IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING ALGORITHMS FOR INTERNET OF MEDICAL THINGS

Received May 3, 2020, accepted May 16, 2020, date of publication May 20, 2020, date of current version June 3, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2995917*

# BAKMP-IoMT: Design of Blockchain Enabled Authenticated Key Management Protocol for Internet of Medical Things Deployment

**NEHA GARG[1], MOHAMMAD WAZID[1], (Senior Member, IEEE), ASHOK KUMAR DAS[2], (Senior Member, IEEE), DEVESH PRATAP SINGH[1], JOEL J. P. C. RODRIGUES[3,4], (Fellow, IEEE), AND YOUNGHO PARK[5], (Member, IEEE)**

[1]Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India
[2]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[3]PPGEE, Federal University of Piauí (UFPI), Teresina 64049-550, Brazil
[4]Instituto de Telecomunicações, 1049-001 Lisboa, Portugal
[5]School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Youngho Park (parkyh@knu.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning under Grant 2017R1A2B1002147, in part by the BK21 Plus Project funded by the Ministry of Education, South Korea, under Grant 21A20131600011, in part by Fundação para a Ciência e Tecnologia/Ministério da Ciência, Tecnologia e Ensino Superior (FCT/MCTES) through National Funds and when applicable Co-Funded European Union (EU) Funds under Project UIDB/EEA/50008/2020, in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 309335/2017-5, and in part by the Ripple Centre of Excellence Scheme, Centre of Excellence (CoE) in Blockchain, IIIT Hyderabad, India, under Grant IIIT/R&D Office/Internal Projects/001/2019.

**ABSTRACT** The Internet of Medical Things (IoMT) is a kind of connected infrastructure of smart medical devices along with software applications, health systems and services. These medical devices and applications are connected to healthcare systems through the Internet. The Wi-Fi enabled devices facilitate machine-to-machine communication and link to the cloud platforms for data storage. IoMT has the ability to make accurate diagnoses, with fewer mistakes and lower costs of care. IoMT with smartphone applications permits the patients to exchange their health related confidential and private information to the healthcare experts (i.e., doctors) for the better control of diseases, and also for tracking and preventing chronic illnesses. Due to insecure communication among the entities involved in IoMT, an attacker can tamper with the confidential and private health related information for example an attacker can not only intercept the messages, but can also modify, delete or insert malicious messages during communication. To deal this sensitive issue, we design a novel blockchain enabled authentication key agreement protocol for IoMT environment, called BAKMP-IoMT. BAKMP-IoMT provides secure key management between implantable medical devices and personal servers and between personal servers and cloud servers. The legitimate users can also access the healthcare data from the cloud servers in a secure way. The entire healthcare data is stored in a blockchain maintained by the cloud servers. A detailed formal security including the security verification of BAKMP-IoMT using the widely-accepted Automated Validation of Internet Security Protocols and Applications (AVISPA) tool is performed to demonstrate its resilience against the different types of possible attack. The comparison of BAKMP-IoMT with relevant existing schemes is conducted which identifies that the proposed system furnishes better security and functionality, and also needs low communication and computational costs as compared to other schemes. Finally, the simulation of BAKMP-IoMT is conducted to demonstrate its impact on the performance parameters.

**INDEX TERMS** Blockchain, Internet of Medical Things (IoMT), authentication, key management, security, simulation.

## I. INTRODUCTION

Internet of Medical Things (IoMT) is an assortment of health care systems (medical devices, software applications and services etc.) to provide secure transmission of health related data between smart devices which help the remotely located doctors, care-providers, medical test centers to store and exchange health data electronically. It further provides real time medical services and assistance through Internet enabled smart devices like smart phones, smart medical wearable

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

and implanted devices, electronic medical reports (EMR) etc. Other benefits of IoMT includes reduced health care costs, provides timely medical responses, fast decision making and improved quality of medical treatment. Presently, billions of devices are connected to Internet of Things (IoT) for various verticals of applications including the healthcare. With this exponential growth, user's privacy and security becomes most challenging issues in IoT (especially in IoMT) and requires essential consideration. Several potentials for security and privacy breaches that might occur in the healthcare systems are unauthorised access to enormous patients sensitive data (i.e., personal and medical records) that helps in making life critical decisions. Other malicious activities are modification of health data, hijacking of medical devices, gaining access to hospitals networks and exploitation of exchanged and stored information are performed to threat the lives of the patients. This necessitates to explore optimized solutions to deal with the threats and attacks on IoMT. Among several security mechanisms blockchain has the potentials to conquer the impediments of conventional ways to deal with the security and user privacy and is considered to be the backbone of future IoT (i.e., IoMT) applications with various benefits like enhanced security, reduced cost, true traceability, improved speed and efficient mechanism etc. Therefore, integrating Blockchain with IoMT can provide resilience to several attacks related to ''user authentication problem & key management'' problem such as ''replay'', ''man-in-the-middle'', ''impersonation'', ''password guessing'', ''illegal session key computation'', ''health data disclosure'', ''denial of service (DoS)'', ''privileged insiders'', etc., and in turn provides better healthcare services in a secured and real time environment.

### A. MOTIVATION

As discussed earlier IoMT communication environment is vulnerable to various types of attacks. Therefore, we need some strong mechanism to secure the communication occur in an IoMT environment. For such purpose, the mechanism of blockchain will be helpful as it is decentralized and temper-proof can resist different types of attacks. Hence, a secure, blockchain-enabled authenticated key management protocol for IoMT is proposed. Moreover, in the designing of the proposed scheme, we consider a private blockchain. This is primarily because the healthcare data are typically strictly confidential and private. Therefore, if we put the confidential and private health related information on a public or even on a hybrid blockchain, it will raise privacy issues. It is then recommended to consider a private blockchain in such an environment.

### B. CONTRIBUTIONS OF PROPOSED WORK

The contributions of this paper are manyfold:

- We propose a novel blockchain-based authentication and key management scheme for IoMT environment, called BAKMP-IoMT. Moreover, the private blockchain is considered in the designing of BAKMP-IoMT. In BAKMP-IoMT, we only incorporate efficient one-way cryptographic hash function and bitwise *XOR* operations.
- The formal security verification of BAKMP-IoMT using the widely-accepted AVISPA tool is performed to demonstrate its resilience against various known passive/active attacks, and also through non-mathematical (informal) security analysis we show that BAKMP-IoMT is robust against other attacks, such as ''replay attack'', ''man-in-the-middle attack'', ''impersonation attacks'', ''Ephemeral Secret Leakage (ESL) attack'', ''privileged-insider attack'', ''physical medical devices capture attack'' and ''data modification attack''. Moreover, BAKMP-IoMT also preserves ''anonymity'' and ''untraceability'' properties.
- The comparison of BAKMP-IoMT with relevant existing schemes is conducted, which identifies that the proposed BAKMP-IoMT furnishes better security and functionality features, and low communication and computational costs as compared to those schemes.
- Finally, the simulation of BAKMP-IoMT is conducted to demonstrate its impact on the performance parameters.

### C. PAPER STRUCTURE

The discussion on the associated existing schemes in healthcare applications is given in Section II. The system models (network and threat) are discussed in Section III. Various phases associated with the proposed ''blockchain based authentication and key management scheme for Internet of Medical Things (BAKMP-IoMT)'' have been discussed in Section IV. The security analysis of BAKMP-IoMT is carried out in Section V. To make further security analysis strong, the detailed formal security verification using AVISPA tool has been provided in Section VI. The comparative performance analysis of BAKMP-IoMT with relevant existing authentication schemes is also provided in Section VII. Moreover, the practical demonstration of BAKMP-IoMT using the strategies of blockchain is given in Section VIII. Finally, we conclude the work in Section IX.

## II. RELATED WORK

In recent years, several authentication and access control mechanisms have been designed in Internet of Things (IoT), wireless sensor networks, healthcare, and other applications [1]–[23].

Monrat *et al.* [24] surveyed the potential applications of blockchain in diverse disciplines like healthcare, voting, energy trading stock exchange, insurance, education etc. They also discussed the opportunities, trade off and challenges of blockchain in respective domains. Moreover, the taxonomy and architecture of blockchain was provided. Furthermore, several other approaches to reach a consensus in the blockchain was surveyed. Some future research directions of the domain were also highlighted.

Zheng *et al.* [25] discussed the concepts and characteristics of blockchain and identifies it's benefits in various domains

apart from ''bitcoin''. Further, the authors discussed a number of technical challenges like scalability, privacy leakage, selfish mining associated with consensus algorithms ''Proof of Work (PoW)'' and ''Proof of Stake (PoS)''. Their study also focused on state-of-art of blockchain including recent growth and future directions. Further, they planned to work on smart contract in a detailed manner to overcome associated defects and limitations.

Aggarwal *et al.* [26] discussed the usage of blockchain in ''smart communities'' like ''smart cities'' and ''smart nations'' where different IoT devices are located in different geographical regions. Processes of block creation and block verification, various consensus mechanisms, cryptographic primitives are also discussed. Further, classification of the application realm in cyber security is provided in an extensive manner. For example, financial systems, intelligent transportation systems, IoT, smart grid, healthcare networks, voting systems, data center networks etc. Thereafter they discussed various process models (behavior model, government sector, business model) where clusters are formed by grouping the similar process and thus provided data security using blockchain technology. Moreover they provided the information regarding communication infrastructure support like wired and wireless networks (5G, Wifi, SDN, Mobile computing).

Lin *et al.* [27] discussed the benefit of smart homes and other smart facilities for the society. At the same time they also focused on the risks associated with the remotely accessed and controlled IoT devices that could be exploited for malicious purpose. Thus they analyzed the importance of a secure and efficient remote user authentication and presented a system. For example, ''Homechain'' which integrates blockchain, group signatures and message authentication code.

Zhang *et al.* [28] discussed the issues in smart grid like secure communication, reliable mutual authentication and privacy credentials, key management etc., and considered key management between smart meters, the most critical aspect of smart grid. They presented a decentralized keyless signature scheme using consortium blockchain which is computationally cost effective, time effective, scalable, robust and efficient.

Wang *et al.* [29] discussed about the ''Internet of Vehicles (IoV)'' as one of the important application of mobile vehicles and wireless technologies, providing the information regarding mobile position, direction, speed and other real time information thus avoiding the traffic jams and accidents. Further, proposed a decentralized authentication mechanism using consensus algorithm of blockchain for IoV. They used blockchain to design new key distribution mechanism for joining the new node's information in the blockchain ledger.

Lin *et al.* [30] focused on the limitation of the IoT system due to resource constrained IoT devices and proposed an ''outsourcing of bilinear pairings (SOBP)'' integrated with permissioned blockchain to address the shortcoming. Thus helps in overcoming the limitations of IoT. They utilized the potentials of permission blockchain (i.e., enhancing security, service availability and system scalability). Finally authors proved and implemented the proposed system to evaluate its security, performance and feasibility.

Chaudhary *et al.* [31] focused on the emerging and recent technology ''Tactile Internet'' in the field of intelligent transportation system and identified the need to create a secure energy trading ecosystem. Further, proposed a blockchain based secure energy trading scheme in electric vehicle (EVs) for validating EVs request, selecting minor nodes to validate the request, etc. In order to make it computationally effective i.e., low latency and provide real time services, SDN was utilized as an under lying architecture.

Feng *et al.* [32] discussed the VANET environment i.e., vehicles were connected through wireless communication medium. Some of the potential benefits were like intelligent routing, weather monitoring emergency call etc. They also identified the importance of accuracy and reliability of the communicated message. Further, proposed a framework named as ''blockchain assisted privacy pressuring authentication system (BPAS)'' that preserved the vehicle privacy and provides automatic authentication in VANET. Blockchain automatically checked the message credibility, monitor vehicle behavior and also traced immutable communication record. Further security analysis was conducted to check the proposed framework for security and privacy of VANET.

Jindal *et al.* [33] proposed a framework to provide energy trading between electric vehicles (EVs) and charging stations (CSs). To secure the energy trading transactions in software defined networking (SDN) aided vehicle to grid environment, consensus based blockchain was used. It reduced the latency and increased the throughout and thus make the proposed system effective and efficient.

Mingxiao *et al.* [34] surveyed the effectiveness of blockchain because of its decentralization, stability, security and non-modifiability properties. It identified that consensus algorithm played an important role in the success of blockchain. Further, they reviewed the principles, characteristics, performance analysis and application scenarios of different consensus algorithm.

Zhang *et al.* [35] highlighted the importance of reaching a consensus in group decision making process (GDM) with minimum cost and maximum return. Further, proposed a generalized ''minimum cost soft consensus models'' and ''maximum return model'' under a certain degree of consensus. The correlation between both the model and their economic significance was also presented. In order to prove its utility the proposed models were applied in loan consensus problem in P2P lending utilizing data from Chinese P2P platform and demonstrated how it helped borrowers and lenders to attain a certain level of consensus about interest rate. It efficiently coordinated the supply and demand in P2P lending.

Jang *et al.* [36] designed a ''hybrid security scheme that uses both heterogeneous cryptosystems, such as symmetric and asymmetric (public) key cryptographic techniques''.

**TABLE 1. Summary of various cryptographic techniques used and limitations of existing authentication schemes.**

| Scheme | Cryptographic Techniques | Drawbacks/Limitations |
|---|---|---|
| Jang *et al.* [36] | * Uses a hybrid security scheme that applies both heterogeneous cryptosystems (symmetric and public key cryptosystems)<br>* Uses "Elliptic Curve Cryptography (ECC)" primitives and one-way cryptographic hash function | * Does not preserve anonymity property<br>* Does not support dynamic controller node addition<br>* Does not support dynamic medical device addition<br>* Does not support blockchain solution |
| He and Zeadally [37] | * Based on authentication using Ambient Assisted Living (AAL) system<br>* Applies wearable sensors in WBANs and assistive robotics<br>* Uses ECC primitives, symmetric encryption/decryption and one-way cryptographic hash function | * Does not support dynamic controller node addition<br>* Does not support dynamic medical device addition<br>* Does not support blockchain solution |
| Merabet *et al.* [38] | * Based on M2C authentication protocol<br>* Based on M2M authentication protocol<br>* Uses ECC primitives and one-way cryptographic hash function | * Does not support dynamic medical device addition<br>* Does not support dynamic medical device addition<br>* Does not provide session key security<br>* Insecure against stolen mobile device/programmer attack<br>* Does not support blockchain solution |

Unfortunately, their scheme fails to maintain "anonymity property" and also it does not support "dynamic controller node addition and medical device addition".

He and Zeadally [37] designed another authentication protocol by using the "ambient intelligence, specifically for an Ambient Assisted Living (AAL) system". Their protocol helps in monitoring health-related information and also in providing "tele-health care services". Their system applies the "wearable sensors in the wireless body area networks (WBANs) and assistive robotics".

Merabet *et al.* [38] discussed "Machine-to-Machine (M2M)" and "Machine-to-Cloud (M2C)" modes needed in the IoT-based healthcare applications. Next, they proposed two protocols: 1) Protocol-I: "M2C authentication protocol" and b) Protocol-II: "M2M authentication protocol". Unfortunately, their protocols do not support "dynamic controller node addition and medical device addition".

Chase and MacBrough [39] provided a comprehensive description and carried detailed analysis of XRP ledger consensus protocol (low latency Byzantine agreement protocol to reach a consensus without universal agreement of network participants. Along with providing the detailed description of the XRP ledger consensus protocol, they validated the networks conditions required to guarantee correctness of the algorithm.

In order to achieve both anonymity and regulation properties, Lin *et al.* [40] designed a "conditional anonymous payment scheme" and applied the designed scheme to construct their first "decentralized conditional anonymous payment system". As compared with the Zerocash, they have shown that their proposed system can be deployed for practical real-world deployment.

Ismail and Materwala [41] provided a detailed study on the evolution of blockchain frameworks and consensus protocols. They identified the need to develop a scalable, cost-efficient and green blockchain frameworks to address the goals for a green collaborative, decentralized and agile ecosystems like smart cities, such as health care and governance. An overview of blockchain technology including blockchain layers i.e., infrastructure layer, platform layer, distributed computing layer and application layer/business logic were also presented. Furthermore, the taxonomy, classification (compute-intensive based, voting based and capability based) and comparison among different consensus protocols were provided.

Finally, in Table 1 we summarize discussed existing authentication schemes with their applied cryptographic techniques and disadvantages/limitations.

## III. SYSTEM MODELS

We follow the following two models in the designing of BAKMP-IoMT.

### A. NETWORK MODEL

BAKMP-IoMT uses network model as shown in Figure 1. In this figure, we have a patient who is implanted with medical devices such as neurostimulator, cochlear implant, cardiac pacemaker and gastric stimulator. There is personal server node (sometimes called as controller node) nearby to the patient which collects the data from the IMDs through some wireless communication technology such as bluetooth. Personal server then transmits the collected data to the cloud server through the access point. Cloud servers receive and store the data which can be used for the further processing. There are some users who are interested in accessing the data of the patient (i.e., doctor, nurse and patient's relative). Various network entities i.e., IMDs, personal server, cloud server and users, in the system is registered by the trusted authority. Patient's data can be accessed by the user from the cloud server after performing the certain steps of the user authentication process. In this model cloud servers are the resource rich devices which have high processing, storage and communication capacity. Cloud server acts the miner node in this blockchain based Internet of Medical of Things communication environment. After receiving the data from the personal server, cloud server (miner node) prepares a block and add this in the blockchain when it is successfully committed by the other miners. This kind of communication environment is susceptible to different attacks such as "replay", "man-in-the-middle","impersonation", "password guessing", "session key computation" through physical capturing of devices, "health data disclosure" and "denial of service". Communication between IMDs to personal server, personal server to cloud server and cloud server to user, must be secured. Hence, session key must be established between the user and the cloud server, and also between the cloud

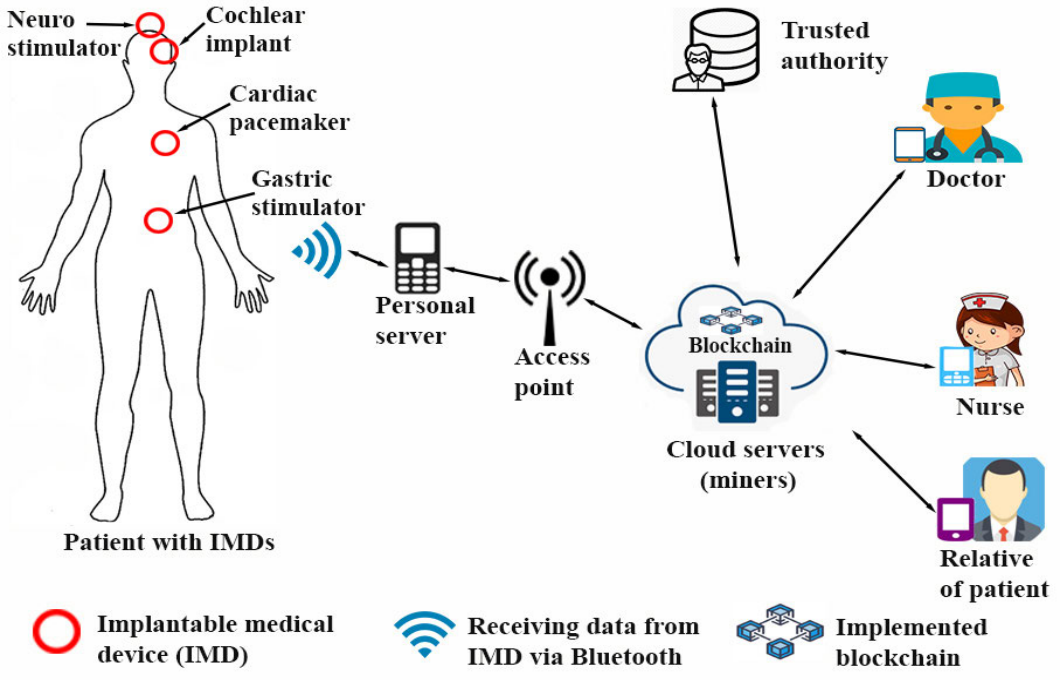


**FIGURE 1.** Network model of blockchain based IoMT communication environment.

server and personal server. Since majority of implantable medical devices have limited resources so we select to utilize lightweight cryptographic operations (i.e., hash, XOR operations) in the different exchanged messages. The addition of blockchain operations at the cloud servers give more resilience to this scheme against the possible attacks. Therefore, we need a secure data access scheme in the blockchain based IoMT communication environment. Using this scheme the communicating entity (i.e., doctor) can securely access the data of a patient without any disclosure. Moreover, in our proposed scheme (BAKMP-IoMT), we consider only a private blockchain because the healthcare data is fully confidential and private. As a result, if these data is somehow put on a public or even on a hybrid blockchain, privacy issues related to healthcare data will arise.

### B. THREAT MODEL

BAKMP-IoMT is designed using the guidelines of widely-used (DY) threat model [42]. According to DY model any two communicating parties communicate over an open insecure channel and also, the end-point entities such as IMDs, personal server and users are not in general trustworthy. An adversary ($\mathcal{A}$) can read (eavesdrop) the communicated messages and can also modify or delete the transmitted messages over insecure channel. We also follow Canetti and Krawczyk's adversary model, known as the CK-adversary model [43], [44] which is current *de facto* standard model in the modeling of an "authenticated key-agreement security protocol". According to the assumptions of CK-adversary model, $\mathcal{A}$ can have all the potentialities as in the DY model along with that he/she can compromise the secret credentials and with the session states (session keys) in the sessions. Apart from that $\mathcal{A}$ can physical capture some IMDs, personal servers or the mobile device of the users and extracts the stored information from these devices by utilizing the sophisticated power analysis attack [45]. This derived information can be used to perform other malicious activities for example acquiring of secret credentials and session key computations, "device impersonation attack", "replay attack", "privileged-insider attack" and "man-in-the-middle attack". At long Last, the trusted authority (*TA*) is thought to be full trusted entity in the network and it won't be compromised. Moreover, cloud servers acts as the miners in the network. Hence, they are also considered as the trusted entities.

## IV. THE PROPOSED SCHEME

In the designing of the proposed blockchain enabaled authentication key agreement protocol for IoMT environment (BAKMP-IoMT), we use following phases. The different notations used in explaining and analyzing BAKMP-IoMT are given in Table 2. BAKMP-IoMT includes following eight stages: 1) pre-deployment, 2) key management, 3) user registration, 4) login, 5) authentication & key agreement, 6) Blockchain construction and addition, 7) password & biometric update and 8) dynamic IMD addition.

In order to improve the security of BAKMP-IoMT, following three factors are used for authentication purpose: 1) mobile device $MD_{U_i}$ of a user $U_i$ which stores significant credentials required for authentication, 2) password of $U_i$, and 3) personal biometric of $U_i$. To safeguard against replay

**TABLE 2. Notations utilized in BAKMP-IoMT.**

| Symbol | Explanation |
|---|---|
| $\mathcal{A}$ | An adversary |
| $U_i$, $MD_i$ | $i^{th}$ user and his/her mobile device, respectively |
| $ID_{U_i}$, $RID_{U_i}$, $PW_{U_i}$, $BIO_{U_i}$ | $U_i$'s identity, his/her Pseudo identity, password and biometric, respectively |
| $TA$, $ID_{TA}$ | Trusted authority and its identity |
| $IMD_k$, $ID_{IMD_k}$ | $k^{th}$ implantable medical device and its identity |
| $PS_l$, $ID_{PS_l}$ | $l^{th}$ Personal server and its identity |
| $CS_j$, $ID_{CS_j}$ | $j^{th}$ Cloud server (miner) and its identity |
| $N$ | 160-bit secret key of $TA$ |
| $RID_{IMD_k}$, $RID_{PS_l}$ | Pseudo identities of $IMD_k$ and $PS_l$, respectively |
| $RID_{CS_j}$, $RID_{TA}$ | Pseudo identities of $CS_j$ and $TA$, respectively |
| $B_{IMD_k}$ | Block in a blockchain for $k^{th}$ implantable medical device data |
| $h(d_{IMD_k})$ | Hash value of $B_{IMD_k}$ |
| $d_{IMD_k}$ | Data send by $IMD_k$ |
| $KP_{IMD_k}$ | Private key for $IMD_k$ |
| $KU_{IMD_k}$ | Public key for $IMD_k$ |
| $E_{KP_{IMD_k}(h(d_{IMD_k}))}$ | Encryption using $KP_{IMD_k}$ on $h(d_{IMD_k})$ |
| $C_{IMD_k}[KU_{IMD_k}]$ | Certificate contains public key for the encrypted data of $IMD_k$ |
| $h(d_{IMD_{k-1}})$ | Hash value of previous block for $IMD_{k-1}$ |
| $n_{IMD}$ | Number of IMDs to be deployed |
| $T_1, T_2, T_3$ | Current timestamps |
| $\Delta T$ | Maximum transmission delay |
| $Gen(\cdot)$ | Generation process in fuzzy extractor |
| $Rep(\cdot)$ | Reproduction process in fuzzy extractor |
| $\sigma_{U_i}$ | Biometric secret key of $U_h$ for $BIO_{U_h}$ |
| $\tau_{U_i}$ | Public reproduction parameter of $U_h$ for $BIO_{U_h}$ |
| $t$ | Error tolerance threshold required by fuzzy extractor |
| $h(\cdot)$ | Cryptographic one-way hash function |
| $SK_{X-Y} (= SK_{Y-X})$ | Session key between entity $X$ and entity $Y$ |
| $\|\|, \oplus$ | Concatenation & bitwise $XOR$ operations, respectively |
| $ECDSA$ | Elliptic curve digital signature algorithm for signature generation and verification on a block |
| $x$ | 160-bit secret number of $U_i$ |
| $E_q(a, b)$ | A non-singular elliptic curve of the form: "$y^2 = x^3 + ax + b \pmod q$ with $4a^3 + 27b^2 \neq 0 \pmod q$" |
| $G$ | A base point in $E_q(a, b)$ whose order is $n_o$ as big as $q$ |
| $k.G$ | Elliptic curve point multiplication: $k.G = G + G + \cdots + G\,(k\,times)$ |
| $r_{CS_j}$ | Private key of $CS_j$ |
| $Pub_{CS_j}$ | Public key of $CS_j$, where $Pub_{CS_j} = r_{CS_j} \cdot G$ |

attack, different random nonces (numbers) and current timestamps are utilized. Moreover, all the network entities participating in BoMT communication environment are assumed to be synchronized with their clocks. This is a fair assumption, which is included in the designing of several newly proposed authentication protocols [46]–[49], [50]. We utilize cryptographic one-way hash function and bitwise *XOR* operations to form BAKMP-IoMT lightweight as IMDs are resource constraint in nature. For biometric verification, we utilize the fuzzy extractor at the user side only. In order to provide better security, expensive computations are carried on resource-rich devices i.e., user's mobile device and cloud server, in the network [51].

It is worth noticing that the trusted authority (*TA*) only involves during the registration phase, which is one-time process. The *TA* does not have any other role in our proposed system. Moreover, the *TA* does not take any participation in active role during the communication (for example, key management phase, user login and authentication & key agreement phases, and blockchain construction and addition phase). In addition, even the *TA* does not know what kind of information the entities exchange and what are the values of established session keys among various network entities. Hence, it will be definitely a risky task to involve the cloud servers for the registration of different network entities, and in that situation, some active attacks, such as "privileged insider attack", "illegal credential leakage attack" and "unauthorised session key computation attack" may be possible. As a result, we utilize only the trusted authority (*TA*) for the purpose of registration of various network entities instead of the cloud servers.

### A. PRE-DEPLOYMENT

In this phase the registration of various network entities is done by *TA*. The pre-deployment phase is explained below.

#### 1) IMD REGISTRATION

The *TA* selects a unique identity $ID_{IMD_k}$ for implantable medical device $IMD_k$ and computes corresponding pseudo-identity $RID_{IMD_k} = h(ID_{IMD_k}||N)$ where the *TA*'s secret key is $N$. Thereafter, the *TA* choose a exclusive "symmetric bivariate polynomial" $\chi(x, y) = \sum_{m,n=0}^{t} a_{m,n} x^m y^n \in GF(p)[x, y]$ of degree $t$ over a finite field (Galois field) $GF(p)$ $(= Z_p)$, where the co-efficients $a_{i,j}$'s are selected from $GF(p)$ and $Z_p = \{0, 1, 2, \cdots, p-1\}$ with $p$ being an adequately large prime and $t$ is sufficiently greater than the total count of IMDs to be deployed. For instance, a bivariate polynomial $\chi(x, y) = x^4 + 3x^3 + 2x^2y^2 + 3y^3 + y^4$ over $GF(5)$ is symmetric as $\chi(y, x) = y^4 + 3y^3 + 2y^2x^2 + 3x^3 + x^4 = \chi(x, y)$. Furthermore, *TA* computes a polynomial share $\chi(RID_{IMD_k}, y) = \sum_{m,n=0}^{t} [a_{m,n}(RID_{IMD_k})^m] y^n$, which is certainly a univariate polynomial of the same degree $t$. Then *TA* stores the credentials $\{RID_{IMD_k}, \chi(RID_{IMD_k}, y)\}$ in $IMD_k$'s memory before their implementation. Note that for safeguarding unconditional security and $t$-collusion resistant property against IMD physical capture attack by an adversary $\mathcal{A}$ [52], [53], it is necessary to achieve the property $t \gg n_{IMD}$. This is because if greater than $t$ IMDs are not trapped by $\mathcal{A}$, the original polynomial $\chi(x, y)$ will not be computed by the compromised polynomial shared from these IMDs's memory respectively.

#### 2) PERSONAL SERVER REGISTRATION

The *TA* first chooses his/her own identity $ID_{TA}$ and computes corresponding pseudo-identity $RID_{TA} = h(ID_{TA}||N)$. *TA* further selects a unique identity $ID_{PS_l}$ for personal server $PS_l$ and computes corresponding pseudo-identity $RID_{PS_l} = h(ID_{PS_l}||N)$ using the *TA*'s secret key $N$ and temporal credential $TC_{PS_l} = h(ID_{PS_l}||ID_{TA}||RTS_{PS_l}||N)$ where the registration timestamp of $PS_l$ is $RTS_{PS_l}$. *TA* then computes the polynomial share $\chi(RID_{PS_l}, y) = \sum_{m,n=0}^{t} [a_{m,n}(RID_{PS_l})^m] y^n$ for each $PS_l$, where $\chi(x, y) = \sum_{m,n=0}^{t} a_{m,n} x^m y^n \in GF(p)[x, y]$ is the same symmetric bivariate polynomial of degree $t$ that is previously chosen in Section IV-A1. Then *TA* stores the information $\{RID_{PS_l}, TC_{PS_l}, RID_{TA}, \chi(RID_{PS_l}, y))\}$ in $PS_l$'s database before its deployment.

#### 3) CLOUD SERVER (MINERS) REGISTRATION

The *TA* first chooses a unique identity $ID_{CS_j}$ for cloud server $CS_j$ and computes corresponding pseudo-identity $RID_{CS_j} = h(ID_{CS_j}||N)$ using the *TA*'s secret key $N$. After that *TA* stores the information $\{RID_{CS_j}, RID_{TA}\}$ in $CS_j$'s database before its deployment.

### B. KEY MANAGEMENT

Key management phase is required to secure the communication between $IMD_k$ to $PS_l$ and $PS_l$ to $CS_j$. After the successful completion of all steps of key management phase $IMD_k$ to $PS_l$ and $PS_l$ to $CS_j$ establish secret pairwise keys for their secure communication.

#### 1) KEY MANAGEMENT BETWEEN IMPLANTABLE MEDICAL DEVICE AND PERSONAL SERVER

To establish secret pairwise key between an implantable medical device $IMD_k$ and a personal server $PS_l$ following steps are used:

- $IMD_k$ first generates current timestamp $TIP_1$ and sends the message $\{RID_{IMD_k}, TIP_1\}$ to $PS_l$ via open channel.
- After receiving $\{RID_{IMD_k}, TIP_1\}$, $PS_l$ examines the timeliness of $TIP_1$ by the expression $|TIP_1 - TIP_1^*| \leq \Delta T$, where $TIP_1^*$ denotes the receiving timestamp of the message. If it is valid, $PS_l$ again produce current timestamp $TIP_2$ calculates $\omega = \chi(RID_{PS_l}, RID_{IMD_k})$, the secret key $SK_{PS_l,IMD_k} = h(\omega||TIP_1)$ and $MS_1 = h(SK_{PS_l,IMD_k}||TIP_2)$. $PS_l$ then transmits the message $\{RID_{PS_l}, MS_1, TIP_2\}$ to $IMD_k$ through public channel.
- After receiving $\{RID_{PS_l}, MS_1, TIP_2\}$ from $PS_l$, $IMD_k$ substantiate the timeliness of $TIP_2$ by investigating if $|TIP_2 - TIP_2^*| \leq \Delta T$ is valid, where $TIP_2^*$ is the receiving time of the message. If $TIP_2$ is successfully verified, $IMD_k$ computes $\omega' = \chi(RID_{IMD_k}, RID_{PS_l})$, the secret key $SK_{IMD_k,PS_l} = h(\omega'||TIP_1)$ and $MS_1' = h(SK_{IMD_k,PS_l}||TIP_2)$. Then $IMD_k$ checks whether $MS_1' = MS_1$. If the condition is fulfilled, the calculated secret pairwise key is correct.

Later on, both $IMD_k$ and $PS_l$ store $SK_{IMD_k,PS_l}$ $(= SK_{PS_l,IMD_k})$ for their future secure communication. This phase is summarized in Figure 2.

#### 2) KEY MANAGEMENT BETWEEN PERSONAL SERVER AND CLOUD SERVER

To establish secret pairwise key between a personal server $PS_l$ and a cloud server $CS_j$ following steps are performed:

- $PS_l$ first produces a random nonce $r_1$ and a current timestamp $TPC_1$ and calculates $M_1 = h(r_1||TC_{PS_l}) \oplus RID_{TA}$ and $M_2 = h(h(r_1||TC_{PS_l})||RID_{TA}||TPC_1)$. $PS_l$ then sends message $\{M_1, M_2, TPC_1\}$ to $CS_j$ via open channel.
- After receiving $\{M_1, M_2, TPC_1\}$, $CS_j$ examines the timeliness of $TPC_1$ by the expression $|TPC_1 - TPC_1^*| \leq \Delta T$, where $TPC_1^*$ is the receiving timestamp of the message. If it is valid, $CS_j$ computes $h(r_1||TC_{PS_l}) = M_1 \oplus RID_{TA}$ and $M_2' = h(h(r_1||TC_{PS_l})||RID_{TA}||TPC_1)$. Then $CS_j$ checks whether $M_2' = M_2$, if so $PS_l$ is authenticated with $CS_j$; Otherwise $CS_j$ immediately terminates the session with $PS_l$. Further, $CS_j$ produces a random nonce $r_2$ and a current timestamp $TPC_2$, and computes $M_3 = h(r_2||RID_{CS_j}) \oplus RID_{TA}$. After that $CS_j$ computes secret key $SK_{CS_j,PS_l} = h(h(r_2||RID_{CS_j})||RID_{TA}||h(r_1||TC_{PS_l})||TPC_1||TPC_2)$ and $M_4 = h(SK_{CS_j,PS_l}||TPC_2)$ then transmits the message $\{M_3, M_4, TPC_2\}$ to $PS_l$ through public channel.
- After receiving $\{M_3, M_4, TPC_2\}$ from $CS_j$, $PS_l$ substantiates the timeliness of $TPC_2$ by examining if $|TPC_2 - TPC_2^*| \leq \Delta T$ is valid, where $TPC_2^*$ is the receiving time of the message. If $TPC_2$ is successfully verified, $PS_l$ computes $h(r_2||RID_{CS_j}) = M_3 \oplus RID_{TA}$, secret key $SK_{PS_l,CS_j} = h(h(r_2||RID_{CS_j})||RID_{TA}||h(r_1||TC_{PS_l})||TPC_1||TPC_2)$ and $M_4' = h(SK_{PS_l,CS_j}||TPC_2)$. Then $PS_l$ checks whether $M_4' = M_4$, if so $CS_j$ is authenticated with $PS_l$ and computed session key is correct.
- Further $PS_l$ generates a current timestamp $TPC_3$ and computes $M_5 = h(SK_{PS_l,CS_j}||TPC_3)$ sends message $\{M_5, TPC_3\}$ to $CS_j$ through open channel. After receiving $\{M_5, TPC_3\}$ from $PS_l$, $CS_j$ verifies the timeliness of $TPC_3$ by examining if $|TPC_3 - TPC_3^*| \leq \Delta T$ is valid, where $TPC_3^*$ is the receiving time of the message. If it holds $CS_j$ computes $M_5' = h(SK_{CS_j,PS_l}||TPC_3)$ and check whether $M_5' = M_5$, if so $CS_j$ confirms that $PS_l$ computed the correct secret key.

At the end, both $PS_l$ and $CS_j$ store $SK_{PS_l,CS_j}$ $(= SK_{CS_j,PS_l})$ for their future secure communication. This phase is summarized in Figure 3.

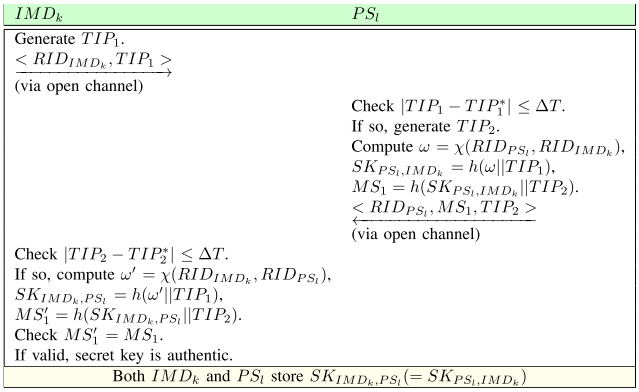


**FIGURE 2.** Summary of key management between implantable medical device and personal server.

### C. USER REGISTRATION

For accessing the data of $IMD_k$ in a "blockchain based Internet of Medical Things" communication, user (i.e., doctor) $U_i$ is registered using User registration method. For this purpose, $U_i$ requires secure registration at the trusted authority *TA* either in person or through a secure channel using the below mentioned steps:

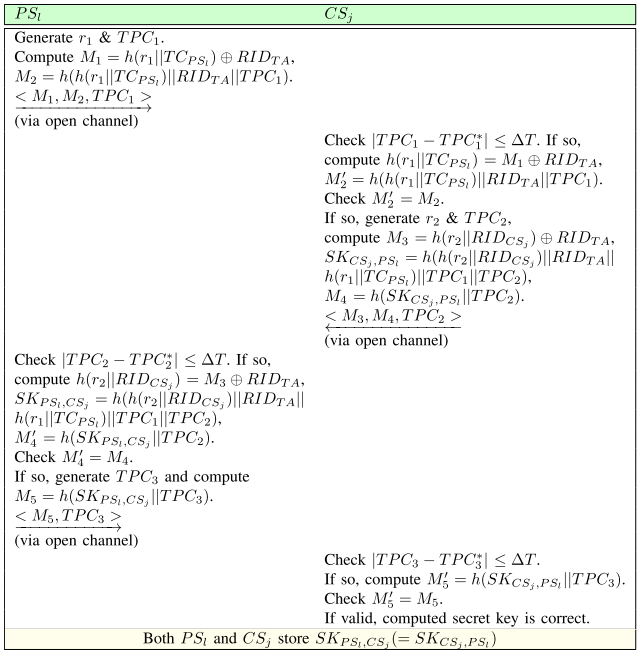


**FIGURE 3.** Summary of key management between personal server and cloud server.

- **Step REG1.** $U_i/MD_{U_i}$ chooses his/her identity $ID_{U_i}$ and sends the request for registration $\langle ID_{U_i} \rangle$ to $TA$ securely.
- **Step REG2.** After receiving registration request, $TA$ computes $U_i$'s pseudo identity as $RID_{U_i} = h(ID_{U_i}||N)$. $TA$ also computes temporal credential of $U_i$ as $\alpha = h(RID_{U_i}||\ RID_{TA}\ ||RTS_{U_i})$, where $RTS_{U_i}$ is the registration timestamp generated for $U_i$ by $TA$. $TA$ then prepares a registration reply for $U_i$ as $\langle RID_{U_i}, \alpha, RID_{TA}, h(\cdot)\rangle$ and send that to $U_i$ through a secure channel.
- **Step REG3.** After receiving registration reply $\langle RID_{U_i}, \alpha, RID_{TA}, h(\cdot)\rangle$ from $TA$, $U_i$ selects a password $PW_{U_i}$ according to his/her liking and also furnish biometric data $BIO_{U_i}$ to the terminal of a specific device. $MD_{U_i}$ then determines $(\sigma_{U_i}, \tau_{U_i}) = Gen(BIO_{U_i})$, where $\sigma_{U_i}$ and $\tau_{U_i}$ are the biometric secret key of $l$ bits and public reproduction parameter.
- **Step REG4.** $U_i$ selects 160-bit secret number $x$ and computes pseudo-random password $RPW_{U_i} = h(PW_{U_i}||x)$, other parameters such as $\beta = x \oplus h(ID_{U_i}||\ PW_{U_i}||\sigma_{U_i})$, $RID'_{U_i} = RID_{U_i} \oplus h(PW_{U_i}||\ \sigma_{U_i})$, $RID'_{TA} = RID_{TA} \oplus h(ID_{U_i}||x||\ \sigma_{U_i})$, $\alpha' = \alpha \oplus h(x||PW_{U_i}||\ \sigma_{U_i})$, $\delta = h(\alpha||RPW_{U_i}||\sigma_{U_i})$ and $EQ = h(ID_{U_i}||\ RID_{TA}||\delta)$. Finally, $MD_{U_i}$ of $U_i$ stores the information $\{RID'_{U_i}, RID'_{TA}, \alpha', EQ, \beta, \tau_{U_i}, h(\cdot), Gen(\cdot), Rep(\cdot), t\}$ in its memory. $U_i$ discards the information $\langle RID_{U_i}, RID_{TA}, \alpha, x\rangle$ from $MD_{U_i}$'s memory as a means to safeguard the lost/stolen smart card/mobile device attack by an adversary.
- **Step REG5.** For the secure communication between $TA$ and $CS_j$, we assume a master symmetric key $MK_{TA-CS_j}$. After $U_i$'s successful registration, $TA$ transmits the information $\{RID_{U_i}\}$ encrypted utilizing the symmetric key $MK_{TA-CS_j}$, and then the $CS_j$ decrypts the received information utilizing the same symmetric key $MK_{TA-CS_j}$ to store it in its database. Finally, the $CS_j$ accumulates information $\{RID_{CS_j}, RID_{TA}, RID_{U_i}\}$ in its database.

This phase is abridged in Figure 4.

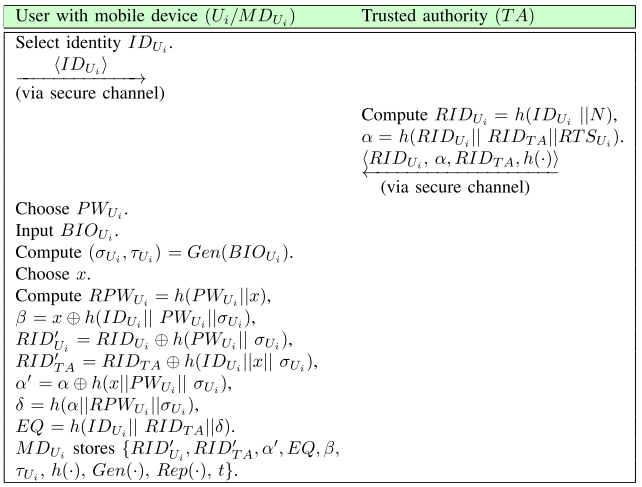


**FIGURE 4.** Summary of user registration phase.

### D. LOGIN PHASE

In order to login into the system, $U_i$ is required to execute the below mentioned steps:

- **Step LOG1.** $U_i$ furnishes his/her identity $ID_{U_i}$ and password $PW'_{U_i}$ and also marks biometrics information $BIO'_{U_i}$ at the sensor of the specified device. $MD_{U_i}$ computes biometric secret key $\sigma'_{U_i} = Rep(BIO'_{U_i}, \tau'_{U_i})$ with the constraint that the hamming distance between the actual biometrics $BIO_{U_i}$ furnished at the time of user registration process and currently provided $BIO'_{U_i}$ is less than or equal to the error tolerance threshold value $t$.
- **Step LOG2.** $MD_{U_i}$ next computes other parameters such as $x = \beta \oplus h(ID_{U_i}||\ PW'_{U_i}||\sigma'_{U_i})$, $RPW'_{U_i} = h(PW'_{U_i}||x)$, $RID_{U_i} = RID'_{U_i} \oplus h(PW_{U_i}||\ \sigma'_{U_i})$, $RID_{TA} = RID'_{TA} \oplus h(ID_{U_i}||x||\ \sigma'_{U_i})$, $\alpha = \alpha' \oplus h(x||PW_{U_i}||\ \sigma'_{U_i})$, $\delta' = h(\alpha||RPW'_{U_i}||\sigma'_{U_i})$. After the computation of these parameters, $MD_{U_i}$ also calculates $EQ' = h(ID_{U_i}||\ RID_{TA}||\delta')$, and then examines the equality of $EQ' = EQ$. If it results in a match, $MD_{U_i}$ validates the authenticity of $U_i$ locally, else the login phase ends immediately.
- **Step LOG3.** $MD_{U_i}$ generates the current timestamp $TS_1$ besides with 128-bit random nonce $r_{U_i}$. $MD_{U_i}$ then calculates $M_1 = r_{U_i} \oplus h(RID_{TA}||RID_{U_i})$ and $M_2 = h(r_{U_i}||RID_{U_i}||RID_{TA}||TS_1)$. Finally, $MD_{U_i}$ sends the message $Msg_1 = \langle M_1, M_2, TS_1\rangle$ as a login request to the $CS_j$ via insecure channel.

### E. AUTHENTICATION AND KEY AGREEMENT PHASE

After receiving the login request $\langle M_1, M_2, TS_1\rangle$ from $U_i$, the specified steps are executed between a user $U_i$ and cloud

server i.e., $CS_j'$ which provides service to $U_i$. After the successful completion of these steps, mutual authentication is achieved between communicating parties $U_i$ and $CS_j'$ and for their secure communication, session Key is also established.

- **Step AU1.** $CS_j'$ first examines the timeliness of $TS_1$ by the expression $|TS_1 - TS_1^*| \leq \Delta T$, where the maximum transmission delay is denoted by $\Delta T$ and $TS_1^*$ is the reception time of the message $\langle M_1, M_2, TS_1\rangle$. If it matches, $CS_j'$ computes $r_{U_i} = M_1 \oplus h(RID_{TA}||RID_{U_i})$ and $M_2' = h(r_{U_i}||RID_{U_i}||\ RID_{TA}||TS_1)$ by using the stored value $RID_{U_i}$ for user $U_i$. Further $CS_j'$ verifies the equality of the equation $M_2' = M_2$. If it holds, $U_i$ is authenticated by $CS_j'$. Else, $CS_j'$ terminates the session with $U_i$ immediately.
- **Step AU2.** Then $CS_j'$ generates the current timestamp $TS_2$ along with 128-bit random nonce $r_{CS_j'}$ and computes $M_3 = h(r_{CS_j'}||RID_{CS_j'}) \oplus h(RID_{U_i}||RID_{TA})$. Further $CS_j'$ computes the session key $SK_{CS_j',U_i} = h(r_{U_i}||RID_{U_i}||TS_1||TS_2||h(r_{CS_j'}||RID_{CS_j'})||RID_{TA})$ and $M_4 = h(SK_{CS_j',U_i}||TS_2)$. After that $CS_j'$ sends authentication reply $Msg_2 = \langle M_3, M_4, TS_2\rangle$ to $U_i$ through open channel.
- **Step AU3.** After receiving authentication reply $Msg_2 = \langle M_3, M_4, TS_2\rangle$ from $CS_j'$, $U_i$ first examines the timeliness of $TS_2$ by the expression $|TS_2 - TS_2^*| \leq \Delta T$, where the maximum transmission delay is indicated by $\Delta T$ and $TS_2^*$ is the reception time of the message $\langle M_3, M_4, TS_2\rangle$. If it matches, $U_i$ computes $h(r_{CS_j'}||RID_{CS_j'}) = M_3 \oplus h(RID_{U_i}||RID_{TA})$, session key $SK_{U_i,CS_j'} = h(r_{U_i}||RID_{U_i}||TS_1||TS_2||h(r_{CS_j'}||RID_{CS_j'})||RID_{TA})$ and $M_4' = h(SK_{U_i,CS_j'}||TS_2)$. Further $U_i$ verifies the equality of the equation $M_4' = M_4$. If it holds, $CS_j'$ is authenticated by $U_i$ and calculated session key is correct. Else, $U_i$ terminates the session with $CS_j'$ immediately.
- **Step AU4.** Then $U_i$ generates the current timestamp $TS_3$ and computes $M_5 = h(SK_{U_i,CS_j'}||TS_3)$, and then sends message $Msg_3 = \langle M_5, TS_3\rangle$ to $CS_j'$ via open channel for cross verification of computed session key. After receiving authentication reply $Msg_3 = \langle M_5, TS_3\rangle$ from $U_i$, $CS_j'$ first examines the timeliness of $TS_3$ by the condition $|TS_3 - TS_3^*| \leq \Delta T$, where the maximum transmission delay is denoted by $\Delta T$ and $TS_3^*$ is the reception time of the message $\langle M_5, TS_3\rangle$. If it matches, $U_i$ computes $M_5' = h(SK_{CS_j',U_i}||TS_3)$ and examines $M_5' = M_5$. If it holds, $CS_j'$ confirms that computed session key by $U_i$ is correct.

For their secure communication in future, both $U_i$ and $CS_j'$ store $SK_{U_i,CS_j'}$ $(= SK_{CS_j',U_i})$ at the end. Login and authentication & key agreement phases are summarized in Figure 5.

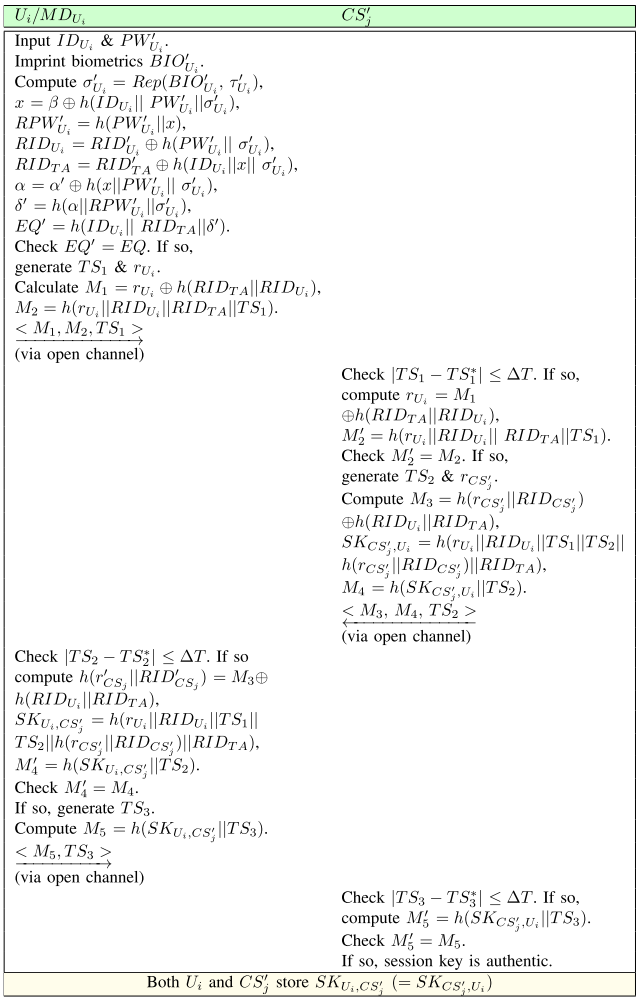


**FIGURE 5.** Summary of login and authentication phases.

### F. BLOCKCHAIN CONSTRUCTION AND ADDITION PHASE

To serve this purpose, an implantable medical device $IMD_k$, personal server $PS_l$, cloud server $CS_j$ and user $U_i$ can perform following steps:

- **Step BDA1.** When $IMD_k$ has to send some data to $PS_l$, it first generates a current timestamp $TP_1$ and prepares a message by applying the encryption using the computed and established pairwise secret key $SK_{IMD_k,PS_l}$, say $msg_1 = E_{SK_{IMD_k,PS_l}}(d_{IMD_k},\ TP_1)$ and sends this to $PS_l$ via open channel. After receiving $msg_1 = E_{SK_{IMD_k,PS_l}}(d_{IMD_k},\ TP_1)$ from $IMD_k$, $PS_l$ decrypts the message $E_{SK_{IMD_k,PS_l}}(d_{IMD_k},\ TP_1)$ to get the values of $d_{IMD_k}$ and $TP_1$. $PS_l$ then examines the timeliness of $TP_1$ by the expression $|TP_1 - TP_1^*| \leq \Delta T$, where $TP_1^*$ is the receiving timestamp of the message. If it holds, $PS_l$ generates a current timestamp $TP_2$ and prepares a message by applying the encryption using the computed and established pairwise secret key $SK_{PS_l,CS_j}$ as $msg_2 = E_{SK_{PS_l,CS_j}}(d_{IMD_k},\ TP_2)$ and sends this to $CS_j$ via open channel.
- **Step BDA2.** In the proposed model, the cloud servers act as the miner nodes. Each cloud server receives the data of the patient from the personal server through the established secret key securely between the cloud server and the personal server. After receiving

$msg_2 = E_{SK_{PS_l,CS_j}}(d_{IMD_k}, TP_2)$, $CS_j$ decrypts the message $E_{SK_{PS_l,CS_j}}(d_{IMD_k}, TP_2)$ to get the values of $d_{IMD_k}$ and $TP_2$. Next, $CS_j$ examines the timeliness of $TP_2$ by the expression $|TP_2 - TP_2^*| \leq \Delta T$, where $TP_2^*$ is the receiving timestamp of the message. If it holds, the message is treated as a valid one.

- **Step BDA3.** Using data $d_{IMD_k}$, $CS_j$ starts the procedure of block creation and its addition into the blockchain. The procedure is explained as follows.
  Here, the details of generating a block by the $CS_j$ and verification of the created block by the P2P cloud server (CS) network is provided. The block is added in the blockchain when the consensus is successfully committed among all the cloud servers in P2P CS network. The structure of block is depicted in Figure 6. After successfully collecting the data $d_{IMD_k}$, $CS_j$ builds the transactions of the collected data. Let $Tx_1$, $Tx_2$, $\cdots$, $Tx_{n_t}$ be the transactions generated during a session by the $CS_j$. The creation of a block, say $BLK_i$, contains the transactions $Tx_1, Tx_2, \cdots, Tx_{n_t}$ as provided in Figure 6. The concept of private blockchain is used in the BAKMP-IoMT for the better secrecy and confidentiality of the information. The block $BLK_i$ can be formed as follows.
  - A public key encryption $E(\cdot)$ is used to encrypt all the transactions utilizing public key $Pub_{CS_j}$ (for BAKMP-IoMT, we employ the "ECC encryption").
  - Next, the encrypted transactions $E_{Pub_{CS_j}}(Tx_i)$, $(i = 1, 2, \cdots, n_t)$, are utilized to compute the "Merkle tree root" ($MR$) by building the " Merkle tree".
  - The hash of the information $\{BVer, PBHash, MR, TR, OB, Pub_{GSS}$, block payload (encrypted transactions)$\}$ is computed for the current hash block ($CBHash$).
  - Further, the ECDSA signature, say $BSign = (r_m, s_m)$ on the message $msg = CBHash$ is computed using the ECDSA signature generation algorithm [54] on the current hash block $CBHash$.

| **Block Header** | |
|---|---|
| Block Version | $BVer$ |
| Previous Block Hash | $PBHash$ |
| Merkle Tree Root | $MR$ |
| Timestamp | $TR$ |
| Owner of Block | $OB$ |
| Public Key of Owner | $Pub_{CS_j}$ |
| **Block Payload (Encrypted Transactions)** | |
| Encrypted Transaction #1 | $E_{Pub_{CS_j}}(Tx_1)$ |
| Encrypted Transaction #2 | $E_{Pub_{CS_j}}(Tx_2)$ |
| ⋮ | ⋮ |
| Encrypted Transaction $\#n_t$ | $E_{Pub_{CS_j}}(Tx_{n_t})$ |
| Current Block Hash | $CBHash$ |
| Signature on Block using ECDSA | $BSign$ |

**FIGURE 6.** Formation of a block $BLK_i$ on the transactions by $CS_j$.

When the block $BLK_i$ is built by $CS_j$, this block is forwarded to other miner nodes (cloud servers) in the P2P CS network. For the verification and addition of the constructed block (i.e., $BLK_i$) using the consensus algorithm by P2P CS network, the following steps are needed to be executed.

- When a block $BLK_i$ is received by a cloud server, a miner (cloud server $CS_j$) will be chosen as a leader, say $L$ from the all available cloud servers in the P2P CS network using the existing leader selection process described in [28].
- The "Ripple Protocol Consensus Algorithm (RPCA)" [29] for block verification and addition via voting method is utilized. It is also assumed that each cloud server $CS$ in the P2P CS network has an ECC-based private-public key pair $(r_{CS}, Pub_{CS})$, where $Pub_{CS} = r_{CS} \cdot G$.
- Moreover, the public keys of other cloud servers are accessible to $CS$.
- Algorithms 1 and 2 are used to explain this process, which is similar to the consensus mechanism applied in [1].

- **Step BDA4.** Suppose a user $U_i$ needs to access secret information $d_{IMD_k}$ from $B_{IMD_k}$. For this issue, his/her request goes to the corresponding cloud server (miner node, i.e., $CS_j'$). Furthermore, note that $CS_j$ and $CS_j'$ maintain a common ledger. Therefore, $CS_j'$ has access $B_{IMD_k}$. $CS_j'$ extracts $d_{IMD_k}$ from $B_{IMD_k}$ by applying the operations of blockchain. For example, $CS_j'$ extracts $KU_{IMD_k}$ from certificate part of $B_{IMD_k}$. $CS_j'$ further applies the hash function $h(\cdot)$ on data $d_{IMD_k}$ and gets $h(d_{IMD_k})$. $CS_j'$ then decrypts $E_{KP_{IMD_k}}(h(d'_{IMD_k}))$ and again gets $h(d'_{IMD_k})$. If both hashes are equal, that is, $h(d'_{IMD_k}) = h(d_{IMD_k})$, the block $B_{IMD_k}$ was not modified; otherwise, $CS_j'$ discards the data of $B_{IMD_k}$.
- **Step BDA5.** $U_i$ and $CS_j'$ already established the session key $SK_{CS_j,U_i}$. Therefore, $CS_j'$ generates a current timestamp $TP_3$, and computes a message $msg_3 = E_{SK_{CS_j',U_i}}(d_{IMD_k}, TP_3)$. Then $CS_j'$ sends $msg_3 = E_{SK_{CS_j',U_i}}(d_{IMD_k}, TP_3)$ to $U_i$ via open channel. After receiving $msg_3$, $U_i$ decrypts the message $E_{SK_{CS_j',U_i}}(d_{IMD_k}, TP_3)$ and gets the values of $d_{IMD_k}$ and $TP_3$. $U_i$ checks the timeliness of $TP_3$ by the condition $|TP_3 - TP_3^*| \leq \Delta T$, where $TP_3^*$ is the receiving timestamp of the message. If it holds, the message is treated as valid one.

This phase is summarized in Figure 7.

### G. PASSWORD AND BIOMETRIC UPDATE PHASE

BAKMP-IoMT provides the facility to update password and biometric information by any authorized user. Following steps can be used by a legal user to update his/her password and biometric information regardless of time without involving $TA$. These steps must be rigorously executed to maintain

**Algorithm 1** Consensus Procedure for Block Verification and Addition in Blockchain

**Input:** Given a block $BLK_i = \{BVer, PBHash, MR, TR, OB, Pub_{CS_j}, \{E_{Pub_{CS_j}}(Tx_s) | s = 1, 2, \cdots, n_t\}, CBHash, BSign\}$; private-public keys pairs $(r_{CS_j}, Pub_{CS_j} = r_{CS_j} \cdot G)$ for all other cloud servers in the P2P CS network.

**Output:** Verification and addition of $BLK_i$ in the blockchain.

1: First, a leader ($L$) is selected among the peer nodes in the P2P CS network using the existing leader selection process described in [28]. Assume $L$ has a block $BLK_i = \{BVer, PBHash, MR, TR, OB, Pub_{CS_j}, \{E_{Pub_{CS_j}}(Tx_s) | s = 1, 2, \cdots, n_t\}, CBHash, BSign\}$.
2: $L$ sets $VTCount \leftarrow 0$, where $VTCount$ signifies the vote counter.
$L$ also sets $flag_{CS'_j} = 0, \forall\{j' = 1, 2, \cdots, n_{cs}, L \neq CS'_j\}$, where $n_{cs}$ is the number of cloud servers in the P2P CS network.
3: $L$ creates distinct random nonce $rn_j$ and a current timestamp $TS_j$ for each cloud server.
4: $L$ encrypts $rn_j$ with the public key $Pub_{CS_j}$ of each $CS_j$ as $E_{Pub_{CS_j}}(rn_j, TS_j)$.
5: $L$ sends the message $SMsg_j = \{BLK_i, E_{Pub_{CS_j}}(rn_j, TS_j), TS_j\}$ to all other cloud servers $CS'_j$, $(j' = 1, 2, \cdots, n_{cs}, L \neq CS'_j)$.
6: Assume $SMsg_j$ receives from $L$ by each $CS'_j$ in the P2P CS network at time $TS^*_j$.
7: **for** each cloud server $CS'_j$ **do**
8: **if** $(|TS_j - TS^*_j| < \Delta T)$ **then**
9: Compute the Merkle tree root, $MR^*$ on the encrypted transactions $\{E_{Pub_{CS_j}}(Tx_s) | s = 1, 2, \cdots, n_t\}$.
10: **if** $(MR^* \neq MR)$ **then**
11: Terminate the consensus process.
12: **else**
13: Calculate block hash $CBHash^*$ on the received block $Block_i$ as $CBHash^* = h(BVer || PBHash || MR^* || TR || OB || Pub_{CS_j} || E_{Pub_{CS_j}}(Tx_1) || E_{Pub_{CS_j}}(Tx_2) || \cdots || E_{Pub_{CS_j}}(Tx_{n_t}))$.
14: **if** $(CBHash^* = CBHash)$ **then**
15: Verify the signature $BSign = (r_m, s_m)$ on $BLK_i$ on the message $msg = CBHash^*$ with the help of ECDSA signature verification algorithm.
16: **if** signature is valid **then**
17: Decrypt the encrypted random nonce using pre-computed private key $r_{CS_j}$ as $(rn^*_j, TS''_j) = D_{r_{CS_j}}[E_{Pub_{CS_j}}(rn_j, TS_j)]$ by applying the ECC decryption algorithm.
18: **if** $(TS''_j = TS_j)$ **then**
19: Send the block verification message containing verification status ($VStatus$) and decrypted random nonce as $RMsg_j = \{E_{Pub_L}(rn^*_j, VStatus)\}$ to the leader $L$.
20: **end if**
21: **end if**
22: **end if**
23: **end if**
24: **end if**
25: **end for**

**Algorithm 2** Consensus for Block Verification and Addition in Blockchain (Continued. . .)

26: **for** each received $RMsg_j$ from the responders $CS'_j$ **do**
27: $L$ computes $(rn'_j, VStatus) = D_{r_L}[E_{Pub_L}(rn^*_j, VStatus)]$.
28: **if** $((rn'_j = rn_j)$ and $(VStatus = valid)$ and $(flag_{CS'_j} = 0))$ **then**
29: $L$ sets $VTCount = VTCount + 1$ and $flag_{CS'_j} = 1$.
30: **end if**
31: **end for**
32: **if** ($VTCount$ is more than 50% of the votes) **then**
33: Transaction enters to the next round.
34: **if** ($VTCount$ less than the threshold value, that is, 80% of the votes) **then**
35: Continue from Step 26.
36: **else**
37: Send the commit response to all other followers peer nodes $CS'_j$.
Add block $BLK_i$ to the blockchain and adjourn the consensus process.
38: **end if**
39: **end if**

the efficiency of the system. The required steps are given below:

- **Step PBU1.** First of all $U_i$ needs to provide his/her current identity $ID^o_{U_i}$ and password $PW^o_{U_i}$ to the terminal of $MD_{U_i}$. $U_i$ also has to imprint his/her current biometrics information $BIO^o_{U_i}$ at the sensor of the specified device. Then $MD_{U_i}$ computes biometric secret key $\sigma^o_{U_i} = Rep(BIO^o_{U_i}, \tau_{U_i})$ with the condition that the hamming distance between the actual biometrics $BIO_{U_i}$ provided at the time of registration and currently furnished $BIO^o_{U_i}$ is less than or equal to the error tolerance threshold value $t$.
- **Step PBU2.** $MD_{U_i}$ again computes the required parameters such as $x = \beta \oplus h(ID_{U_i} || PW^o_{U_i} || \sigma^o_{U_i})$, $RPW^o_{U_i} = h(PW^o_{U_i} || x)$, $RID_{TA} = RID'_{TA} \oplus h(ID_{U_i} || x || \sigma^o_{U_i})$, $\alpha = \alpha' \oplus h(x || PW^o_{U_i} || \sigma^o_{U_i})$, $\delta^o = h(\alpha || RPW^o_{U_i} || \sigma^o_{U_i})$. After these computations, $MD_{U_i}$ further computes $EQ^o = h(ID_{U_i} || RID_{TA} || \delta^o)$, and checks the equality of $EQ^o = EQ$. If it holds, $U_i$ is validated as the genuine user and can proceed for password and biometric updates.
- **Step PBU3.** $U_i$ inputs new password $PW^n_i$ and imprints new biometric information $BIO^n_i$ according to his/her liking at the sensor of the specified device. Old biometrics information $BIO^o_i$ can also be used if $U_i$ does not wish to modify his/her biometrics information. In such a scenario, $BIO^n_i$ will be considered as $BIO^o_i$. Otherwise, $MD_{U_i}$ computes $Gen(BIO^n_i) = (\sigma^n_i, \tau^n_i)$. $MD_{U_i}$ then computes $RPW^n_{U_i} = h(PW^n_{U_i} || x)$, other parameters such as $\beta^n = x \oplus h(ID_{U_i} || PW^n_{U_i} || \sigma^n_{U_i})$, $RID^n_{U_i} = RID_{U_i} \oplus h(PW^n_{U_i} || \sigma^n_{U_i})$, $RID^n_{TA} = RID_{TA} \oplus h(ID_{U_i} || x || \sigma^n_{U_i})$, $\alpha^n = \alpha \oplus h(x || PW^n_{U_i} || \sigma^n_{U_i})$, $\delta^n = h(\alpha || RPW^n_{U_i} || \sigma^n_{U_i})$ and $EQ^n = h(ID_{U_i} || RID_{TA} || \delta^n)$. $MD_{U_i}$ then replaces $RID'_{U_i}$, $RID'_{TA}$, $\alpha'$, $EQ$, $\beta$, and $\tau_{U_i}$ with $RID^n_{U_i}$, $RID^n_{TA}$,

| $IMD_k$ | $PS_l$ | Blockchain $CS_j, CS'_j$ | $U_i$ |
|---|---|---|---|
| Generate current timestamp $TP_1$<br>Prepare $msg_1$<br>$= E_{SK_{IMD_k,PS_l}}(d_{IMD_k}, TP_1)$.<br>$msg_1 \xrightarrow{}$<br>(via open channel) | | | |
| | Decrypt $E_{SK_{IMD_k,PS_l}}(d_{IMD_k}, TP_1)$.<br>Get $d_{IMD_k}$ and $TP_1$.<br>Check $\lvert TP_1 - TP_1^*\rvert \leq \Delta T$.<br>If so, generate current timestamp $TP_2$.<br>Prepare $msg_2$<br>$= E_{SK_{PS_l,CS_j}}(d_{IMD_k}, TP_2)$.<br>$msg_2 \xrightarrow{}$<br>(via open channel) | | |
| | | Decrypt $E_{SK_{PS_l,CS_j}}(d_{IMD_k}, TP_2)$.<br>Get $d_{IMD_k}$ and $TP_2$.<br>Check $\lvert TP_2 - TP_2^*\rvert \leq \Delta T$. If so,<br>$CS_j$ prepares a block for addition into blockchain.<br>Block is added into blockchain when other miners committed successfully (Algorithms 1 and 2).<br>$CS'_j$ extracts $d_{IMD_k}$ from the concerned block.<br>$CS'_j$ generate current timestamp $TP_3$.<br>Prepare $msg_3 = E_{SK_{CS'_j,U_i}}(d_{IMD_k}, TP_3)$.<br>$msg_3 \xrightarrow{}$<br>(via open channel) | |
| | | | Decrypt $E_{SK_{CS'_j,U_i}}(d_{IMD_k}, TP_3)$.<br>Obtain $d_{IMD_k}$ and $TP_3$.<br>Check $\lvert TP_3 - TP_3^*\rvert \leq \Delta T$.<br>If so, obtain $d_{IMD_k}$ securely. |

**FIGURE 7.** Summary of blockchain based procedure for secure data exchange.

$\alpha^n$, $EQ^n$, $\beta^n$, and $\tau_{U_i}^n$, respectively. Finally, $MD_{U_i}$ stores the information $\{RID_{U_i}^n, RID_{TA}^n, \alpha^n, EQ^n, \beta^n, \tau_{U_i}^n, h(\cdot), Gen(\cdot), Rep(\cdot), t\}$ in its memory.

### H. DYNAMIC NODES ADDITION PHASE

Following steps can be used by the BAKMP-IoMT to add a new device i.e., *IMD* in the network any time.

#### 1) DYNAMIC IMD ADDITION

The addition of *IMD* can be done using the following steps.

- **Step DAI1.** A unique identity for new *IMD*, $ID_{IMD_v}$ is selected by the *TA* and it also computes the corresponding pseudo-identity $RID_{IMD_v} = h(ID_{IMD_v}||N)$ where the *TA*'s secret key is $N$. Next the *TA* determines a unique symmetric bivariate polynomial $\chi(x, y) = \sum_{m,n=0}^{t} a_{m,n}x^m y^n \in GF(p)[x, y]$ of degree $t$ over a finite field (Galois field) $GF(p)$ $(= Z_p)$, where the co-efficients $a_{i,j}$'s are chosen from $GF(p)$ and $Z_p = \{0, 1, 2, \cdots, p-1\}$ with $p$ being a satisfactorily large prime and $t$ is enough larger than the total count of IMDs to be deployed. For instance, a bivariate polynomial $\chi(x, y) = x^4 + 3x^3 + 2x^2y^2 + 3y^3 + y^4$ over $GF(5)$ is symmetric as $\chi(y, x) = y^4 + 3y^3 + 2y^2x^2 + 3x^3 + x^4 = \chi(x, y)$.
- **Step DAI2.** The *TA* computes a polynomial share $\chi(RID_{IMD_v}, y) = \sum_{m,n=0}^{t}[a_{m,n}(RID_{IMD_v})^m]y^n$, which is clearly a univariate polynomial of the same degree $t$. Then *TA* stores the credentials $\{RID_{IMD_v}, \chi(RID_{IMD_v}, y)\}$ in $IMD_v$'s memory before their deployment.

#### 2) DYNAMIC PERSONAL SERVER ADDITION

The addition of new personal server can be done using the specified steps.

- **Step DAP1.** The *TA* chooses a unique identity $ID_{PS_\eta}$ for new personal server $PS_\eta$ and determines corresponding pseudo-identity $RID_{PS_\eta} = h(ID_{PS_\eta}||N)$ utilizing the *TA*'s secret key $N$ and temporal credential $TC_{PS_\eta} = h(ID_{PS_\eta}||ID_{TA}||RTS_{PS_\eta}||N)$ where the registration timestamp of $PS_\eta$ is $RTS_{PS_\eta}$. *TA* next computes the polynomial share $\chi(RID_{PS_\eta}, y) = \sum_{m,n=0}^{t}[a_{m,n}(RID_{PS_\eta})^m]y^n$ for each $PS_\eta$, where $\chi(x, y) = \sum_{m,n=0}^{t} a_{m,n}x^m y^n \in GF(p)[x, y]$ is the same symmetric bivariate polynomial of degree $t$ that was previously chosen in Section IV-A1.
- **Step DAP2.** The *TA* stores the information $\{RID_{PS_\eta}, TC_{PS_\eta}, RID_{TA}, \chi(RID_{PS_\eta}, y))\}$ in $PS_\eta$'s database before its deployment.

#### 3) DYNAMIC CLOUD SERVER (MINERS) ADDITION

The addition of new cloud server can be done using the following steps.

- **Step DAC1.** The *TA* first chooses a unique identity $ID_{CS_j}^{new}$ for cloud server $CS_j^{new}$ and computes corresponding pseudo-identity $RID_{CS_j}^{new} = h(ID_{CS_j}^{new}||N)$ utilizing the *TA*'s secret key $N$.
- **Step DAC2.** After that *TA* stores the information $\{RID_{CS_j}^{new}, RID_{TA}\}$ in $CS_j^{new}$'s database before its deployment. *TA* then announce the addition of new cloud server to the other parties of the network through some secure channel.

*Remark 1: In BAKMP-IoMT, various involved communicating entities include implantable medical devices ($IMD_k$), personal server ($PS_l$), users having mobile devices ($U_i/MD_i$) and cloud servers ($CS_j$). Among all these entities, $IMD_k$, $PS_l$ and $MD_i$ are resource constrained end devices which have less computation, communication and storage capabilities as compared to the TA and cloud servers $CS_j$. Moreover, we need these devices for collection and local processing of essential healthcare related data. Apart from it, $CS_j$ are resource rich nodes which have high computation, communication and storage capabilities. Therefore, it is not typically desirable to use $IMD_k$, $PS_l$ and $MD_i$ for the block creation and consensus via voting tasks because of their resource constrained factor. Hence, in practice it is preferable to use the cloud servers ($CS_j$) for block consensus via voting and blockchain implementation. Though the RPCA mechanism is not that heavy, but still we preferred to execute block consensus via voting and blockchain implementation over the peer nodes ($CS_j$) in the P2P cloud servers network only in order to reduce the computational burden on $IMD_k$, $PS_l$ and $MD_i$. As discussed earlier, we consider a private blockchain in BAKMP-IoMT, which provides more security, and at the same time it has several restrictions and limited entities. As far as the annual fee is concerned, in most of the countries the healthcare facilities (including healthcare infrastructure) are supported by the government organisations. Therefore, payment of annual fee will not be an issue for healthcare infrastructure users in the proposed scheme.*

## V. SECURITY ANALYSIS OF BAKMP-IoMT

This section contains the "security analysis of BAKMP-IoMT" in Propositions 1–8, which prove the resilience of BAKMP-IoMT against the below mentioned attacks.

*Proposition 1: BAKMP-IoMT can resist "replay attack".*

*Proof:* In BAKMP-IoMT, we have used different current timestamps values in the transmitted messages. Each exchanged message of BAKMP-IoMT, have a maximum transmission delay $\Delta T$ component (a small value). Moreover, replaying the old transmitted messages does not provide any profit to the adversary $\mathcal{A}$ which were required for "authentication and key management procedure" among $IMD_k$, $PS_l$, $CS_j$ and $U_i$ within $\Delta T$. Therefore, BAKMP-IoMT is capable to resist replay attack. □

*Proposition 2: BAKMP-IoMT is able to protect against the "man-in-the-middle attack (MITM)".*

*Proof:* Let an adversary $\mathcal{A}$ eavesdrops an authentication request message $\{M_1, M_2, TS_1\}$ which was exchanged between $U_i$ and $CS_j'$, and after that $\mathcal{A}$ tries to update this message so that it resembles like the original authentication message, as $\{M_1^a, M_2^a, TS_1^a\}$ with the help of parameters such as $M_1^a = r_{U_i}^a \oplus h(RID_{TA}||RID_{U_i})$, and $M_2^a = h(r_{U_i}||RID_{U_i}||RID_{TA}||TS_1^a)$. For the launching of *MITM*, $\mathcal{A}$ can start the generation of random nonce $r_{U_i}^a$ and current timestamp $TS_1^a$. However, in the absence of knowledge of "long term secret" $RID_{TA}$, $RID_{U_i}$ and $N$ the secret key of *TA*, $\mathcal{A}$ can not regenerate another valid authentication request message $\{M_1^a, M_2^a, TS_1^a\}$. Similarly, we can also elucidate that other messages can not be computed again by $\mathcal{A}$ which are utilized in the "authentication and key management" phase without the "long-term secrets" utilized by $U_i$ and $CS_j'$ or the other entities of the network. Hence, BAKMP-IoMT is secured against the man-in-the-middle attack. □

*Proposition 3: BAKMP-IoMT is secured against various "impersonation attacks".*

*Proof:* Let an adversary $\mathcal{A}$ tries to impersonate as an valid communicating entity of the network by creating an authentication request message on behalf of that entity say $U_i$. After obtaining $U_i$'s authentication request $Msg_1 = \{M_1, M_2, TS_1\}$ which was sent to $CS_j'$ where $M_1 = r_{U_i} \oplus h(RID_{TA}||RID_{U_i})$ and $M_2 = h(r_{U_i}||RID_{U_i}||RID_{TA}||TS_1)$. Here it is important to notice that $Msg_1$ uses $M_1$ and $M_2$ which are generated through "long term secrets" i.e., $RID_{U_i}$, $RID_{TA}$ and $N$, and also through the "short term secrets" for example, random nonce $r_{U_i}$. $\mathcal{A}$ is not capable to generate a valid "authentication request message" representing the legitimate user $U_i$ without having the knowledge of these secret values. Therefore, BAKMP-IoMT is secured against the "user impersonation attack". By using the same methodology, we can prove that BAKMP-IoMT is also secured against other types of impersonation attacks i.e., "cloud server", "personal server" and "implantable medical device", because the creation of other messages also utilize both "long term" and "short term" secrets. Hence, BAKMP-IoMT is resilient against the various impersonations attacks. □

*Proposition 4: BAKMP-IoMT protects "Ephemeral Secret Leakage (ESL) attack".*

*Proof:* In BAKMP-IoMT, "session key" is calculated by a legitimate user $U_i$ and the cloud server $CS_j'$, during the "authentication and key management process" as $SK_{CS_j',U_i} = h(r_{U_i}||RID_{U_i}||TS_1||TS_2||\ h(r_{CS_j'}||RID_{CS_j'})||RID_{TA})$. In the creation of the session key, the pseudo identities $RID_{U_i}$, $RID_{CS_j'}$ of user $U_i$ and cloud server $CS_j'$ are used. It also consists of the different random nonce values $r_{U_i}$, $r_{CS_j'}$ of user and cloud server. It is important to notice that "session key" is the amalgamation of both the session-temporary (ephemeral) information also known as "short term secrets" (various timestamp and random nonce values) along with "long-term secrets" (various identities and secret keys). Therefore, session key can only be revealed in a situation if $\mathcal{A}$ compromises both the "short-term" and "long-term" secret values. Furthermore, as various random nonce and timestamps values are utilized in calculation of the session keys among various entities i.e., $U_i$ and $CS_j'$, $IMD_k$ and $PS_l$, and $PS_l$ and $CS_j$ in all different sessions, even if a session key is revealed for a specific session. It will not cause the revealing (i.e., illegal computation) of session keys of other sessions because of coalescence of short and long term secret values. Thus, BAKMP-IoMT is capable to protect session-temporary information attack and it also preserves the "perfect forward secrecy" goal. Hence, BAKMP-IoMT is secured against "ESL attack". □

*Proposition 5: BAKMP-IoMT is capable to protect "privileged-insider attack".*

*Proof:* The privileged-insider user of the trusted authority i.e., attacker $\mathcal{A}$ knows the registration information of the various entries i.e., $IMD_k$, $PS_l$ $CS_j$ and $U_i$. Still, he/she is not able to calculate the "session key" on behalf of a genuine communicating entity as the "session key" is created using the credentials that are only known to that particular entity i.e., $IMD_k$, $PS_l$ $CS_j$ and $U_i$. The "session key" computed by a legitimate cloud server $CS_j'$ is $SK_{CS_j',U_i} = h(r_{U_i}||RID_{U_i}||TS_1||TS_2||h(r_{CS_j'}||RID_{CS_j'})||RID_{TA})$ which consists of various short and long term secret values i.e., random nonces, timestamps, secret keys and identities, as elaborated earlier. The privileged-insider user of the *TA* does not have the knowledge of this information. Thus $\mathcal{A}$ is not able the compute the "session key" on the behalf of a legitimate communicating entity. Moreover, he/she can not impersonate as the "legitimate communicating entity i.e., $CS_j'$ " as explained in Proposition 3. Therefore, BAKMP-IoMT is able to protect the "privileged-insider attack". □

*Proposition 6: BAKMP-IoMT preserves anonymity and untraceability properties.*

*Proof:* Suppose an adversary $\mathcal{A}$ seizes the messages $MSG_1 = M_1, M_2, TS_1, MSG_2 = M_3, M_4, TS_2$ and $MSG_3 = M_5, TS_3$ during the "authentication and key management phase" between $U_i$ and $CS_j'$. These messages are computed by using the various random nonce values and current timestamps values which help to generate dynamic and unique messages in distinct sessions. Apart from that any identity information is not transmitted in the "plaintext format". Similar method is also followed for other exchanged messages among other entities i.e., $IMD_k$ and $PS_l$, and $PS_l$ and $CS_j$. This mechanism helped us to achieve both "anonymity and untraceability" properties in BAKMP-IoMT. □

*Proposition 7: BAKMP-IoMT is resilient against "implantable medical device (IMD)" physical capture attack.*

*Proof:* Each IMD contains the credentials $\{RID_{IMD_k}, \chi(RID_{IMD_k}, y)\}$ that are utilized for the "authentication and key establishment" related work with different communicating entities. To safeguard against IMD device physical capture attack is an crucial requirement from the security point of view [55], [56]. Let say $n_c$ IMD devices are physically trapped by an adversary $\mathcal{A}$. We do evaluation of "IMD physical capture attack" as the fraction of total secure communications which are compromised from the capturing (physical stolen) of $n_c$ IMDs *not including* the communication in which the "compromised IMD" are clearly extended. For instance, one can calculate the probability of $\mathcal{A}$'s expertise to decrypt the "secure communication" between a personal server $PS_l$ and a non-compromised IMD $IMD_k$ when $n_c$ IMDs are already compromised (under the influence of attack). Let's assume this probability as $P_e(n_c)$. If $P_e(n_c) = 0$, an "authentication and key management" scheme is treated as the "unconditionally secure against IMD physical capture attack". From a physically stolen IMD $IMD_k$, attacker $\mathcal{A}$ will have deduced information i.e., $\{RID_{IMD_k}, \chi(RID_{IMD_k}, y)\}$ along with the secret pairwise session key $SK_{PS_l,IMD_k} = h(\omega||TIP_1)$ shared between $IMD_k$ and $PS_l$ from its memory, where $\omega = \chi(RID_{PS_l}, RID_{IMD_k})$. However, it is worthy to notice that all $RID_{IMD_k}$ and $\chi(RID_{IMD_k}, y)$ are different for different IMDs. Therefore, the physical stolen of $IMD_k$ by $\mathcal{A}$ can only help him/her in the obtaining of secret session key between that $IMD_k$ and $PS_l$ not the other session keys. Thus all other secret keys between the $PS_l$ and other non-compromised IMDs are still unrevealed. Hence, the compromising of an IMD will not cause the compromising of the entire communication, and the secure communications among personal server and other non-compromised IMDs can be still achieved. Thus BAKMP-IoMT is unconditionally secure against the IMD physical capture attack. □

*Proposition 8: BAKMP-IoMT is resilient against the data modification attack at the cloud server.*

*Proof:* Cloud server (miner node) $CS_j$ receives data from the concerned personal server $PS_l$ and prepares a block to add this block into the existing blockchain. For that purpose, all miner nodes call the steps of Algorithm 1 and Algorithm 2. After the completion of all steps of these algorithms, a block will be added into the blockchain. Since, blockchain is a tamper proof technology and the attacker $\mathcal{A}$ does not have ability to update the required fraction of blocks. Therefore, $\mathcal{A}$ can not modify the data of a block. Hence, BAKMP-IoMT is resilient against the data modification attack. □

## VI. FORMAL SECURITY VERIFICATION USING AVISPA TOOL

In this section, we do simulation of the proposed BAKMP-IoMT for the "formal security verification using the widely accepted automated software validation tool, called AVISPA" [57].

A tested security protocol is first implemented under the High Level Protocol Specification Language (HLPSL) [58]. HLPSL is a "role-oriented language" containing the following types of roles:

- Basic roles: They denote "different participating entities in the protocol".
- Composition roles: They present "different scenarios involving basic roles".

An intruder always takes part in the protocol as "one of the basic legitimate roles and is always represented by *i*". The "HLPSL specification of the protocol" is converted into the "Intermediate Format (IF)" using the HLPSL2IF translator. The IF is then given as input to one of the four backends available in AVISPA to its "Output Format (OF)". The four back-ends of AVISPA are: a) "On-the-fly Model-Checker (OFMC)", b) "Constraint Logic based Attack Searcher (CL-AtSe)", c) "SAT-based Model-Checker (SATMC)" and d) "Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)". More detailed

discussions on these back-ends and HLPSL can be found in [57], [58].

The OF contains following sections [58]:

- SUMMARY: It tells ''whether the tested protocol is safe, unsafe, or whether the analysis is inconclusive''.
- DETAILS: It states an explanation of ''why the tested protocol is concluded as safe, or under what conditions the test application or protocol is exploitable using an attack, or why the analysis is inconclusive''.
- PROTOCOL: It denotes the ''HLPSL specification of the target protocol in the IF''.
- GOAL: It presents the ''goal of the analysis which is being performed by AVISPA using HLPSL specification''.
- BACKEND: It is the ''name of the back-end that is used for the analysis, that is, one of OFMC, CL-AtSe, SATMC and TA4SP''.
- Final section includes the ''trace of a possible vulnerability to the target protocol, if any, along with some useful statistics and relevant comments''.

We have implemented the registration, login and authentication phases of our proposed BAKMP-IoMT in HLPSL specification. The basic roles for the *TA*, a user $U_i$ and a cloud server $CS_j$ are shown in figures 8, 9 and 10, respectively. The mandatory roles (session and goal & environment) are also implemented as composite roles in Figure 11.

```
role trustedauthority(U, TA, CS: agent, H: hash_func,
                      Snd, Rcv: channel(dy))
played_by TA
def=
local State: nat,
    IDui, IDta, IDcsj, N, RTSui, RIDta, RIDcsj : text,
    RIDui, Alpha, CSj,
    SKuta, SKcsta: symmetric_key
const sp1, sp2, sp3: protocol_id
% Initialize state, State to 0
init State := 0
transition
%%% User registration phase
%%% Receive registration request from user (U) securely
1. State = 0 /\ Rcv({IDui}_SKuta) =|>
 State' := 1 /\ RTSui' := new()
          /\ RIDui' := H(IDui.N) /\ RIDta' := H(IDta.N)
          /\ Alpha' := H(RIDui'.RIDta'.RTSui')
%%% Send registration response to U securely
          /\ Snd({RIDui'.Alpha'.RIDta'}_SKuta)
          /\ secret({IDui}, sp1, {U,TA})
          /\ secret({IDta, RTSui',N}, sp2, TA)

%%% Cloud server registration phase
%%% Send registration request to cloud server (CS) securely
          /\ RIDcsj' := H(IDcsj.N)
          /\ Snd({RIDcsj'.RIDta'}_SKcsta)
          /\ secret({IDcsj}, sp3, {CS,TA})
end role
```

**FIGURE 8.** HLPSL specification for the role of *TA*.

In the simulation under the AVISPA backends, three verifications, namely: a) ''executability checking on non-trivial HLPSL specifications'', b) ''replay attack checking'', and c) ''Dolev-Yao threat model (DY model) [42] checking''. The ''executability check'' is essential to assure that ''the protocol

```
role user(U, TA, CS: agent, H: hash_func,
          Snd, Rcv: channel(dy))
played_by U
def=
local State: nat,
    IDui, N, IDta, RTSui, TS1, Rui, M1, M2, M5,
    Rcsj, IDcsj, TS2, TS3, SKucs :text,
    SKuta: symmetric_key
const sp1, sp2, sp3, u_cs_ts1, u_cs_rui, cs_u_rcsj,
    cs_u_ts2, u_cs_ts3: protocol_id
% Initialize state, State to 0
init State := 0
transition
%%% User registration phase
1. State = 0 /\ Rcv(start) =|>
%%% Send registration request to the TA securely
 State' := 2 /\ Snd({IDui}_SKuta)
          /\ secret({IDui}, sp1, {U,TA})
%%% Receive registration response from the TA securely
2. State = 2 /\ Rcv({H(IDui.N).H(H(IDui.N).H(IDta.N).
                RTSui').H(IDta.N)}_SKuta) =|>
 State' := 4 /\ secret({IDta, RTSui',N}, sp2, TA)

%%% User login phase
          /\ TS1' := new() /\ Rui' := new()
          /\ M1' := xor(Rui', H(H(IDta.N).H(IDui.N)))
          /\ M2' := H(Rui'.H(IDui.N).H(IDta.N).TS1')
%%% Send login request to CS via public channel
          /\ Snd(M1'.M2'.TS1')
%%% U has freshly generated the values TS1 and Rui for the CS
          /\ witness (U, CS, u_cs_ts1, TS1')
          /\ witness (U, CS, u_cs_rui, Rui')

%% Authentication and key agreement phase
%% Receive authentication request from the CS via public channel
3. State = 4 /\ Rcv( xor(H(Rcsj'.H(IDcsj.N)), H(H(IDui.N).H(IDta.N))).
                H(H(Rui'.H(IDui.N).TS1'.TS2'.H(Rcsj'.H(IDcsj.N)).
                H(IDta.N)).TS2').
                TS2') =|>
 State' := 6 /\ TS3' := new()
          /\ SKucs' := H(Rui'.H(IDui.N).TS1'.TS2'.
                H(Rcsj'.H(IDcsj.N)).H(IDta.N))
          /\ M5' := H(SKucs'.TS3')
%%% Send authentication reply to CS via public channel
          /\ Snd(M5'.TS3')
%%% U's acceptance of values TS2 and Rcsj generated for U by CS
          /\ request (CS, U, cs_u_ts2, TS2')
          /\ request (CS, U, cs_u_rcsj, Rcsj')
end role
```

**FIGURE 9.** HLPSL specification for the role of a user $U_i$.

will reach to a state where a possible attack can happen, during the run of the protocol''. From figures 8, 9 and 10, it is clear that ''BAKMP-IoMT has been properly translated to HLPSL specification and it meets the design goals by ensuring the executability''. For the ''replay attack checking'' on BAKMP-IoMT, both OFMC and CL-AtSe check ''if the legitimate agents can execute the specified protocol by performing a search of a passive intruder''. Finally, both OFMC and CL-AtSe backends verify ''whether any man-in-the-middle attack is possible by *i* for the DY model checking''. The simulation results reported in figures 12 and 13 clearly exhibit that the proposed BAKMP-IoMT is safe against both 'replay'' and 'man-in-the-middle'' attacks.

## VII. PERFORMANCE ANALYSIS

In this section, we first evaluate the performance of the user registration, and user login and authentication phases of our

```
role cloudserver(U, TA, CS: agent, H: hash_func,
                 Snd, Rcv: channel(dy))
played_by CS
def=
local State: nat,
      IDcsj, N, IDta, IDui, RTSui, Rui, TS1,
      TS2, Rcsj, M3, SKcsu, M4, TS3: text,
      SKcsta: symmetric_key
const sp1, sp2, sp3, u_cs_ts1, u_cs_rui, cs_u_rcsj,
      cs_u_ts2, u_cs_ts3: protocol_id
% Initialize state, State to 0
init State := 0
transition
%%% Cloud server registration phase
%%% Receive registration request to the TA securely
1. State = 0 /\ Rcv({H(IDcsj.N).H(IDta.N)}_SKcsta) =|>
 State' := 3 /\ secret({IDui}, sp1, {U,TA})
          /\ secret({IDta, RTSui',N}, sp2, TA)
          /\ secret({IDcsj}, sp3, {CS,TA})

%%% User login phase
%%% Receive login request message from U via public channel
2. State = 3 /\ Rcv(xor(Rui', H(H(IDta.N).H(IDui.N))).
                H(Rui'.H(IDui.N).H(IDta.N).TS1').TS1') =|>

%%% Authentication and key agreement phase
 State' := 5 /\ TS2' := new() /\ Rcsj' := new()
          /\ M3' := xor(H(Rcsj'.H(IDcsj.N)), H(H(IDui.N).H(IDta.N)))
          /\ SKcsu' := H(Rui'.H(IDui.N).TS1'.TS2'.
                    H(Rcsj'.H(IDcsj.N)).H(IDta.N))
          /\ M4' := H(SKcsu'.TS2')
%%% Send authentication request to U via public channel
          /\ Snd(M3'.M4'.TS2')
%%% CS has freshly generated the values TS2 and Rcsj for U
          /\ witness (CS, U, cs_u_ts2, TS2')
          /\ witness (CS, U, cs_u_rcsj, Rcsj')
%%% Receive authentication reply from U via public channel
3. State = 5 /\ Rcv(H(Rui'.H(IDui.N).TS1'.TS2'.H(Rcsj'.
                H(IDcsj.N)).H(IDta.N)).TS3') =|>
% CS's acceptance of values TS1, Rui and TS3 generated for CS by U
 State' := 7 /\ request (U, CS, u_cs_ts1, TS1')
          /\ request (U, CS, u_cs_rui, Rui')
          /\ request (U, CS, u_cs_ts3, TS3')
end role
```

**FIGURE 10.** HLPSL specification for the role of a cloud server $CS_j$.

```
role session(U, TA, CS: agent, H: hash_func)
def=
  local SD1, RV1, SD2, RV2, SD3, RV3: channel (dy)
  composition
    trustedauthority(U, TA, CS, H, SD1, RV1)
 /\ user(U, TA, CS, H, SD2, RV2)
 /\ cloudserver(U, TA, CS, H, SD3, RV3)
end role

role environment()
def=
  const u, ta, cs: agent,
      h: hash_func,
      ts1, ts2, ts3 : text,
      sp1, sp2, sp3, u_cs_ts1, u_cs_rui,
      cs_u_rcsj, cs_u_ts2, u_cs_ts3: protocol_id
intruder_knowledge = {u, ta, cs, ts1, ts2, ts3}
composition
   session(u, ta, cs, h)
/\ session(u, ta, cs, h)
/\ session(i, ta, cs, h)
/\ session(u, ta, i, h)
end role

goal
%%% Confidentiality (privacy)
secrecy_of sp1, sp2, sp3
%%% Authentication
authentication_on u_cs_ts1, u_cs_rui
authentication_on cs_u_rcsj, cs_u_ts2
authentication_on u_cs_ts3
end goal
environment()
```

**FIGURE 11.** HLPSL specification for the roles of session, goal and environment.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
  /home/akdas/Desktop/span
  /testsuite/results/auth.if

GOAL
   as specified

BACKEND
   OFMC

STATISTICS

  TIME 766 ms
  parseTime 0 ms
  visitedNodes: 748 nodes
  depth: 9 plies
```

**FIGURE 12.** Analysis of results of BAKMP-IoMT under OFMC backend.

proposed BAKMP-IoMT. Next, we compare the performance of BAKMP-IoMT with the relevant existing authentication schemes for healthcare applications.

During the login and authentication phase, the messages $Msg_1 = \langle M_1, M_2, TS_1 \rangle$, $Msg_2 = \langle M_3, M_4, TS_2 \rangle$ and $Msg_3 = \langle M_5, TS_3 \rangle$ are exchanged between a registered user $U_i$ and a cloud server $CS_j$. Assume that an ''identity'', a ''timestamp'', a ''random number (nonce)'' and a ''hash output (if SHA-256 hashing algorithm is applied)'' need 160 bits, 32 bits, 160 bits and 256 bits, respectively. Then, the messages $Msg_1 = \langle M_1, M_2, TS_1 \rangle$, $Msg_2 = \langle M_3, M_4, TS_2 \rangle$ and $Msg_3 = \langle M_5, TS_3 \rangle$ require $(256+256+32) = 544$ bits, $(256+256+32) = 544$ bits, and $(256+32) = 288$ bits. As a result, the total communication cost in BAKMP-IoMT due to three messages exchange needs 1376 bits.

For computational cost analysis, assume that $T_h$ and $T_{fe}$ denote the time needed to execute a ''one-way hash function (say, SHA-256 hashing algorithm)'' and a ''fuzzy extractor function ($Gen(\cdot)/Rep(\cdot)$'', respectively. During the login and authentication phase, a user $U_i$ needs the computational cost of $12T_h + T_{fe}$ and a cloud server $CS_j$ requires computational cost of $7T_h$. Thus, the total computational cost in BAKMP-IoMT becomes $19T_h + T_{fe}$. Since only hash function and fuzzy extractor are applied, the proposed scheme is very efficient.

During the user registration process, a user $U_i$'s mobile device $MD_{U_i}$ requires the credentials $\{RID'_{U_i}, RID'_{TA}, \alpha', EQ, \beta, \tau_{U_i}, t\}$. If we assume that ''public reproduction parameter $\tau_{U_i}$'' and ''error-tolerance threshold value $t$'' require

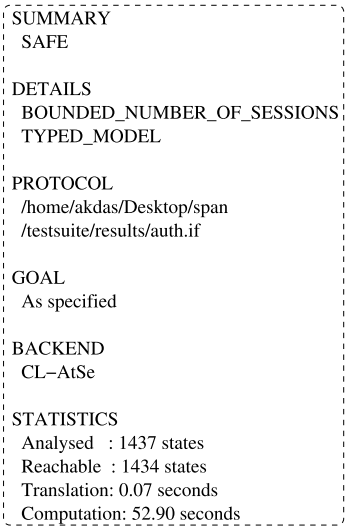


**FIGURE 13.** Analysis of results of BAKMP-IoMT under CL-AtSe backend.

160 bits and 32 bits, respectively, the storage cost needed to store these credentials in BAKMP-IoMT becomes $(256 + 256 + 256 + 256 + 256 + 160 + 32) = 1472$ bits.

We now perform a detailed comparative study for the "computation and communication costs" and "security and functionality features" among the proposed BAKMP-IoMT and other related existing schemes, such as the schemes of Jang *et al.* [36], He and Zeadally [37] and Merabet *et al.* [38].

In Table 3, we have shown a comparative analysis on "security and functionality features" among BAKMP-IoMT and other schemes against the considered features $SFF_1$–$SFF_{17}$. It is seen that the proposed BAKMP-IoMT provides superior security and also more "functionality features" while these are compared with the schemes of Jang *et al.* [36], He and Zeadally [37] and Merabet *et al.* [38].

**TABLE 3.** Comparison of security and functionality features.

| Feature | Merabet *et al.* [38] | Jang *et al.* [36] | He-Zeadally [37] | BAKMP-IoMT |
|---|---|---|---|---|
| $SFF_1$ | ✓ | ✓ | ✓ | ✓ |
| $SFF_2$ | ✓ | × | ✓ | ✓ |
| $SFF_3$ | ✓ | ✓ | ✓ | ✓ |
| $SFF_4$ | ✓ | ✓ | ✓ | ✓ |
| $SFF_5$ | × | ✓ | ✓ | ✓ |
| $SFF_6$ | ✓ | ✓ | ✓ | ✓ |
| $SFF_7$ | ✓ | ✓ | ✓ | ✓ |
| $SFF_8$ | × | ✓ | ✓ | ✓ |
| $SFF_9$ | ✓ | ✓ | ✓ | ✓ |
| $SFF_{10}$ | ✓ | N/A | ✓ | ✓ |
| $SFF_{11}$ | N/A | N/A | N/A | ✓ |
| $SFF_{12}$ | N/A | N/A | N/A | ✓ |
| $SFF_{13}$ | × | × | × | ✓ |
| $SFF_{14}$ | × | × | × | ✓ |
| $SFF_{15}$ | × | N/A | ✓ | ✓ |
| $SFF_{16}$ | ✓ | ✓ | ✓ | ✓ |
| $SFF_{17}$ | ✓ | × | × | ✓ |

Note: $SFF_1$: "mutual authentication/access control"; $SFF_2$: "anonymity"; $SFF_3$: "untraceability"; $SFF_4$: "session-key agreement"; $SFF_5$: "session key security under CK adversary model"; $SFF_6$: "confidentiality"; $SFF_7$: "integrity"; $SFF_8$: "strong replay attack"; $SFF_9$: "man-in-the-middle attack"; $SFF_{10}$: "efficient login phase"; $SFF_{11}$: "password update phase"; $SFF_{12}$: "biometric update phase"; $SFF_{13}$: "dynamic controller node addition"; $SFF_{14}$: "dynamic $IMD$ addition"; $SFF_{15}$: "protection against stolen mobile device/programmer attack"; $SFF_{16}$: "protection against impersonation attack"; $SFF_{17}$: "formal security verification using AVISPA tool".
×: "a scheme is insecure against a particular attack or it does not support a particular feature"; ✓: "a scheme is secure against a particular attack or supports a particular feature"; N/A: "not applicable in a scheme".

**TABLE 4.** Rough estimated time for various cryptographic primitives [59].

| Notation | Description (time to compute) | Approx. computation time (in seconds) |
|---|---|---|
| $T_h$ | "One-way hash function" | 0.00032 |
| $T_{ecm}$ | "ECC point multiplication" | 0.0171 |
| $T_{eca}$ | "ECC point addition" | 0.0044 |
| $T_{senc}$ | "Symmetric key encryption" | 0.0056 |
| $T_{sdec}$ | "Symmetric key decryption" | 0.0056 |
| $T_{me}$ | "Modular exponentiation" | 0.0192 |
| $T_{fe}$ | "Fuzzy extractor function" | 0.0171 |

**TABLE 5.** Computation costs comparison.

| Scheme | Computation cost (in milliseconds) |
|---|---|
| Jang *et al.* | $25T_{ecm} + 15T_{eca} + 5T_h \approx 495.10$ ms |
| He-Zeadally | $6T_{ecm} + 8T_{senc}/T_{sdec} + 4T_h \approx 148.70$ ms |
| Protocol-I (Merabet *et al.*) | $6T_{ecm} + 6T_h + T_{eca} \approx 108.92$ ms |
| Protocol-II (Merabet *et al.*) | $4T_{ecm} + 4T_h \approx 69.68$ ms |
| BAKMP-IoMT | $19T_h + T_{fe} \approx 23.18$ ms |

For comparative analysis on computational costs among the proposed BAKMP-IoMT and other schemes during login and authentication phases, we use various execution time for cryptographic primitives as listed in Table 4 based on the existing experimental results in [59]. Using these results, a comparative study on computational costs among the proposed BAKMP-IoMT and other schemes has been done in Table 5. It is also worth noticing that BAKMP-IoMT requires low computational cost as compared to that for other schemes, such as the schemes of Jang *et al.* [36], He and Zeadally [37] and Merabet *et al.* [38].

Finally, a comparative study on communication costs among the proposed BAKMP-IoMT and other schemes is also provided in Table 6. It is observed that BAKMP-IoMT

**TABLE 6.** Communication overheads comparison.

| Scheme | No. of messages | No. of bits |
|---|---|---|
| Jang *et al.* | 8 | 5920 |
| He-Zeadally | 4 | 3232 |
| Protocol-I (Merabet *et al.*) | 3 | 1472 |
| Protocol-II (Merabet *et al.*) | 3 | 1472 |
| BAKMP-IoMT | 3 | 1376 |

requires low communication cost as compared to that for other schemes, such as the schemes of Jang *et al.* [36], He and Zeadally [37] and Merabet *et al.* [38].

## VIII. PRACTICAL DEMONSTRATION

The pragmatic delineation of BAKMP-IoMT using the strategies of blockchain was proceeded as follows [60]. Table 7 represents the details of different parameters used during the experimentation. Three unique scenarios were considered in the experimentation. The experimentation was conducted on a platform having Windows 10 64-bit OS with Intel (R) core i5-8250U, 1.60 GHz-1.80 GHz processor. The size of random-access memory (RAM) size was 8 GB. The programming platform was eclipse IDE 2019-12 with Java language. The count of IMDs considered were 50 (in scenario-1), 100 (in scenario-2) and 150 (in scenario-3). The number of computed and committed blocks were 5 (in scenario-1), 10 (in scenario-2) and 15 (in scenario-3). The count of users was taken as 10 (in scenario-1), 20 (in scenario-2) and 40 (in scenario-3) alongside four miner nodes (i.e., cloud servers). The voting based mechanism is used for the block verification and addition purpose. The snippets of the ''main blockchain program'' and the ''data inside a block of blockchain'' are given in Figures 14 and 15, respectively. Various fields utilized in a block of the blockchain are as follows:

- **Block Version (BVer):** It denotes the version of a block. The size of this field can be assumed as 32 bits.
- **Hash of previous block (PBHash):** It consists the hash value of the previous block. The size of this field can be considered as 256-bits (if ''SHA256 hash algorithm'' is taken).
- **Merkle Tree Root (MR):** It is the Merkle tree root on the encrypted transactions, whose size is 256-bits as ''SHA-256 hash algorithm'' is used.
- **Timestamp:** It is the value of timestamp for a particular block. The size of this field can be considered as 32-bits.
- **Owner of Block (OB):** It represents the identity of block owner. The size of this field can be considered as 160-bits.
- **Owner public key:** This field have the information of ''public key of the owner (miner) node''. The size of this field is considered as 320-bits (since the ''Elliptic Curve Cryptography (ECC) algorithm'' is considered).
- **Encrypted transaction details:** It comprises the information about the ''ongoing transactions''. For example, which entity (communicating party) is sending ''which information'' and for ''what reason''. The size of each encrypted transaction is ECC-based ciphertext. Thus, it requires $(320 + 320) = 640$ bits. If we consider 100 encrypted transactions, the payload of the block becomes $(100 \times 640) = 64,000$ bits.

**TABLE 7.** Simulation parameters used BAKMP-IoMT.

| Parameter | Remark |
|---|---|
| Platform used | Windows 10 64-bit OS |
| Processor | Intel (R) core (TM), i5-8250U, 1.60 GHz-1.80 GHz |
| Random-access memory (RAM) size | 8 GB |
| Programming platform | Eclipse IDE 2019-12 with Java |
| Number of IMDs | 50 (scenario-1), 100 (scenario-2), 150 (scenario-3) |
| Number of users | 10 (scenario-1), 20 (scenario-2), 40 (scenario-3) |
| Number of miner nodes | 4 in all scenarios |
| Size of a block | 65,632 bits |

```
import java.util.ArrayList;
import java.util.Arrays;
public class Block {
        private int bver;// block's version
        private int ts;// timestamp
        private String oid;// ownerid, owberinfo
        private String puk; // public key of owner
        private String previousHash;// hash of previous block
        private String[] transactions; // details of transactions
        private String mr;// merkle root
        private String sign;// block's signature
        private String blockHash; // hash of this block
        public Block(int bver, int ts, String oid, String puk,
        String previousHash, String[] transactions, String mr) {
                this.bver = bver;
                this.ts = ts;
                this.oid = oid;
                this.puk = puk;
                this.previousHash = previousHash;
                this.transactions = transactions;
                Object[] contens = {Arrays.hashCode(transactions),
                previousHash, ts, bver, oid, puk, mr};
                int xa = Arrays.hashCode(contens);
                String xaz = Integer.toString(xa);
                this.blockHash = xaz;
                int mrz=Arrays.hashCode(transactions);
                mr = Integer.toString(mrz);
                this.mr=mr;
                }
```

**FIGURE 14.** Snippet of the main blockchain program.

```
//block10
int bver10=10;
int ts10=110;
String oid10=vt;
String puk10=pu;
String[] block10Transactions = {
"SD51 sends BP of patient  135-85", "SD12 sends glucose level 140 mg/dL",
"SD52 sends temprature 99 degree F", "SD22 sends BP of patient 130-85 ",
                                        ...};
int mrz10=Arrays.hashCode(genesisTransactions);
String mr10= Integer.toString(mrz10);
Block block10 = new Block (bver10, ts10, oid10, puk10,  mst8,
block10Transactions, mr10);
String s9= block10.getBlockHash();
int s99=Integer.parseInt(s9);
int ms9=Math.abs(s99);
String mst9= Integer.toString(ms9);
System.out.println("Hash of block 10:");
System.out.println(mst9);
String str10 = mst9;
byte[] str10Byte = str10.getBytes("UTF-8");
dsa.update(str10Byte);
byte[] realSig10 = dsa.sign();
System.out.println("Content of block10:");
System.out.println(bver10); System.out.println(oid10);
System.out.println(puk10); System.out.println(ts10);
System.out.println(mr10); System.out.println(mst8);
System.out.println(mst9); System.out.println("Signature of block10:
" + new BigInteger(1, realSig10).toString(16));
```

**FIGURE 15.** Snippet of data inside a block of blockchain.

- **Hash of current block:** It contains the hash value of the current block. The size of this field is 256-bits (if "SHA-256 algorithm" is taken).
- **Block signature:** It contains the "signature information of a particular block". The size of this field requires 320-bits (if ECC algorithm is applied).

The following results were obtained during the re-enactments.

### A. IMPACT ON COMPUTATIONAL COST

The effect on increasing number of IMDs and users was computed as the computation cost (in ms) for all considered scenarios. The various estimations of computation costs are 4.20, 5.01 and 6.21 seconds for scenario-1, scenario-2 and scenario-3, respectively. The reported results are likewise delineated in Figure 16. It is worthy to see that "computation cost" increases with the number of IMDs and users from scenario-1 to scenario-2 as well as from scenario-2 to scenario-3 as increase in number of IMDs and users causes "creation and addition" of more number of blocks in the blockchain.

### B. IMPACT ON TRANSACTION PER SECOND (TPS)

The impact on our proposed BAKMP-IoMT on transactions per second (TPS) is also measured as per the considered scenarios. The values of TPS are 119, 200 and 242 for scenario-1, scenario-2 and scenario-3, respectively. The reported results are additionally delineated in Figure 17. It is worthy to see that the value of TPS increases as the blockchain grows with more number of IMDs and users. The reason is straightforward because it causes the creation and addition of more number of blocks in the blockchain.

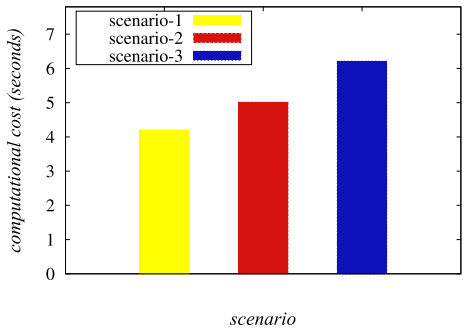


**FIGURE 16.** Obtained results of BAKMP-IoMT: impact on computational cost.

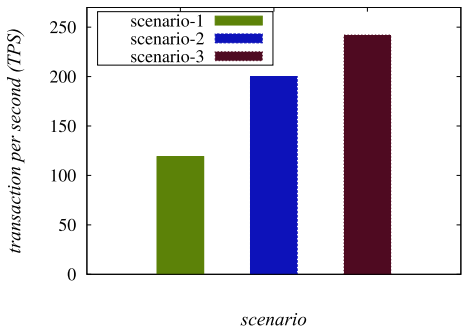


**FIGURE 17.** Obtained results of BAKMP-IoMT: impact on transaction per second (TPS).

## IX. CONCLUSION

In this paper, we designed a novel blockchain enabled authentication key agreement protocol for IoMT environment (BAKMP-IoMT). BAKMP-IoMT provides secure key management among different communicating entities. The legitimate users can also access the healthcare data from the cloud servers in a secure way. The entire healthcare data is stored in a blockchain maintained by the cloud servers. The formal security verification of BAKMP-IoMT is also performed to demonstrate its resilience against the different types of possible attacks using the widely-accepted AVISPA tool and also through formal and informal security analysis. BAKMP-IoMT is compared with other related existing schemes and it also performs better in terms of security and functionality features, less communication and communication costs for authentication and key managment phase as compared to the other schemes. Furthermore, the simulation of BAKMP-IoMT is conducted to demonstrate its impact on the performance parameters.

## ACKNOWLEDGMENTS

The authors thank the anonymous reviewers and the associate editor for their valuable feedback on the paper which helped them to improve its quality and presentation.

## REFERENCES

[1] B. Bera, D. Chattaraj, and A. K. Das, "Designing secure blockchain-based access control scheme in IoT-enabled Internet of drones deployment," *Comput. Commun.*, vol. 153, pp. 229–249, Mar. 2020.

[2] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.

[3] A. K. Das, A. K. Sutrala, S. Kumari, V. Odelu, M. Wazid, and X. Li, "An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 2070–2092, 2016.

[4] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, "Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 1983–2001, Sep. 2016.

[5] S. Chatterjee, A. K. Das, and J. K. Sing, "A secure user anonymity-preserving three-factor remote user authentication scheme for the telecare medicine information systems," *Adhoc SensorWireless Netw.*, vol. 21, no. 1, pp. 121–149, 2014.

[6] A. K. Das, "A secure user anonymity-preserving three-factor remote user authentication scheme for the telecare medicine information systems," *J. Med. Syst.*, vol. 39, no. 3, p. 30, Mar. 2015.

[7] V. Odelu, A. K. Das, M. Khurram Khan, K.-K.-R. Choo, and M. Jo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, vol. 5, pp. 3273–3283, 2017.

[8] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. Goutham Reddy, E.-J. Yoon, and K.-Y. Yoo, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[9] V. Odelu, A. K. Das, and A. Goswami, "SEAP: Secure and efficient authentication protocol for NFC applications using pseudonyms," *IEEE Trans. Consum. Electron.*, vol. 62, no. 1, pp. 30–38, Feb. 2016.

[10] A. K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, and X. Huang, "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3670–3687, Nov. 2016.

[11] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9762–9773, Dec. 2019.

[12] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. Park, "AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8804–8817, Oct. 2019.

[13] N. Kumar, G. S. Aujla, A. K. Das, and M. Conti, "ECCAuth: A secure authentication protocol for demand response management in a smart grid system," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6572–6582, Dec. 2019.

[14] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[15] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K. R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3184–3197, Apr. 2020.

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

[17] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 391–406, Mar. 2020.

[18] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[19] J. Srinivas, A. K. Das, N. Kumar, and J. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Trans. Dependable Secure Comput.*, early access, Apr. 19, 2018, doi: 10.1109/TDSC.2018.2828306.

[20] M. Shuai, B. Liu, N. Yu, L. Xiong, and C. Wang, "Efficient and privacy-preserving authentication scheme for wireless body area networks," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102499.

[21] A. K. Sutrala, P. Bagga, A. K. Das, N. Kumar, J. J. P. C. Rodrigues, and P. Lorenz, "On the design of conditional privacy preserving batch verification-based authentication scheme for Internet of vehicles deployment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5535–5548, May 2020, doi: 10.1109/TVT.2020.2981934.

[22] S. Challa, A. K. Das, V. Odelu, N. Kumar, S. Kumari, M. K. Khan, and A. V. Vasilakos, "An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Comput. Electr. Eng.*, vol. 69, pp. 534–554, Jul. 2018.

[23] M. Wazid, A. K. Das, N. Kumar, V. Odelu, A. Goutham Reddy, K. Park, and Y. Park, "Design of lightweight authentication and key agreement protocol for vehicular ad hoc networks," *IEEE Access*, vol. 5, pp. 14966–14980, 2017.

[24] A. A. Monrat, O. Schelen, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019.

[25] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, p. 352, 2018.

[26] S. Aggarwal, R. Chaudhary, G. S. Aujla, N. Kumar, K.-K.-R. Choo, and A. Y. Zomaya, "Blockchain for smart communities: Applications, challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 144, pp. 13–48, Oct. 2019.

[27] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K.-R. Choo, "HomeChain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 818–829, Feb. 2020.

[28] H. Zhang, J. Wang, and Y. Ding, "Blockchain-based decentralized and secure keyless signature scheme for smart grid," *Energy*, vol. 180, pp. 955–967, Aug. 2019.

[29] X. Wang, P. Zeng, N. Patterson, F. Jiang, and R. Doss, "An improved authentication scheme for Internet of vehicles based on blockchain technology," *IEEE Access*, vol. 7, pp. 45061–45072, 2019.

[30] C. Lin, D. He, X. Huang, X. Xie, and K.-K.-R. Choo, "Blockchain-based system for secure outsourcing of bilinear pairings," *Inf. Sci.*, vol. 527, pp. 590–601, Jul. 2020, doi: 10.1016/j.ins.2018.12.043.

[31] R. Chaudhary, A. Jindal, G. S. Aujla, S. Aggarwal, N. Kumar, and K.-K.-R. Choo, "BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system," *Comput. Secur.*, vol. 85, pp. 288–299, Aug. 2019.

[32] Q. Feng, D. He, S. Zeadally, and K. Liang, "BPAS: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4146–4155, Jun. 2020.

[33] A. Jindal, G. S. Aujla, and N. Kumar, "SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment," *Comput. Netw.*, vol. 153, pp. 36–48, 2019.

[34] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 2567–2572.

[35] H. Zhang, G. Kou, and Y. Peng, "Soft consensus cost models for group decision making and economic interpretations," *Eur. J. Oper. Res.*, vol. 277, no. 3, pp. 964–980, Sep. 2019.

[36] C. S. Jang, D. G. Lee, J.-W. Han, and J. H. Park, "Hybrid security protocol for wireless body area networks," *Wireless Commun. Mobile Comput.*, vol. 11, no. 2, pp. 277–288, Feb. 2011.

[37] D. He and S. Zeadally, "Authentication protocol for an ambient assisted living system," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 71–77, Jan. 2015.

[38] F. Merabet, A. Cherif, M. Belkadi, O. Blazy, E. Conchon, and D. Sauveron, "New efficient M2C and M2M mutual authentication protocols for IoT-based healthcare applications," *Peer-to-Peer Netw. Appl.*, vol. 13, no. 2, pp. 439–474, Mar. 2020.

[39] B. Chase and E. MacBrough, "Analysis of the XRP ledger consensus protocol," 2018, *arXiv:1802.07242*. [Online]. Available: http://arxiv.org/abs/1802.07242

[40] C. Lin, D. He, X. Huang, M. K. Khan, and K.-K.-R. Choo, "DCAP: A secure and efficient decentralized conditional anonymous payment system based on blockchain," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2440–2452, 2020.

[41] L. Ismail and H. Materwala, "A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions," *Symmetry*, vol. 11, no. 10, p. 1198, 2019.

[42] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[43] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology*, B. Pfitzmann, Ed. Berlin, Germany: Springer 2001, pp. 453–474.

[44] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Advances in Cryptology*, L. R. Knudsen, Ed. Amsterdam, The Netherlands: Springer Berlin Heidelberg, 2002, pp. 337–351.

[45] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[46] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2884–2895, Aug. 2018.

[47] M. Wazid, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Secure three-factor user authentication scheme for Renewable-Energy-Based smart grid environment," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3144–3153, Dec. 2017.

[48] M. Wazid, A. K. Das, N. Kumar, M. Conti, and A. V. Vasilakos, "A novel authentication and key agreement scheme for implantable medical devices deployment," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 4, pp. 1299–1309, Jul. 2018.

[49] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 824–839, Sep. 2018.

[50] S. Challa, A. K. Das, S. Kumari, V. Odelu, F. Wu, and X. Li, "Provably secure three-factor authentication and key agreement scheme for session initiation protocol," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5412–5431, Dec. 2016.

[51] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Netw.*, vol. 20, pp. 1–15, Sep. 2014.

[52] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proc. 12th Annu. Int. Cryptol. Conf.* (Lecture Notes in Computer Science), vol. 740. Santa Barbara, California, USA, 1993, pp. 471–486.
[53] A. K. Das and I. Sengupta, "An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials," in *Proc. 3rd Int. Conf. Commun. Syst. Softw. Middleware Workshops (COMSWARE )*, Bangalore, India, Jan. 2008, pp. 9–16.
[54] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.
[55] A. K. Das, P. Sharma, S. Chatterjee, and J. K. Sing, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1646–1656, Sep. 2012.
[56] S. Kumari, A. K. Das, M. Wazid, X. Li, F. Wu, K. K. R. Choo, and M. K. Khan, "On the design of a secure user authentication and key agreement scheme for wireless sensor networks," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 23, p. e3930, 2017.
[57] AVISPA. (2019). *Automated Validation of Internet Security Protocols and Applications*. Accessed: Oct. 2019. [Online]. Available: http://www.avispa-project.org/
[58] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proc. 3rd APPSEM II (Appl. Semantics II) Workshop (APPSEM)*, Frauenchiemsee, Germany, 2005, pp. 1–17.
[59] D. He, N. Kumar, J. H. Lee, and R. S. Sherratt, "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 30–37, Feb. 2014.
[60] M. Fan and X. Zhang, "Consortium blockchain based data aggregation and regulation mechanism for smart grid," *IEEE Access*, vol. 7, pp. 35929–35940, 2019.

**NEHA GARG** received the B.Tech. degree in computer science and engineering from Uttar Pradesh Technical University, Lucknow, India, in 2005, and the M.Tech. degree in computer science and engineering, in 2009. She is currently pursuing the Ph.D. degree in computer science and engineering with Graphic Era deemed to be University, Dehradun, India. She has published more than 25 research articles in her research areas. Her research interests include network security, the Internet of Things, and blockchain.

**MOHAMMAD WAZID** (Senior Member, IEEE) received the M.Tech. degree in computer network engineering from Graphic Era deemed to be University, Dehradun, India, and the Ph.D. degree in computer science and engineering from the International Institute of Information Technology, Hyderabad, India. He was an Assistant Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, MAHE, Manipal, India. He was also a Postdoctoral Researcher with the Cyber Security and Networks Laboratory, Innopolis University, Innopolis, Russia. He is currently working as an Associate Professor with the Department of Computer Science and Engineering, Graphic Era deemed to be University. He is also the Head of the Cybersecurity and IoT Research Group, Graphic Era deemed to be University. He has published more than 70 articles in international journals and conferences in the research areas. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SMART GRID, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS (formerly the IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE), the *IEEE Consumer Electronics Magazine*, IEEE ACCESS, *Future Generation Computer Systems* (Elsevier), *Computers & Electrical Engineering* (Elsevier), *Computer Methods and Programs in Biomedicine* (Elsevier), *Security and Communication Networks* (Wiley), and the *Journal of Network and Computer Applications* (Elsevier). His current research interests include information security, remote user authentication, the Internet of Things (IoT), cloud/fog/edge computing, and blockchain. He has also served as a program committee member in many international conferences. He was a recipient of the University Gold Medal and the Young Scientist Award by UCOST, Department of Science and Technology, Government of Uttarakhand, India. He has received Dr. A. P. J. Abdul Kalam Award for his innovative research works and *ICT Express* (Elsevier) Journal Best Reviewer Award for the year of 2019.

**ASHOK KUMAR DAS** (Senior Member, IEEE) received the M.Tech. degree in computer science and data processing, the M.Sc. degree in mathematics, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. He has authored over 220 articles in international journals and conferences in the research areas, including over 190 reputed journal articles. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SMART GRID, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS (formerly the IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE), the *IEEE Consumer Electronics Magazine*, the IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE ACCESS, the *IEEE Communications Magazine*, *Future Generation Computer Systems*, *Computers & Electrical Engineering*, *Computer Methods and Programs in Biomedicine*, *Computer Standards & Interfaces*, *Computer Networks*, *Expert Systems with Applications*, and the *Journal of Network and Computer Applications*. His current research interests include cryptography, network security, blockchain, security in the Internet of Things (IoT), the Internet of Vehicles (IoV), the Internet of Drones (IoD), smart grids, smart city, cloud/fog computing, industrial wireless sensor networks, and intrusion detection. He has served as a program committee member in many international conferences. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He also severed as one of the Technical Program Committee Chairs of the International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, in June 2019. He is on the Editorial Board of *KSII Transactions on Internet and Information Systems*, the *International Journal of Internet Technology and Secured Transactions* (Inderscience), and *IET Communications*. He is a Guest Editor of *Computers & Electrical Engineering* (Elsevier) for the special issue on Big data and IoT in E-healthcare, *ICT Express* (Elsevier) for the special issue on Blockchain Technologies and Applications for 5G Enabled IoT, and *Wireless Communications and Mobile Computing* (Wiley/Hindawi) for the special issue on Attacks, Challenges, and New Designs in Security and Privacy for Smart Mobile Devices.

**DEVESH PRATAP SINGH** received the M.Tech. degree in computer science and engineering from Uttarakhand Technical University, Dehradun, India, in 2009, and the Ph.D. degree, in 2015. He is currently a Professor and the Head of the Computer Science and Engineering Department, Graphic Era deemed to be University, Dehradun, India. He has published more than 50 research articles in his area of expertise. His research interests include information security, wireless sensor networks, the Internet of Things, and soft computing.

**JOEL J. P. C. RODRIGUES** (Fellow, IEEE) is currently a Professor with the Federal University of Piauí, Brazil. He is also a Senior Researcher with the Instituto de Telecomunicações, Portugal. He is the Leader of the Next Generation Networks and Applications Research Group (CNPq). He has authored or coauthored over 850 articles in refereed international journals and conferences, three books, and one ITU-T Recommendation. He holds two patents. He is a member of the Internet Society and a Senior Member ACM. He is a Steering Committee Member of the IEEE Life Sciences Technical Community. He is a Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He had been awarded several Outstanding Leadership and Outstanding Service Awards by the IEEE Communications Society and several best papers awards. He is a Technical Activities Committee Chair with the IEEE ComSoc Latin America Region Board. He is a Past-Chair of the IEEE ComSoc Technical Committee on eHealth and the IEEE ComSoc Technical Committee on Communications Software. He is a Publications Co-Chair. He has been a General Chair and TPC Chair of many international conferences, including the IEEE ICC, the IEEE GLOBECOM, the IEEE HEALTHCOM, and the IEEE LatinCom. He is the President of the Scientific Council with ParkUrbis—Covilhã Science and Technology Park. He is the Editor-in-Chief of the *International Journal on E-Health and Medical Communications*. He is the editorial board member of several high-reputed journals. He is the Director of the Conference Development—the IEEE ComSoc Board of Governors. He is the IEEE Distinguished Lecturer.

**YOUNGHO PARK** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 1989,1991, and 1995, respectively. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, USA. He is currently a Professor with the School of Electronics Engineering, Kyungpook National University. His research interests include computer networks, multimedia, and information security.