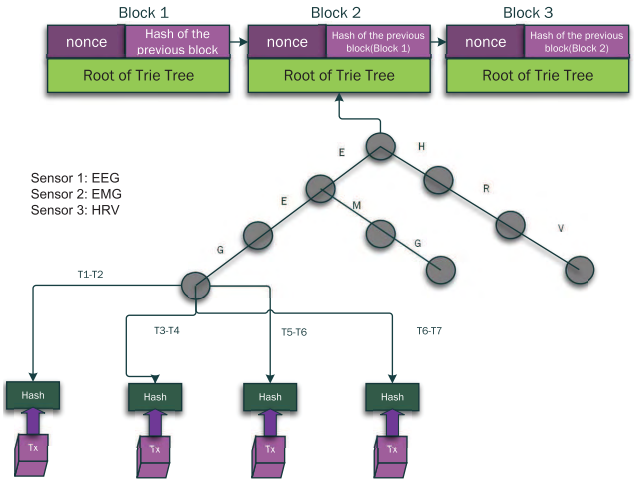

FIGURE 11. Trie tree structure.

The transaction in Block is arranged in the Trie Tree according to a device identifier. The leaf of Trie Tree holds the hash value of the transactions in the Block. In Fig. 11, we show that there are three sensor devices namely EEG,EMG, and HRV and every alphanumeric character of a sensor identifier creates a label in the tree. Transactions of a medical sensor is labeled as $T1 - T2$, $T2 - T3, \ldots, Tn - Tn + 1$ at leaves of that sensor according to transaction generated time frame window. In Fig. 11, the parent node of the leaves contain the hash value$(H(H(TX1)||H(TX2)||H(TX3)||H(TX4)\ldots))$ of the concatenation of its children hash value. Likewise, an ancestor node contains the hash value of the concatenation of its descendants' hash value as well as it's label. The significant advantage of Trie Tree lies is that; transactions can be searched at the complexity that is equal to the length(L) of a sensor identification $O(|L|)$. Trie Tree also preserves the integrity of data as parent node and leaf node contain hash value of the transaction. We can check the integrity of the transactions just by observing the root like Merkle tree. Further, Trie Tree involves fewer hash operations than Merkle Tree.

4) TRANSACTION FEE PROTOCOL

In IoT Blockchain healthcare, the management of huge streamed data is prime target to make the system efficient and effective. Blockchain in IoT healthcare also need to deal with Transaction Fee and healthcare provider's fee in a secure manner. Digital Currency(DC) in Bitcoin or Ethereum is still not as widespread as traditional currency. Therefore, we propose to incorporate the conventional banking system into our IoT Blockchain healthcare system. In the proposed payment protocol, we assume that the patient, Miner and healthcare provider own Virtual Credit Account(VCA) in the Banking system. Every node in Blockchain network is associated with one or more traditional banks. Due to security threat, traditional Smart Card/Credit Card for financial transactions are discarded in favour of virtual credit.

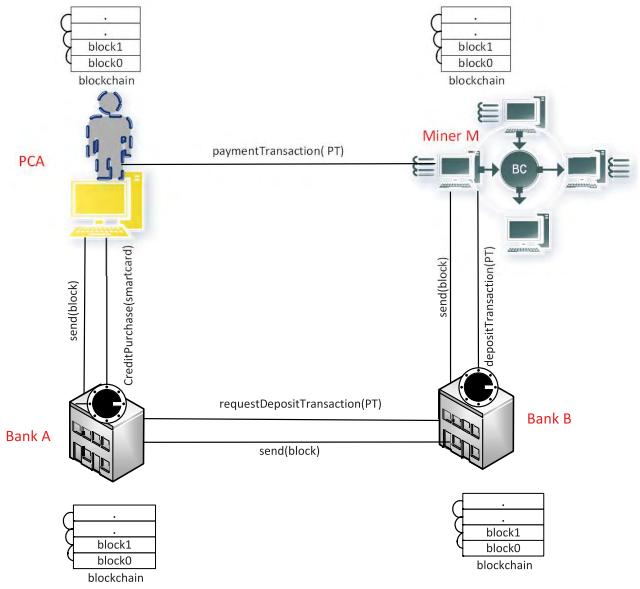

FIGURE 12. Payment protocol.

The PCA buys Virtual Credit Pints(VCP) from a bank using the patient's Smart Card/ Credit Card. The payment protocol of the proposed e-healthcare system is illustrated in Fig. 12. and is described as follows:

1) Patient Centric Agent(PCA) purchases Credit Points(CP) from Bank A in exchange for traditional currency.
2) PCA constructs a Payment Transaction(PT) as shown in Fig. 13. The PT holds PCA's signature and Bank A's signature. The PCA sends PT to Miner M nominated by the PCA for target hash generation.

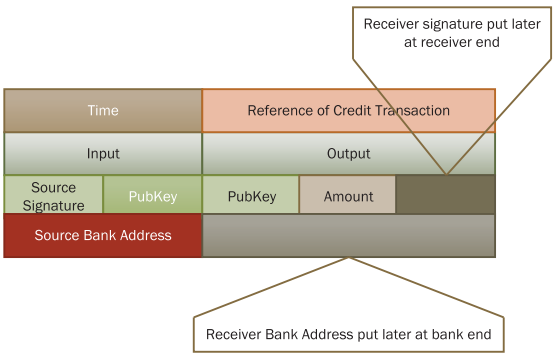

FIGURE 13. Format of Payment Transaction.

3) The Miner M puts its signature after verification of the PCA's signature and Bank A's signature in PT. The Miner M transfers the PT(Payment Transaction) to Miner's Bank B.
4) Bank B verifies all signatures on the PT and inserts its signature into the PT. The Bank B requests Bank A to make a Deposit Transaction(DT) for Miner M.
5) Bank A prepares a new transaction called UTXO(Unspent transaction) for the PCA and Output Transaction(OT) for Miner M. Finally, Bank A builds a block with all transactions produced to complete the payment and sends the block leaving the Previous Transaction Hash field empty to Miner M and the PCA.
6) The PCA and Miner M generate their respective hash value by placing the Previous Hash Block in local Blockchain. Banks in Blockchain maintain a global Blockchain for processing of UTXO and OT like Bitcoin.

D. CUSTOMIZED BLOCKCHAIN TO HEALTHCARE PROVIDER

1) HEALTHCARE PROVIDER AGENT(HPA)

Healthcare Provider Agent is healthcare provider Centric server to store and analyze patient health data. The HPA's functionalities are similar to PCA. For example, HPA nominates a Miner in Blockchain and performs Key Management at the healthcare's end etc.

2) HEALTHCARE PROVIDER WALLET(HPW)

The HPW has three modules called Blockchain Interface Module(BIM), Registration Interface Module(RIM) and Ex-filtration Detection Module(EDM) shown in Fig. 2. BIM(Blockchain Interface Module) provides the healthcare provider with Blockchain access and processes transactions for the Blockchain. EDM(Ex-filtration Detection Module) prevents insider attacker from breaching patient's information. Patient health data privacy can be breached by the healthcare service providers in attacks known as Insider Attacks. The RIM(Registration Interface Module) performs healthcare provider registration with the PCA(Patient Centric Agent) and HCU(Healthcare Control Unit) to safeguard against healthcare provider attempts to use patient's data without permission.

E. HEALTHCARE CONTROL UNIT

Healthcare Control Unit placed in the upper tier of the RPM architecture proposed here is a Trust Center for heathcare provider and the PCA. Trust Management(TM) of HCU monitors the activities of miners and the PCA in Blockchain, authorizes and certifies healthcare providers.

IV. PERFORMANCE ANALYSIS

In this section, firstly, we analysis the performance of proposed Patient Centric Agent based monitoring architecture in terms of energy and End to End delay. Secondly, we discuss the security strength of the architecture in terms of some common attacks. After that, the simulation environment and results for the performance analysis are presented.

A. END TO END ENERGY ANALYSIS

In our architecture, we allocate less processing to the body area sensor devices because of energy and processing power constraints. The sensor devices in BSN generate symmetric key in lightweight computation for authentication.

The authentication process involves HMAC operation and voice identification module which are not expensive in terms of speed and energy consumption. In the architecture, SDP device only receives physiological data from BSN and send data to the PCA. As SDP device such as smartphone is also energy constrained and memory limited, we let SDP device run only cryptography related algorithm to transfer data securely to PCA. Classification of physiological data, Block Generation, Block Verification, communication with Blockchain are some energy and processing power-hungry tasks and those are accomplished by Patient Centric Agent. This ensures more reliable processing of patient health data than traditional architecture where mobile devices normally act as coordinating node and the device has high chance to fail. Data that the PCA deems uneventful will not be stored in the Blockchain resulting in the consumption of less energy than conventional Blockchain architectures. Since, the PCA nominates only one Miner node to mine a block, the overall energy consumption of the customized Blockchain is further reduced. In addition, we propose Trie Tree based transaction packing where fewer hashing operations are required leading to further energy savings.

B. END TO END DELAY

Bitcoin processes around 3 *to* 4 and Ethereum [69] processes around 20 transactions per second. Normally, the number of processing transactions per second depends on the consensus process and difficulty level of Target Hash and Hash(Ethash, SHA-3, Blake2 etc.). In our customized Blockchain, the PCA selects only one miner. Hence, Blockchain does not demand the high difficulty level because of the absence of minor competition. So, the number of transactions per second in our Blockchain is more than that in Bitcoin. In emergency cases, the PCA can bypass the Blockchain and directly send data to an authorized healthcare provider. Later, the PCA can store the emergency data into the Blockchain. This approach also significantly improves the End to End delay of health data processing. In addition, a patient record can be quickly retrieved from the Trie Tree which also helps the minimum End to End response.

C. ATTACK ANALYSIS

1) MAN IN THE MIDDLE ATTACK

Man in the middle attack normally happens when sender and receiver exchange keys. In our architecture, we let devices at different segment come up with the same key during every session to safeguard against man in the attack.

2) REPLAY ATTACK

HMAC authentication is susceptible to a replay attack if it is not modified by some other means. An authentication protocol with time and session random number is designed to prevent an attacker from replaying.

3) EAVESDROPPING

The channel between BSN devices, SDP devices and PCA exchanges encrypted health data. So, attackers cannot modify health data after intercepting data packet. Attackers cannot gain knowledge about the source and destination from the intercepted data packet because of dynamic identification.

4) SPOOFING ATTACK

Attackers sometimes change the identity of the data owner; this is known as a spoofing attack. In our architecture, the source and destination agree on dynamic identification and GPS while performing authentication. As a result, an attacker cannot inject the wrong source address or destination address. The Mining fee discourages attackers from making a fake transaction however the attempt would be discarded anyway during the verification stages owing to invalid signature.

5) COMPROMISED KEY ATTACK

As described above, periodically generated symmetric key is used to perform authentication among devices at different segment. Attackers cannot have the key without capturing hardware and software control of the devices. BSN allows access of the device based on proximity. Therefore, attacker will not able to get unauthorized access to BSN device because of its physical location. Attacker can control all devices by compromising one device if only one shared key is used by all BSN and SDP devices. So, we consider device wise dynamic key generation. In this case, even adversary compromises one device, other devices are still protective from the attack. Moreover, the Security Service Module of PCA analyzes the network traffic from SDP and BSN to separate the affected device at patient's end.

6) DENIAL OF SERVICE ATTACK

A DoS attack cannot succeed in Blockchain because attackers cannot stop the activities of all the nodes in the Blockchain network by sending fake blocks. BSN and SDP are safe from DoS attack because the PCA blocks fake requests and all traffic goes through the PCA. In Blockchain, although the PCA and SDP devices are susceptible to denial of service attack, due to patients intervention, such attack can be mitigated.

7) PATIENT PRIVACY

Patient Centric Agent can preserve patient's privacy using public key/private key encryption in Blockchain. Blockchain processes, verifies and stores block anonymously. In Blockchain, attackers cannot link patient's prescription record to patient's relevant physiological data. As patient's identity of real-life is hidden in the system, the attacker does not benefit even if it gains some data access. Further, BSN device, SDP device and PCA communicate with each other using their sessional identifier. Consequently, session

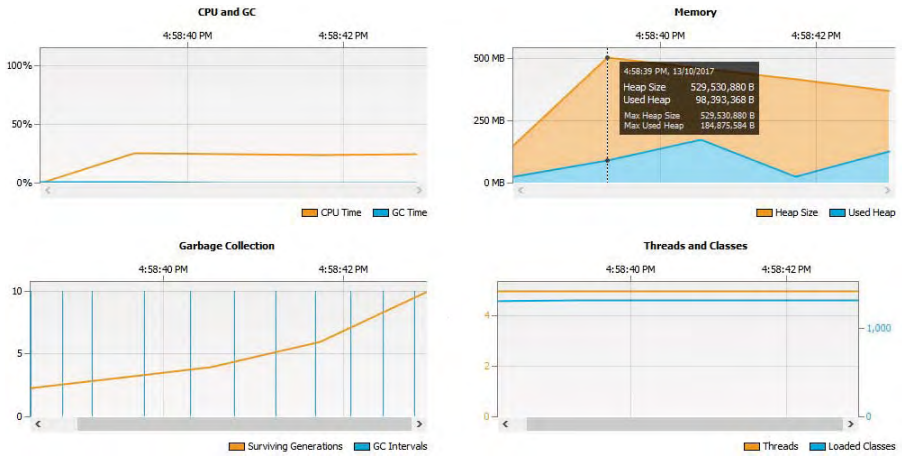

FIGURE 14. The VM Telemetry of CPMwPCA PoW in Miner 2.

identification helps patients conceal the device's real identification to attackers.

8) RELIABLE SERVICE

Our system provides reliable service for the patient. The Blockchain is a distributed ledger and open to all. Consequently, an attacker might claim to be a specialist healthcare professionals to gain the patients data or to earn money. Further, patients prefer a healthcare provider with a good reputation. So, an attacker might appear as a reputed healthcare professional. To safeguard against this, we propose Healthcare Control Unit that authorizes legitimate healthcare professionals.

D. SIMULATION ENVIRONMENT AND RESULTS

First, we discuss the simulation environment and performance of Miner Selection Algorithm executed in PCA. Later, we discuss the performance of security protocol at patient's end(BSN, SDP and PCA).

1) SIMULATION & PERFORMANCE ANALYSIS FOR MINER SELECTION ALGORITHM IN CUSTOMIZED BLOCKCHAIN

We implement the Miner Selection Algorithm executed by the PCA using Java programming environment. We use Java 8 Development Kit 64 bit and Netbeans IDE 8.1 as editor. We ran the Bitcoin proof of work on three machines specified in Table 11. Profiler of Netbeans IDE 8.1 act as performance analysis tools in our simulation. We analyze the performances of our Miner Selection Algorithm(MSA) with Ethash used in Ethereum [69] as Proof of Work and Bitcoin Proof of Work in terms of CPU time and memory. Ethash in Ethereum is faster than SHA-3 used in Bitcoin. Ethash is memory bound operation whereas SHA-3 is CPU bound operation [69]. In Java profile, we can monitor the CPU time and memory of the host machine consumed by the application program. We define the following metrics for the performance evaluation:

TABLE 11. The miner specification.

SL No	Component	Description
M_1	Processor	Intel(R)Core(TM)I3-2310M CPU@2.10 GHz 2.10
	Memory	4.00GB
M_2	Processor	Intel(R)Core(TM)I5-7200U CPU@2.50 GHz 2.71
	Memory	8.00GB
M_3	Processor	Intel(R)Core(TM)I7-4770 CPU@3.40 GHz 3.40
	Memory	16.00GB

CPU Time Monitoring represents the required amount of CPU time to execute a program. The dark line indicates the percentage of CPU usage of the specific application. The CPU time for individual method of an application is traced in the profiling tools.

Memory Monitoring indicates the amount of heap used by an application. The light portion estimates the available heap and the dark portion estimates the amount used by the dynamic objects.

Surviving Generations indicates the number of generations that are currently alive on the heap where generation means a set of instances produced between two garbage collections.

In our simulation, we consider three miners and a Patient Centric Agent as Clients-Server where all the clients act as miners send the necessary information to the PCA to calculate the ratings of the miner for the selection process. Blocks with different numbers of transactions(252, 512, 1024) is used in the simulation. The number of leading zeroes in the Target Hash was set at 6. The telemetry view of the MSA and PoW(Proof of Work) in our Continuous Patient Monitoring with a Patient Centric Agent(CPMwPCA) is illustrated in Fig. 14. The proof of work of Bitcoin is executed in three miner nodes and the telemetry view of two of those miners are illustrated in Fig. 15 and Fig. 16. It is observed that the memory and CPU utilization of the Bitcoin proof of work is higher than our MSA+PoW(the proposed miner selection algorithm and Proof of Work using Ethash utilized around

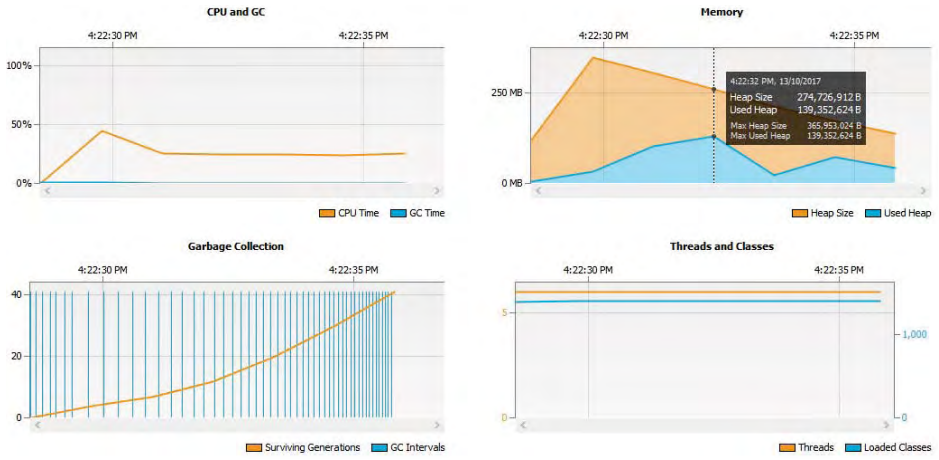

FIGURE 15. The VM Telemetry of Bitcoin PoW in Miner 2.

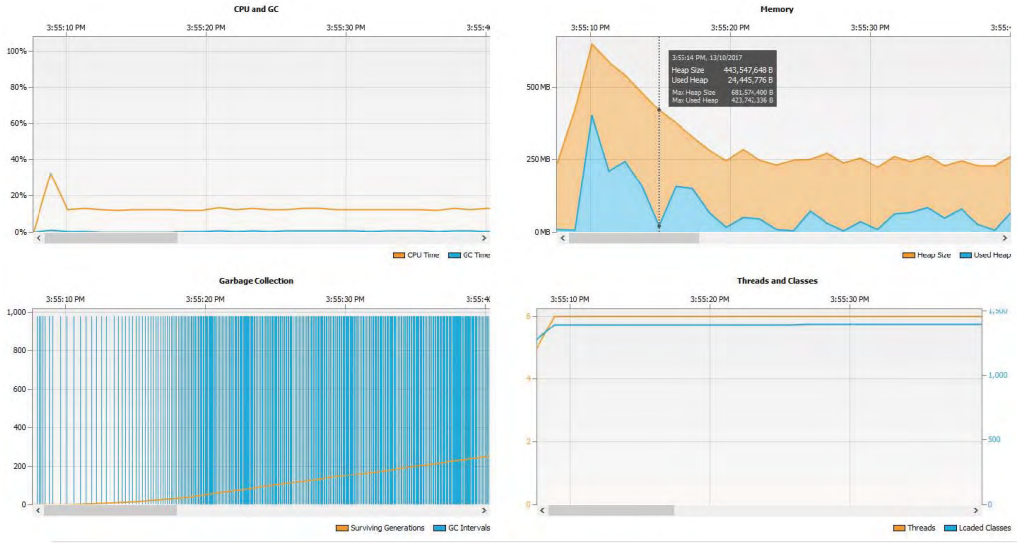

FIGURE 16. The VM Telemetry of Bitcoin PoW in Miner 3.

25% of CPU and 98MB memory whereas the average CPU utilization and memory of three miners in Bitcoin Proof of Work is around 45% and 195MB respectively). The MSA is executed for the first block. The PCA does not run MSA for the rest of the blocks. We use Trie Tree to store transactions, which incurs less cost than Merkle Tree and only one machine executes the Proof of Work. As a result, The proposed solutions significantly save power consumption of the Blockchain. Power saving is appropriate for a personalized Blockchain like remote patient monitoring where individuals, government, different institutions, and healthcare providers contribute to Blockchain's node. In Fig. 17, we show a comparison between CPU time MSA+PoW and Bitcoin Proof of Work. Our algorithm improves over the Bitcoin PoW because we select a group of miners when MSA is executed. Later, we let them mine patient data transactions one by one and Ethash are faster than SHA-3 in Bitcoin. Further,as the system allows only one miner to generate the target hash of the block, the system does not require to increase the difficulty level with the addition of new miners. Difficulty level remains constant over time.

2) PERFORMANCE ANALYSIS OF SECURITY PROTOCOL AT PATIENT END

Our security protocol at the patient's end is implemented in Intel(R) Core(TM) i5-6500CPU@3.20GHz machine by using Java. We show the comparative study of performance analysis of our protocol with ACLF [70] and BSN-Care [10] in terms of reliability, a number of error packets and throughput. The reliability of the proposed security protocol as depicted in Fig. 18 improves over the ACLF and BSN-Care because of our lightweight proximity based authentication, and multilevel storage(Patient Local Server,

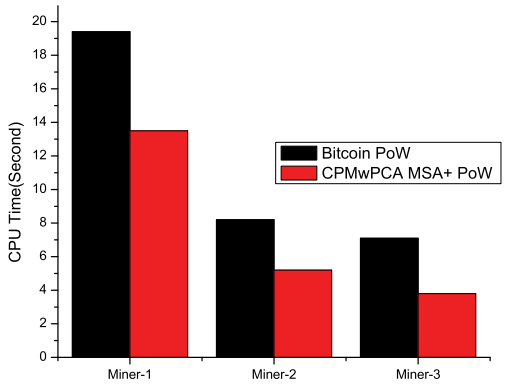

FIGURE 17. The CPU time comparison of Bitcoin PoW and CPMwPCA MSA+PoW.

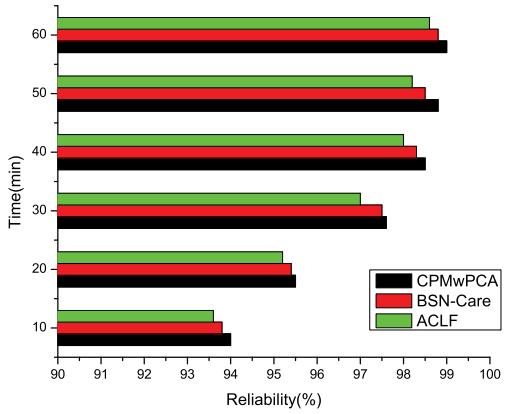

FIGURE 18. The Comparison of reliability.

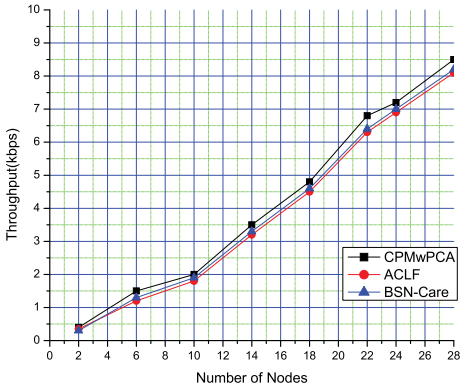

FIGURE 19. The Comparison of throughput.

Cloud, Blockchain). In ACLF, the pre-deployment of the triple key is applied for the lifetime of the sensor, therefore, if the key is exposed to attackers, adversaries might control all of the devices and hence it reduces the normal rate of transferring data in ACLF that affects the throughput illustrated in Fig. 19. In contrast, the BSN-Care proposed single server for the storage of all patients' record and therefore network congestion reduces the throughput. In CPMwCPA, we present the periodically generated key mechanism instead

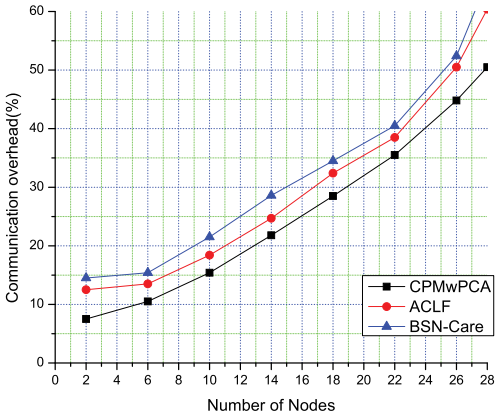

FIGURE 20. The Comparison of communication overhead.

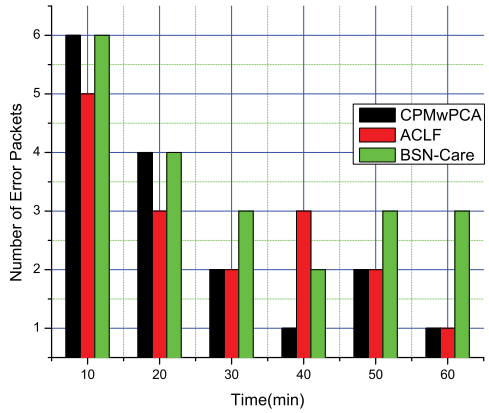

FIGURE 21. The Comparison of packet error.

of sharing information except during deployment to protect the devices from eavesdropping which reduces the communication overhead as depicted in Fig. 20. In CPMwPCA, proximity based authentication ensures SDP devices receive data from legitimate BSN devices and the probability of receiving error packet as shown in Fig. 21 is comparatively low. On the other hand, neither ACLF nor BSN-Care consider proximity authentication in their security proposal.

V. CONCLUSIONS

In this paper, we present a Patient-Centric Agent based healthcare architecture. The architecture consists of BSN, Smartphone(Sensor Data Provider), Patient Centric Agent, Blockchain, and Healthcare Provider Interface. There are multiple communication channels from End to End of this architecture such as BSN to Smartphone, Smartphone to PCA, PCA to Blockchain. Every channel requires security against different network attacks such eavesdropping, Sybil, and man in middle. Further, BSN is a power constraint network in eHealthcare architecture. High computational encryption and authentication are not appropriate for BSN network. So, our research focuses on the proposal of lightweight encryption and authentication for BSN

to Smartphone channel as well as Smartphone to PCA. Secondly, BSN produces a huge stream of data and needs to perform some pre-processing on data before sending data to Blockchain. Further, the processing rate of a block produced from real-time data might be slower than that of data arrival in Blockchain. Therefore, we focus on the development of an intelligent Patient Centric Agent that coordinates among the BSN and Smartphone. The PCA categorizes patient's data as eventful and uneventful, defines security level, controls access for patient data and generates alarms during the emergency, nominates miners in Blockchain to optimize the overall energy of the customized Blockchain. Blockchain network confirms the privacy of patient documents, tamper-proof, availability, and guards against a single point of failure. Energy and security analysis of the proposed architecture was done to demonstrate its applicability in continuous health monitoring system.

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers for their comments that improved the quality of this paper. This research was partially supported by Research Office, Federation University Australia.

REFERENCES

[1] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, Jun. 2015.

[2] L. Wang *et al.*, "A wireless biomedical signal interface system-on-chip for body sensor networks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 2, pp. 112–117, Apr. 2010.

[3] A. Marrington, D. Kerr, and J. Gammack, *Management of Security Issues in Wearable Technology*. Hershey, PA, USA: IGI Global, 2016.

[4] S. S. Khanuja, S. Garg, and I. P. Singh, "Method and apparatus for remotely monitoring the condition of a patient," U.S. Patent 12 259 905, May 7, 2009.

[5] M. Chan, D. Estève, J.-Y. Fourniols, C. Escriba, and E. Campo, "Smart wearable systems: Current status and future challenges," *Artif. Intell. Med.*, vol. 56, no. 3, pp. 137–156, 2012.

[6] R. Clarke. (1999). *Introduction to Dataveillance and Information Privacy, and Definitions of Terms*. Accessed: Oct. 10, 2017. [Online]. Available: http://www.rogerclarke.com/DV/Privacy.html

[7] P. PWC, "Managing cyber risks in an interconnected world: Key findings from the global state of information security survey 2015," Tech. Rep., 2015.

[8] L. Malina, J. Hajny, R. Fujdiak, and J. Hosek, "On perspective of security and privacy-preserving solutions in the Internet of Things," *Comput. Netw.*, vol. 102, pp. 83–95, Jun. 2016.

[9] *Health Informatics-Information Security Management in Health Using ISO/IEC 27002*, document ISO/IEC 27799, 2008.

[10] P. Gope and T. Hwang, "BSN-care: A secure IoT-based modern healthcare system using body sensor network," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1368–1376, Mar. 2016.

[11] H. Wang, D. Peng, W. Wang, H. Sharif, H.-H. Chen, and A. Khoynezhad, "Resource-aware secure ECG healthcare monitoring through body sensor networks," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 12–19, Feb. 2010.

[12] E. Al Alkeem, D. Shehada, C. Y. Yeun, M. J. Zemerly, and J. Hu, "New secure healthcare system using cloud of things," *Cluster Comput.*, vol. 20, no. 3, pp. 2211–2229, 2017.

[13] M. C. Chuah and F. Fu, "ECG anomaly detection via time series analysis," in *Proc. Int. Symp. Parallel Distrib. Process. Appl.*, 2007, pp. 123–135.

[14] H. Sivaraks and C. A. Ratanamahatana, "Robust and accurate anomaly detection in ECG artifacts using time series motif discovery," *Comput. Math. Methods Med.*, vol. 2015, Nov. 2015, Art. no. 453214.

[15] P. Ghorbanian, A. Ghaffari, A. Jalali, and C. Nataraj, "Heart arrhythmia detection using continuous wavelet transform and principal component analysis with neural network classifier," in *Proc. Comput. Cardiol.*, Sep. 2010, pp. 669–672.

[16] L. Clifton, D. A. Clifton, M. A. Pimentel, P. J. Watkinson, and L. Tarassenko, "Predictive monitoring of mobile patients by combining clinical observations with data from wearable sensors," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 3, pp. 722–730, May 2014.

[17] C. Zenger, "Physical-layer security for the Internet of Things," Doctor Eng., Univ. Bochum, Bochum, Germany, 2017.

[18] N. Ye, Y. Zhu, R.-C. Wang, R. Malekian, and L. Qiao-Min, "An efficient authentication and access control scheme for perception layer of Internet of Things," *Appl. Math. Inf. Sci.*, vol. 8, no. 4, p. 1617, 2014.

[19] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Jul. 17, 2017. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[20] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An *ad hoc* sensor network infrastructure for emergency medical care," in *Proc. Int. Workshop Wearable Implant. Body Sensor Netw.*, Boston, MA, USA, vol. 5, 2004, pp. 12–35.

[21] A. Wood *et al.*, "ALARM-NET: Wireless sensor networks for assisted-living and residential monitoring," Dept. Comput. Sci., Univ. Virginia, Charlottesville, VA, USA, Tech. Rep. CS-2006-13, 2006, p. 17, vol. 2.

[22] S. Pai *et al.*, "Transactional confidentiality in sensor networks," *IEEE Secur. Privacy*, vol. 6, no. 4, pp. 28–35, Jul./Aug. 2008.

[23] P. Kumar and H.-J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: A survey," *Sensors*, vol. 12, no. 1, pp. 55–91, 2012.

[24] J. W. Ng *et al.*, "Ubiquitous monitoring environment for wearable and implantable sensors (UbiMon)," in *Proc. Int. Conf. Ubiquitous Comput. (Ubicomp)*, 2004, pp. 1–2.

[25] R. Chakravorty, "A programmable service architecture for mobile medical care," in *Proc. 4th Annu. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2006, pp. 1–5.

[26] K. Malasri and L. Wang, "SNAP: An architecture for secure medical sensor networks," in *Proc. 2nd IEEE Workshop Wireless Mesh Netw. (WiMesh)*, 2006, pp. 160–162.

[27] J. Ko *et al.*, "MEDiSN: Medical emergency detection in sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 10, no. 1, p. 11, 2010.

[28] S. R. Moosavi *et al.*, "End-to-end security scheme for mobility enabled healthcare Internet of Things," *Future Gener. Comput. Syst.*, vol. 64, pp. 108–124, Nov. 2016.

[29] K.-H. Yeh, "A secure IoT-based healthcare system with body sensor networks," *IEEE Access*, vol. 4, pp. 10288–10299, 2016.

[30] H. Krawczyk, R. Canetti, and M. Bellare, *HMAC: Keyed-Hashing for Message Authentication*, document RFC 2104, Feb. 1997.

[31] H. Yang and V. A. Oleshchuk, "A dynamic attribute-based authentication scheme," in *Proc. Int. Conf. Codes, Cryptol., Inf. Secur.*, 2015, pp. 106–118.

[32] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Secure and scalable cloud-based architecture for e-health wireless sensor networks," in *Proc. 21st Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul./Aug. 2012, pp. 1–7.

[33] X. Liang, X. Li, R. Lu, X. Lin, and X. Shen, "Enabling pervasive healthcare with privacy preservation in smart community," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 3451–3455.

[34] T.-T. Kuo, H.-E. Kim, and L. O. Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *J. Amer. Med. Inform. Assoc.*, vol. 24, no. 6, pp. 1211–1220, 2017.

[35] H. Zhao, Y. Zhang, Y. Peng, and R. Xu, "Lightweight backup and efficient recovery scheme for health blockchain keys," in *Proc. IEEE 13th Int. Symp. Auto. Decentralized Syst. (ISADS)*, Mar. 2017, pp. 229–234.

[36] J. D. Halamka and A. Ekblaw, "The potential for blockchain to transform electronic health records," *Harvard Bus. Rev.*, vol. 3, 2017.

[37] Chet Stagnaro. *White Paper: Innovative Blockchain Uses in Health Care*. Accessed: Apr. 1, 2018. [Online]. Available: www.freedassociates.com

[38] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proc. 2nd Int. Conf. Internet Things Design Implement.*, 2017, pp. 173–178.

[39] J. Zhang, N. Xue, and X. Huang, "A secure system for pervasive social network-based healthcare," *IEEE Access*, vol. 4, pp. 9239–9250, 2016.

[40] Boosts Health. (2017). *The Future of Digital Health*. Accessed: Sep. 9, 2017. [Online]. Available: https://www.bowheadhealth.com

[41] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017.
[42] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *J. Med. Syst.*, vol. 40, no. 10, p. 218, 2016.
[43] L. A. Linn and M. B. Koo, "Blockchain for health data and its potential use in health it and health care related research," in *Proc. ONC/NIST Blockchain Healthcare Res. Workshop*, Gaithersburg, MD, USA, 2016, pp. 1–10.
[44] M. Haghi, K. Thurow, and R. Stoll, "Wearable devices in medical Internet of Things: Scientific research and commercially available devices," *Healthcare Inform. Res.*, vol. 23, no. 1, pp. 4–15, 2017.
[45] D. Wu, B. Yang, H. Wang, D. Wu, and R. Wang, "An energy-efficient data forwarding strategy for heterogeneous WBANs," *IEEE Access*, vol. 4, pp. 7251–7261, 2016.
[46] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.
[47] S. J. Preece, J. Y. Goulermas, L. P. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors—A review of classification techniques," *Physiol. Meas.*, vol. 30, no. 4, p. R1, 2009.
[48] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Gener. Comput. Syst.*, vol. 78, pp. 956–963, Jan. 2018.
[49] N. Z. Gong *et al.* (2017). "PIANO: Proximity-based user authentication on voice-powered Internet-of-Things devices." [Online]. Available: https://arxiv.org/abs/1704.03118
[50] H. Shafagh and A. Hithnawi, "Poster: Come closer—Proximity-based authentication for the Internet of Things," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 421–424.
[51] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang, "Proximity based IoT device authentication," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.
[52] G. C. Pereira, R. C. Alves, F. L. Da Silva, R. M. Azevedo, B. C. Albertini, and C. B. Margi, "Performance evaluation of cryptographic algorithms over IoT platforms and operating systems," *Secur. Commun. Netw.*, vol. 2017, Aug. 2017, Art. no. 2046735.
[53] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1082–1090, Aug. 2008.
[54] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.
[55] A. Darwish and A. E. Hassanien, "Wearable and implantable wireless sensor network solutions for healthcare monitoring," *Sensors*, vol. 11, no. 6, pp. 5561–5595, 2011.
[56] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2456–2466, Sep. 2011.
[57] S. Li, L. Da Xu, and X. Wang, "A continuous biomedical signal acquisition system based on compressed sensing in body sensor networks," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1764–1771, Aug. 2013.
[58] A. M. R. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ECG and EMG wireless biosensors," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 2, pp. 156–166, Apr. 2012.
[59] D. Watkins. *Scrypt Mining With ASICs*. Accessed: Oct. 12, 2017. [Online]. Available: http://davidwatkinsvalls.com/files/2014_spring_SMWA.pdf
[60] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A survey of lightweight-cryptography implementations," *IEEE Design Test Comput.*, vol. 24, no. 6, pp. 522–533, Nov./Dec. 2007.
[61] T. Li, J. Ren, and X. Tang, "Secure wireless monitoring and control systems for smart grid and smart home," *IEEE Wireless Commun.*, vol. 19, no. 3, pp. 66–73, Jun. 2012.
[62] K. Theocharoulis, I. Papaefstathiou, and C. Manifavas, "Implementing rainbow tables in high-end FPGAs for super-fast password cracking," in *Proc. Int. Conf. Field Program. Logic Appl. (FPL)*, Aug. 2010, pp. 145–150.
[63] H. H. Ngo, X. Wu, P. D. Le, C. Wilson, and B. Srinivasan, "Dynamic key cryptography and applications," *IJ Netw. Secur.*, vol. 10, no. 3, pp. 161–174, 2010.
[64] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A brief survey of cryptocurrency systems," in *Proc. 14th Annu. Conf. Privacy, Secur. Trust (PST)*, Dec. 2016, pp. 745–752.
[65] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
[66] S. Park, K. Pietrzak, J. Alwen, G. Fuchsbauer, and P. Gaži, "Spacecoin: A cryptocurrency based on proofs of space," IACR Cryptol. ePrint Arch., Tech. Rep., 2015.
[67] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Dependable Secure Comput.*, to be published.
[68] *Clinical Knowledge Manager, OpenEHR Clinical Knowledge Manager*, Ocean Informatics, 2015.
[69] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," White Paper, 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper
[70] V. Balasubramanian, D. B. Hoang, and T. A. Zia, "Addressing the confidentiality and integrity of assistive care loop framework using wireless sensor networks," in *Proc. 21st Int. Conf. Syst. Eng. (ICSEng)*, Aug. 2011, pp. 416–421.

MD. ASHRAF UDDIN received the B.Sc. and M.S. degree in computer science and engineering from the University of Dhaka, Bangladesh. He is currently pursuing the Ph.D. degree with the Faculty of Science and Technology, Federation University Australia. His research interests include the area of security and privacy in remote patient monitoring, blockchain, modeling, analysis, and optimization of protocols, artificial neural network, data mining, and wireless sensor network.

ANDREW STRANIERI is currently a Researcher with the Centre for Informatics and Applied Optimisation, Federation University Australia. His research in health informatics spans data mining in health, complementary and alternative medicine informatics, telemedicine, and intelligent decision support systems. He has authored over 229 peer-reviewed journal and conference articles and has published two books.

IQBAL GONDAL was the Director of ICT strategy with the Faculty of IT, Monash University. He is currently the Director of the Internet Commerce Security Laboratory (ICSL), Federation University Australia, where he conducts research in the application of advance analytics techniques for cybersecurity and provides innovative cybersecurity solutions to the industry. He is also the Non-Executive Director of the Oceania Cyber Security Centre and University engagement for the Defence Science Institute. He has served in the capacity of Director of Postgraduate studies for six years, member of faculty board, and member of Monash academic board. He has published over 164-refereed conference and journal papers. To date, he has successfully supervised 18 Ph.D. students. He is a fellow of the Institute of Engineers Australia and a Graduate Member of the Australian Institute of Company Directors. He is a member of the Advisory Board of the *International Journal for Distributed Sensor Networks* and an Editor of the *Journal of Information Processing in Agriculture*, China. His research interests are remote condition monitoring, wireless and sensor networks information processing, and cyber security analytics.

VENKI BALASUBRAMANIAN received the Ph.D. degree in body area wireless sensor network (BAWSN) for remote healthcare monitoring applications. He is the Pioneer in building (pilot) remote healthcare monitoring application (rHMA) for pregnant women for the New South Wales Healthcare Department. His research establishes a dependability measure to evaluate rHMA that uses BAWSN. His research opens up a new research area in measuring time-critical applications. He contributed immensely to eResearch software research and development that uses cloud-based infrastructure and a core member for the project sponsored by Nectar âĂŞ Australian research cloud provider. His contribution to both research and development in healthcare applications, sensor networks, and cloud computing is evident with his recent publications and source codes.