IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/.2023.DOI

Design of Robust Blockchain-Envisioned Authenticated Key Management Mechanism for Smart Healthcare Applications

SIDDHANT THAPLIYAL, (Student Member, IEEE)[1], MOHAMMAD WAZID, (Senior Member, IEEE)[2], DEVESH PRATAP SINGH, (Member, IEEE)[3], ASHOK KUMAR DAS, (Senior Member, IEEE)[4], SACHIN SHETTY, (Senior Member, IEEE)[5], ABDULLAH ALQAHTANI [6]

[1]"Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India (e-mail: sthapliyal37@gmail.com)".
[2]"Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India (e-mail: wazidkec2005@gmail.com)".
[3]"Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India (e-mail: devesh.geu@gmail.com)".
[4]"Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India (e-mail: ashok.das@iiit.ac.in, iitkgp.akdas@gmail.com)".
[5]"Virginia Modeling, Analysis and Simulation Center, Department of Computational Modeling and Simulation Engineering, Old Dominion University, Suffolk, VA 23435, USA (e-mail: sshetty@odu.edu)".
[6]"Software Engineering Department, College of Computer Engineering and Sciences Prince Sattam bin Abdulaziz University, P.O. Box 151, Al-Kharj 11942, KSA (e-mail: aq.alqahtani@psau.edu.sa)"

*Corresponding author: Mohammad Wazid (e-mail: wazidkec2005@gmail.com).

This work is supported in part by "DoD Center of Excellence in AI and Machine Learning (CoE-AIML) under Contract Number W911NF-20-2-0277 with the U.S. Army Research Laboratory. The work is also supported in part by funding from Prince Satam bin Abdulaziz University project number (PSAU/2023/R/ 1444)."

ABSTRACT The healthcare sector is a very crucial and important sector of any society, and with the evolution of the various deployed technologies, like the Internet of Things (IoT), machine learning and blockchain it has numerous advantages. However, in this section, the data is much more vulnerable than others, because the data is strictly private and confidential, and it requires a highly secured framework for the transmission of data between entities. In this article, we aim to design a blockchain-envisioned authentication and key management mechanism for the IoMT-based smart healthcare applications (in short, we call it SBAKM-HS). We compare the various attributes of the proposed SBAKM-HS and other existing schemes to demonstrate that SBAKM-HS outperforms other existing schemes. The conducted security analysis and formal security verification via Scyther automated validation tool prove the security of the proposed SBAKM-HS against various possible potential attacks. Next, a real-tested implementation of SBAKM-HS is provided to observe its impact on the performance of the system.

INDEX TERMS Internet of Medical Things (IoMT), Smart healthcare, Blockchain, Authentication, Key Agreement, Security.

I. INTRODUCTION

Smart healthcare is the widespread use of electronic devices, and communication technology over the internet to support and enhance existing healthcare facilities and to develop advanced and automated healthcare systems [1]. Internet of Things (IoT) is the use of electronic devices fitted with sensors to record data from their neighbouring environment. These devices can also be equipped with components such as processors, software, and transmitting/receiving the ability to process and exchange information over the internet or communication channels. IoT devices are used to convert otherwise human-operated systems into automated smart devices, such as smart TVs or Smart Vehicles. Smart Environments such as a smart home or smart office can also be created using multiple IoT devices connected in coordination. One of the sub-branches of IoT is the Internet of Medical Things (IoMT) which deals with IoT devices designed specifically for the collection, processing, and exchange of medical data. IoMT is a communication network consisting of hardware and software components

connected to healthcare IT systems [2]. Smart healthcare makes extensive usage of the Internet of Medical Things environment to monitor patients in real-time and administer necessary treatment. IoMT-based smart healthcare can keep up-to-date records of patient's vitals which are regularly recorded at pre-defined intervals, which would otherwise require a lot of human intervention and is also prone to human error [3], [4], [5]. Smart healthcare systems provide remote access to gather information regarding patients' Vitals (body temperature, SpO2 (Blood Oxygen) Levels, Blood Pressure, Electrocardiogram (ECG), Heart Rate, and Respiration) using IoMT devices (such as Heart Rate sensor (MAX30100), actuators) connected to a personal or cloud server. The data can be accessed through the cloud server by a remote Medical Personnel (e.g., Doctor) making diagnosis easier and effective medicine can be prescribed [6], [7]. Following are the outcomes of a smart healthcare system.

- **Efficient monitoring:** All the patient's vitals can be effectively recorded without the need for human help. Recorded data is also free of human error. Also, Data is periodically recorded and updated on the servers.
- **Remote diagnosis:** Patients Can Consult Doctors from the comfort of their homes, and doctors can easily diagnose and prescribe the patients by analyzing the vitals of the patient.
- **Cost factor:** Such a system will reduce the cost of treatment as patients can freely move whilst wearing smart healthcare devices, thus it is not mandatory to admit the patient for a long time, which further reduces the operational cost of the hospital.
- **Access:** Accessibility of medical services such as diagnosis of a patient in case of emergency will improve since the doctor can monitor the patient's health remotely. Additionally, they can also suggest treatment without any delay.
- **Improve patient involvement:** Smart healthcare provides the consumer with more access to their medical records, and improves their overall involvement in their medical care, which in turn would improve their overall satisfaction and healthcare experience.

The patient's information is gathered, processed, and stored with the aid of an IoMT-based smart healthcare system using a personal server as an intermediary node. This system enables the personal server to gain access to all of the user's data, which can then be sent to the cloud server for processing and secure storage. The patient's data may also be accessed by external users (such as a doctor or the patient's attendant) via a cloud server.

Health-related information, however, is particularly sensitive by nature and should only be kept and shared using secure methods. We require a secure authentication/ access control mechanism between the user's personal server and smart healthcare device for the system to be secured. The exchange of data between the cloud server and distant devices (i.e., smartphones of users) should likewise follow a similar authentication/ access control policy. To prevent data alterations or tampering, a blockchain-based mechanism should also be set up on the cloud server of the peer-to-peer cloud server network.

Blockchain is the modern data storage and exchange technology, which is tamper-proof, and makes it difficult to alter, update or hack the data stored on it. Blockchain is a distributed ledger technology, in which data (in the form of certain transactions) is duplicated and stored among thousands of servers spread over the globe, thus forming a "peer-to-peer" network. Here each peer or node is of equal value, eliminating the need for a master or server node thus holding the integrity of data. A blockchain network also allows each node to access the information uploaded by any node, thus maintaining the availability of data. To alter the data in a blockchain 51% of the peers need to simultaneously vote in favour of the update, which is exceedingly difficult to achieve, thus making the blockchain practically tamper-proof [8]. In a smart healthcare system, most of the communication or data exchange is wireless in nature. IoMT devices fitted to the patient's body communicate with the patient's personal server wirelessly (using Wi-Fi or similar technology). With IoMT-based smart healthcare systems, such data transfer is susceptible to attacks like man-in-the-middle (MiTM), privileged insider, replay, etc [8].

Furthermore, impersonation or masquerading attacks can allow malicious users to pose as the personal server and directly connect to the body sensors, while in close proximity to the patient. This can cause sensitive information related to the patient's health to be revealed to someone that can cause harm to the social and personal life of the patient. The Healthcare data of a patient is overly sensitive and should not be modified or tampered with in any case. To achieve secure transmission of data and prevent attacks, an access control mechanism or protocol should be deployed that will authenticate and verify the participating parties (for instance, personal server and healthcare device) in the communication channel and will only begin the transmission in case of successful authentication. From the above discussion, it is noticeably clear that to implement an IoMT-based smart healthcare system, we need to also implement a "secure authentication and key management mechanism" to store and exchange important healthcare data.

A. MOTIVATION

Our daily healthcare supports and usage demand the use of an intelligent healthcare system. Unfortunately, it is vulnerable to a variety of security and privacy issues, such as "replay attacks, denial of service attacks, malware injection, physical smart healthcare device theft attacks, man-in-the-middle attacks, impersonation attacks, the unauthorised session key computation, stolen verifier attacks, and others". Moreover, conventional security methods such as authentication, key formation, and access control are not very safe and may fail as a result of possible attacks. [9] [10] [11] For which, we require strong security procedures to fight

against these possible risks and assaults. In this paper, we have presented a blockchain-envisioned mechanism, which provides better security and extra functionality features than the various existing schemes.

B. RESEARCH CONTRIBUTIONS

The following lists the research contributions made by the proposed SBAKM-HS.

- We propose a blockchain-envisioned authentication and key management mechanism for IoMT-based smart healthcare applications (in short SBAKM-HS).
- We compare the various attributes of the proposed SBAKM-HS and other existing schemes. SBAKM-HS outperforms other existing schemes.
- The formal security verification (using the Scyther tool) and performed security analysis demonstrate the security of the proposed SBAKM-HS against a variety of potential threats/ attacks.
- The testbed implementation of SBAKM-HS is offered as a final step so that system performance can be evaluated in real-time.

II. LITERATURE REVIEW

Researchers are already working on the same concept and approach, as many of them already had presented their research articles [12]–[16]. Xiao *et al.* [17] presented intelligent UAV-based monitoring systems, which were turning into a crucial tool for crowd surveillance. Uses of such systems might include recognising strange or hostile behaviour in a crowd to preserve the security and safety of the public, especially during epidemics or periods of social upheaval when technology was designed to replace the human element in order to assure scalability and reduce danger. Therefore, a safe architecture must be achieved using an efficient technique that takes into account the system's dispersed nature as well as the UAV agents' limited processing power. A secure, open, and effective network infrastructure for UAV systems will be provided by a blockchain, a distributed network technology. As a result, they suggested a distributed monitoring system that uses drone swarms in a network that runs on blockchain technology. In order to achieve the cooperative drone swarm's ability to consistently carry out monitoring tasks, the security protocol and encryption algorithm was implemented in the suggested monitoring mechanism.

The introduction of blockchain technology enables tamper-proof monitoring log recording and supports collective monitoring decision-making. Ferrag *et al.* [1] presented that IoT security and privacy solutions based on blockchain. They have started it by outlining the previous studies that address blockchain security for Internet of Things networks. Then, they looked into blockchain-based security and privacy solutions for seventeen various varieties of IoT applications, including "Industry 4.0, software-defined networking, edge computing, the Internet of Drones, the Internet of Cloud, the Internet of Energy, the Internet of Vehicles, etc." Also, they contrasted the nine characteristics–latency, throughput, processing, storage, and communication costs; scalability; attack model; benefit; and disadvantage–of the different consensus approaches. In addition, they performed security analysis techniques and divided them into four groups, including the "AVISPA tool, game theory, theory analysis, and Burrows, Abadi, and Needham (BAN) logic." On the basis of the findings, the critical steps to follow while creating and evaluating blockchain-based security and privacy strategies were also provided.

Lu *et al.* [11] presented a scheme for smart factory systems, driven through the Industrial Internet of Things (IIoT). They applied IIoT technologies to industrial contexts. It made use of a variety of sensors to gather data from industrial machines. In order to increase industrial productivity and product quality, the data were examined. Data outsourcing could be solved by using cloud storage, particularly for sensors with constrained local storage and processing power. The acquired data should be kept in a formal cipher text in order to guarantee that the devices' privacy was preserved. The cloud storage option for sensors was taken into consideration in their article.

Paul *et al.* [18] discussed that The Internet of Things (IoT) was exploding in both academia and business. IoT had many common and customary security solutions, yet it still faced several privacy and security issues. Because of the dispersed architecture and resource limitations, the bulk of IoT devices were not suited for standard security protocols. Blockchain was most effectively used to maintain the five fundamental cryptographic primitives of secrecy, authenticity, integrity, availability, and non-repudiation. When standard blockchain was used in IoT security, it results in excessive energy consumption, delays, and computational overheads, which were inappropriate for diverse IoT devices with limited resources. For the purpose of maintaining all cryptographic security and privacy concerns, the suggested IoT-based smart city design used blockchain technology. Their scheme had very little overhead. Their study looked at current threat models and important access control issues that dealt with a variety of permissions on various processing nodes in order to find relevant inconsistencies. As compared to current literature in terms of all security concerns, their suggested architecture was comparably efficient. This study's main objective was to investigate if blockchain might serve as a substitute for traditional security measures in low-resource IoT applications.

Yang *et al.* [19] discussed that IoT has gained significant attraction over the past ten years in a number of industries, including education, business, government, and healthcare. One of the difficult problems associated with the growing number of connected devices in the IoT system is ensuring device authenticity so that users may make decisions with high trust. The heterogeneity of the IoT system and the resource-constrained devices raised additional concerns about how to govern such a system and guarantee security and privacy for devices. We proposed a new "blockchain-

driven authentication and key management technique" for IoMT-based smart healthcare systems to overcome the problems.

III. SYSTEM MODELS

The required system models of the proposed SBAKM-HS are given below.

A. NETWORK MODEL

Fig. 1 depicts the network model of the proposed SBAKM-HS. In the given scenario, we have linked smart healthcare devices to a patient, a personal server, and a number of users, like doctors and nursing staffs. Smart healthcare devices examine and monitor individuals' health before uploading data to a personal server for its processing and forwarding to the cloud server. Therefore, the data is transferred to the associated cloud servers for further usage. The cloud server's function in this circumstance is critical since it processes and stores healthcare data. This cloud data might then be utilized to provide beneficial outcomes through some machine learning algorithms (for example, prediction about a patient's illness). Various types of users (i.e., doctors) want to access healthcare data. As a result, for safe exchange of data among smart healthcare devices and personal servers, as well as among personal servers and cloud servers, secure mutual authentication and session key establishment are essentially needed. In a similar way, it is also required among the cloud servers and users. Using this technique, the entities, namely smart healthcare devices and personal servers, mutually authenticate and establish session keys for safe data transmission. This paper provides a secure mutual authentication and key establishment procedure between the legitimate smart healthcare devices and their respective personal server(s). Apart from that, there is a separate procedure for key management between the legitimate personal server and the cloud server to assure their secure data transmission. Moreover, a trusted authority, TA, which is also the system's trusted entity, registers other entities (i.e., devices, servers, and users) and provides them with the secret credentials so that the "mutual authentication and key establishment" phase can be proceed as planned.

B. THREAT MODEL

In the proposed SBAKM-HS, we take into consideration the following threat and adversary models. The Dolev-Yao (DY) threat model, which is rather popular, has been used as the basis for the design of the SBAKM-HS [20]. According to the DY model, two interacting entities are required to connect with one another through a path that is not secure, specifically the Internet. In general, end-point entities, such as smart healthcare devices and personal servers, are not to be fully trusted. Hence, messages that are sent across a public channel that is not secure are susceptible to being snooped on, modified, or even deleted by an adversary who is either active or passive, such as an attacker $\mathcal{A}$. In addition, when we construct an authentication and key establishment

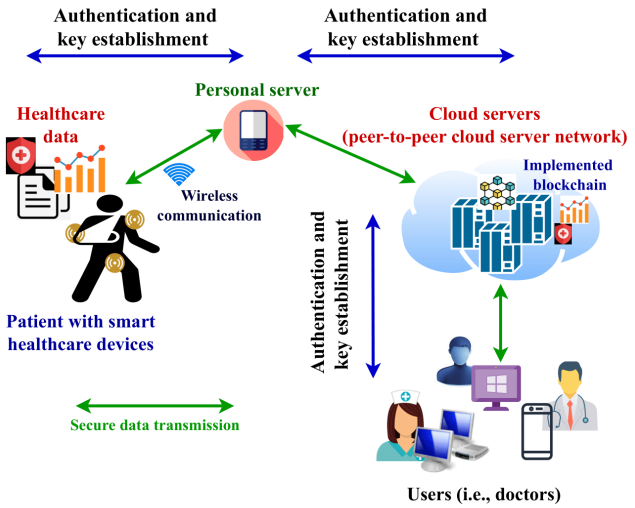

FIGURE 1: Network model of the proposed SBAKM-HS

protocol, we adhere to the adversary model developed by Canetti and Krawczyk (CK), which is the model that is currently considered to be the de-facto standard [10] as compared to the DY model. In the CK adversary model, $\mathcal{A}$ can have all of the powers that are available in the DY model. In addition, the adversary $\mathcal{A}$ can get the session states, also known as the credentials associated with the session, and the session keys for a particular session. $\mathcal{A}$ can physically capture certain smart healthcare devices and steal the information that is stored in their memory by applying the methods of a sophisticated power analysis attack. The information that is gathered could also be used to carry out malicious operations, such as calculating secret credentials and session keys, as well as to launch other attacks, such as impersonating smart healthcare devices, replaying data, and launching privileged-insider and man-in-the-middle (MiTM) attacks. These attacks could be launched by using the information that is gathered. The personal servers are deployed under some physical security to thwart the attempts of physical personal server stolen attack [21]. In addition, it was anticipated that the network's trusted registration authority TA was a secure entity that could not be compromised in any way. Despite the fact that cloud servers are considered to be partially trusted entities within the network [9].

IV. DETAILS OF THE PROPOSED SBAKM-HS

We describe the processes of the proposed SBAKM-HS in this section. It is broken down into multiple stages, including the phases of "registration, key establishment, authentication, dynamic device addition, and blockchain implementation." Table 1 lists the specifics of the notations used in SBAKM-HS. The following parts of this section provide more detail on the different SBAKM-HS phases.

TABLE 1: Notations used in the proposed SBAKM-HS

Notation	Meaning
$\mathcal{A}$	An adversary
HD_i, ID_{HD_i}, RID_{HD_i}	i^{th} smart healthcare device, its identity and pseudo-identity, respectively
PS_j, ID_{PS_j}, RID_{PS_j}	j^{th} personal server, its identity and pseudo-identity, respectively
CS_k, ID_{CS_k}, RID_{CS_k}	k^{th} cloud server, its identity and pseudo-identity, respectively
TA, k_{TA}	Trusted registration authority, its secret key and its pseudo-identity, respectively
k_{HD_i}, k_{PS_j}	Private keys HD_i and PS_j
MK_{HD_i}	Master secret key of HD_i
K_{HD_i,PS_j}	A unique (distinct) random shared secret key of HD_i with PS_j
RTS_{HD_i} & RTS_{PS_j}	Registration timestamp values of HD_i and PS_j
TC_{HD_i} & TC_{PS_j}	Temporal credential values of HD_i and PS_j
TS_x	Various used timestamp values
rs_x	various used random secret values
$\mu(x, y)$	A unique symmetric bivariate polynomial parameter
PIN_{PS_j} & PIN_{CS_k}	The pseudo identification numbers of PS_j and CS_k
ΔT	The transmission delay permitted
$h(\cdot)$	A cryptographic operation via one-way hash function
SK_{x_i,y_j}	Establishment of session key in between parties x_i and y_j
$\|\|$	Operation via concatenation
$\oplus$	Operation via bitwise exclusive-OR (XOR)

A. PRE-DEPLOYMENT PHASE

This phase permits the TA to fulfill the registration of sensors and servers before they are installed in the network.

B. REGISTRATION PHASE

In this stage, smart healthcare devices, personal and cloud servers are registered. The dependable registration authority TA completes all entity registrations. The information is provided below.

1) Registration of Smart Healthcare Device HD_i

Following the steps below, TA registers smart healthcare device HD_i.

- **RHD1:** For the registration of HD_i, TA does the generation of identity of HD_i as ID_{HD_i}. It then generates a temporary identity of HD_i as TID_{HD_i} and a secret key of itself as K_{TA}. TA also does the creation of secret key of HD_i as k_{HD_i} and its master secret key as $MK_{HD_i} = h(ID_{HD_i}|| \ k_{HD_i})$ and pseudo identity as $RID_{HD_i} = h(ID_{HD_i}|| \ k_{TA})$. Again, TA computes temporal credential of HD_i as $TC_{HD_i} = h(ID_{HD_i}|| \ k_{HD_i}|| \ k_{TA}|| \ RTS_{HD_i})$, where RTS_{HD_i} is taken as the registration timestamp value corresponding to HD_i. TA again generates a unique (distinct) random shared secret key of HD_i as K_{HD_i,PS_j} with the respective registered personal server PS_j.
- **RHD2:** Finally, TA stores the registration values like, $\{TID_{HD_i}, \ RID_{HD_i}, \ K_{HD_i,PS_j}, \ TC_{HD_i}, h(\cdot), MK_{HD_i}\}$ in the memory of HD_i.

2) Registration of Personal Server PS_j

TA does the registration of personal server PS_j as per the following steps.

- **RPS1:** The TA generates the identity of PS_j as ID_{PS_j} and its secret key as k_{PS_j}. It then computes the pseudo identity of PS_j as $RID_{PS_j} = h(ID_{PS_j}|| \ k_{TA})$, its temporal credentials as $TC_{PS_j} = h(ID_{PS_i}|| \ k_{PS_j}|| \ k_{TA}|| \ RTS_{PS_j})$, where RTS_{PS_j} is taken as the registration timestamp corresponding to PS_j. After that, TA also generates a pseudo identification number of PS_j as $PIN_{PS_j} = h(ID_{PS_j}|| \ k_{TA}|| \ k_{PS_j})$ then TA selects a "unique symmetric bivariate polynomial parameter $\mu(x, y) = \sum_{m,n=0}^{t} a_{m,n} x^m y^n \in GF(p)[x, y]$ of degree t over a finite field (Galois field) $GF(p)$ $(= Z_p)$, where the coefficients $a_{i,j}$'s are selected 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 number of personal servers to be deployed." For instance, a "bivariate polynomial $\mu(x, y) = x^4 + 3x^3 + 2x^2y^2 + 3y^3 + y^4$ over $GF(5)$ is symmetric as $\mu(y, x) = y^4 + 3y^3 + 2y^2x^2 + 3x^3 + x^4 = \mu(x, y)$." Furthermore, TA calculates a polynomial share $\mu(PIN_{PS_j}, y) = \sum_{m,n=0}^{t} [a_{m,n}(PIN_{PS_j})^m] y^n$, which is clearly a univariate polynomial of the same degree t.
- **RPS2:** Finally, TA stores the registration values like, $\{\{(TID_{HD_i}, \ RID_{HD_i}, \ K_{HD_i,PS_j}), i = 1, \ 2, \cdots, n_{HD}\}, \ RID_{PS_j} \ TC_{PS_j}, PIN_{PS_j}, \mu(PIN_{PS_j}, y), h(\cdot)\}$ in the secured region of database of PS_j.

3) Registration of Cloud Server CS_k

TA does the registration of cloud server CS_k as per the following steps.

- **RCS1:** The TA generates the identity of CS_k as ID_{CS_k} and its secret key as k_{CS_k}. After that, TA also generates a pseudo identification number of CS_k as $PIN_{CS_k} = h(ID_{CS_k}|| \ k_{TA}|| \ k_{CS_k})$ then TA selects a "unique symmetric bivariate polynomial parameter" as stated earlier. Furthermore, TA calculates a "polynomial share $\mu(PIN_{CS_k}, y) = \sum_{m,n=0}^{t} [a_{m,n}(PIN_{CS_k})^m] y^n$, which is clearly a univariate polynomial of the same degree t."
- **RPS2:** Finally, TA stores the registration values like, $\{PIN_{CS_k} \ \mu(PIN_{CS_k}, y), h(\cdot)\}$ in the secured region of database of CS_k.

Remark 1. *The TA deletes all secret values, such as K_{HD_i,PS_j}, MK_{HD_i}, k_{HD_i}, k_{PS_j}, RTS_{HD_i}, RTS_{PS_j}, RID_{HD_i}, RID_{PS_j}, PIN_{PS_j}, PIN_{CS_k}, $\mu(PIN_{PS_j}, y)$, and $\mu(PIN_{CS_k}, y)$ from its memory in order to thwart attempts of stolen verifier attacks and other potential attacks. The process for adding dynamic devices uses a similar methodology. Additionally, all registration values for HD_i and PS_j are kept in the secured region of the database of PS_j. Moreover, PS_j is also under a physical locking system to mitigate the attempts of physical stolen attack of PS_j [21].*

C. AUTHENTICATION AND KEY AGREEMENT PHASE BETWEEN HD_I AND PS_J

The authentication and key establishment between HD_i and PS_j must take place during this step. The information is

provided below.

- **AKAH1:** HD_i generates the new timestamp value TS_1, random secret value rs_1 to start the procedure. After that, parameters such as $M_1 = h(RID_{HD_i}||\ ||K_{HD_i,PS_j}\ ||TS_1) \oplus h(rs_1||\ TC_{HD_i}||\ MK_{HD_i})$, $M_2 = h(h(rs_1||\ MK_{HD_i}||TC_{HD_i})||\ TS_1)$ are computed. Then HD_i sends message $msg_1 = \{TID_{HD_i},\ M_1,\ M_2, TS_1\}$ to PS_j through the open channel.
- **AKAH2:** At the arrival of msg_1, PS_j verifies the timeliness of TS_1 via condition $|TS_1 - TS_1^*| \leq \Delta T$, where TS_1 is the standard receiving time of msg_1 and TS_1^* is the time when msg_1 is actually received. If the verification of TS_1 happens successfully, then PS_j fetches RID_{HD_i}, and K_{HD_i,PS_j} corresponding to the received TID_{HD_i} from its database. After that PS_j computes $h(rs_1||\ MK_{HD_i}||TC_{HD_i}) = M_1 \oplus h(RID_{HD_i}||K_{HD_i,PS_j}||TS_1)$ and $M_2' = h(h(rs_1||\ MK_{HD_i}||TC_{HD_i})||\ TS_1)$. It then checks $M_2' = M_2$? If the condition holds then HD_i is authenticated with PS_j; Otherwise, it aborts the session with HD_i. After that PS_j generates a fresh timestamp value TS_2 and a random secret value rs_2. Further PS_j computes $M_3 = h(rs_2||\ RID_{PS_j}||\ TC_{PS_j}) \oplus h(RID_{HD_i}||\ K_{HD_i,PS_j}||TS_1||\ TS_2)$. It again computes the session key as $SK_{PS_j,HD_i} = h(h(rs_2||\ RID_{PS_j}||\ TC_{PS_j})||\ h(rs_1||\ MK_{HD_i}||\ TC_{HD_i})||\ RID_{HD_i}||\ K_{HD_i,PS_j}||TS_1||\ TS_2)$ and $M_4 = h(SK_{PS_j,HD_i}||\ RID_{HD_i}||\ TS_1||TS_2)$. After that PS_j generates a new temporary identity of HD_i as $TID_{HD_i}^{new}$ and computes $M_5 = TID_{HD_i}^{new} \oplus h(RID_{HD_i}||\ h(rs_2||\ RID_{PS_j}||\ TC_{PS_j})||\ TS_1||\ TS_2)$. It then sends message $msg_2 = \{M_3,\ M_4,\ M_5,\ TS_2\}$ to HD_i through the open channel.
- **AKAH3:** At the arrival of msg_2, HD_i verifies the timeliness of TS_2 as per the condition discussed earlier. If that happens successfully, it then computes $h(rs_2||\ RID_{PS_j}||\ TC_{PS_j}) = M_3 \oplus h(RID_{HD_i}||\ K_{HD_i,PS_j}||\ TS_1||\ TS_2)$. It again computes the session key as $SK_{HD_i,PS_j} = h(h(rs_2||\ RID_{PS_j}||\ TC_{PS_j})||\ h(rs_1||\ MK_{HD_i}||\ TC_{HD_i})||\ RID_{HD_i}||\ K_{HD_i,PS_j}||TS_1||\ TS_2)$ and $M_4' = h(SK_{HD_i,PS_j}||\ RID_{HD_i}||\ TS_1||TS_2)$. After that, it checks the condition $M_4' = M_4$? If it holds then PS_j is authenticated with HD_i and the session key computed by HD_i is correct. HD_i also computes its new temporary identity as $TID_{HD_i}^{new} = M_5 \oplus h(RID_{HD_i}||\ h(rs_2||\ RID_{PS_j}||\ TC_{PS_j})||\ TS_1||\ TS_2)$. HD_i computes another message for session key verification, which is done by PS_j. Here HD_i generates another TS_3 and computes $M_6 = h(SK_{HD_i,PS_j}||\ TS_3||\ TID_{HD_i}^{new})$. It then sends message $msg_3 = \{M_6,\ TS_3\}$.
- **AKAH4:** At the arrival of msg_3, PS_j verifies the timeliness of TS_3 as per the condition discussed earlier. If that happens successfully, it computes $M_6' = h(SK_{PS_j,HD_i})||\ TS_3||\ TID_{HD_i}^{new})$. Then PS_j checks $M_6' = M_6$? If it holds, PS_j assumes that the session key computed by HD_i is correct and HD_i has successfully updated its new temporary identity. Then both HD_i and PS_j establish session key $SK_{HD_i,PS_j}(= SK_{PS_j,HD_i})$ for the secure transmission of their information.

D. DYNAMIC DEVICE ADDITION PHASE

This phase is required to add a new device to the network when there is some need for that deployment. As some of the devices may malfunction. Sometimes, there is a need for the deployment of more devices in the network. The details of dynamic device addition are given below.

- **DDAH1:** First of all TA generates the identity of the new device HD_i^ν as $ID_{HD_i}^\nu$ along with its temporary identity as $TID_{HD_i}^\nu$. TA again generates its secret key as $k_{HD_i}^\nu$. TA then computes its master key as $MK_{HD_i}^\nu = h(ID_{HD_i}^\nu||\ k_{HD_i}^\nu)$, pseudo-identity as $RID_{HD_i}^\nu = h(ID_{HD_i}^\nu||\ k_{TA})$ and temporal credentials as $TC_{HD_i}^\nu = h(ID_{HD_i}^\nu||\ k_{TA}||\ RTS_{HD_i}^\nu)$. TA again generates a unique (distinct) random shared secret key of HD_i^ν as K_{HD_i,PS_j}^ν with the respective registered personal server PS_j.
- **DDAH2:** Finally, TA stores the registration values like, $\{TID_{HD_i}^\nu,\ RID_{HD_i}^\nu,\ K_{HD_i,PS_j}^\nu,\ TC_{HD_i}^\nu, h(\cdot), MK_{HD_i}^\nu\}$ in the memory of HD_i^ν. Then HD_i^ν can be deployed in the required region. TA also informs about the addition of HD_i^ν to PS_j via some secure mechanism.

E. KEY MANAGEMENT BETWEEN PS_J AND CS_K

This phase is required to perform the key management between PS_j and CS_k. After their key establishment, they can exchange their data in a secure way. For the secure key management between PS_j and CS_k, we can follow the approach of Blundo *et al.* [22]. The following is an explanation of the reasoning behind the utilization of the scheme developed by Blundo *et al.* [22] for the establishment of pairwise keys between PS_j and CS_k. If an adversary $\mathcal{A}$ is successful in compromising $(t+1)$ or more shares of $\mu(x, y)$, then that $\mathcal{A}$ will be in a position to quickly recreate the original, unique $\mu(x, y)$ using Lagrange's interpolation. Therefore, the exposure of up to t shares does not divulge $\mu(x, y)$ to $\mathcal{A}$, and as a result, non-compromised shared keys that are based on $\mu(x, y)$ continue to maintain their full level of security. Due to the fact that the degree t of $\mu(x, y)$ is significantly more than the total number of personal servers deployed [23]. Therefore, the proposed scheme maintains unconditional security and t collusion-resistant property. Suppose PS_j and CS_k want to establish a pairwise secret key for their secure communication. The procedure is similar to that in Blundo *et al.*'s scheme [22].

- **KMPC1:** PS_j first initiates the process by sending its PIN_{PS_j}, a generated random nonce r_1 and its current timestamp TS_{ps} to CS_k. Similarly, CS_k also sends

PIN_{CS_k}, a generated random nonce r_2 and its current timestamp TS_{cs} to PS_j.

- **KMPC2:** After that PS_j checks the validity of timestamp TS_{ps} and it is valid, PS_j computes the secret key shared with CS_k using its own polynomial share as $SK_{PS_j,CS_k} = h(\mu(PIN_{PS_j}, PIN_{CS_k})|| \ r_1||r_2 \ ||TS_{ps}||TS_{cs})$. Similarly, CS_k checks the validity of timestamp TS_{cs} and it is valid, CS_k computes the same secret key shared with PS_j using its polynomial share as $SK_{CS_k,PS_j} = h(\mu(PIN_{CS_k}, PIN_{PS_j})|| \ r_1||r_2 \ ||TS_{ps}||TS_{cs})$, which is same as SK_{PS_j,CS_k} because $\mu(PIN_{PS_j}, PIN_{CS_k}) = \mu(PIN_{CS_k}, PIN_{PS_j})$ due to symmetry property of the bivariate polynomial $\mu(x, y)$.
- **KMPC3:** Both PS_j and CS_k can now communicate securely through this estimated and established secret key, i.e., $SK_{PS_j,CS_k} = (SK_{CS_k,PS_j})$.

F. BLOCKCHAIN IMPLEMENTATION PHASE

In this phase, we provide the details of the blockchain implementation phase of the SBAKM-HS. This phase is elaborated as follows:

- **BIP1:** The blockchain of the healthcare data is implemented at the peer-to-peer (P2P) cloud server network. The smart healthcare devices, say HD_i monitoring of the health of the patients and send the data to the connected personal server PS_j. PS_j creates the partial block from the received healthcare. The partial block contains information, like, the owner's identity OW_{ID}, the owner's public key OW_{KU} and encrypted transactions. The encrypted transactions are obtained by converting the healthcare data in the form of some transactions and then performing encryption on them through OW_{KU}. For example, we can have $TX = E_{OW_{KU}}(T_x|x = 1, 2, \cdots n_x)$, where n_x are total number of transactions. Further, we get the structure of partial block as $PB_i = \{OW_{ID}, OW_{KU}, TX\}$. Then PS_j generates a fresh timestamp value as t_1 and sends the following message $M_{BK_1} = \{PB_i, t_1\}$ to the connected cloud server with the help of the established session key SK_{PS_j,CS_k} in a secure way.
- **BIP2:** At the Arrival of M_{BK_1}, CS_k performs the verification of timestamp value t_1 as per the condition discussed earlier. If that happens successfully, then CS_k creates the full block FB_i from the received partial block PB_i. The full block contains the values like block's ID BID_{FB_i}, random nonce value RN_{FB_i}, fresh timestamp value TS_{FB_i}, hash of previous block $H_{FB_{i-1}}$, hash of this block H_{FB_i}, Merkle tree root MTR_{FB_i}, OW_{ID}, OW_{KU}, TX and signature of this block SG_{FB_i}. Further, we get the structure of full block as $FB_i = \{BID_{FB_i}, RN_{FB_i}, TS_{FB_i}, H_{FB_{i-1}}, H_{FB_i}, MTR_{FB_i}, OW_{ID}, OW_{KU}, TX, SG_{FB_i}\}$. Then CS_k broadcasts FB_i and a puzzle PZ_i to the authorised minor nodes (cloud servers) of the peer-to-peer cloud server network in a secure way to execute the required consensus process. For the execution of the consensus process, the steps of Practical Byzantine Fault Tolerance (pBFT) can be utilized.
- **BIP2:** The minor nodes verify the genuineness of the received FB_i with the help of the signature verification, which is available (i.e., SG_{FB_i} in the received block. If the verification happens successfully then minor nodes submit the solution of the puzzle PZ_i to the CS_k in a secure way. If a fraction of minor nodes (say 75% yes) agree on the addition of the block FB_i then FB_i is added to the blockchain. Otherwise, the addition of FB_i is aborted and the consensus process is started again.

V. SECURITY ANALYSIS OF THE PROPOSED SBAKM-HS

In this section, we provide the details of the conducted security analysis of the proposed SBAKM-HS. From the conducted security analysis, it has been observed that the SBAKM-HS has the potential to defend the potential attacks of the domain. The details are given below.

Proposition 1. *SBAKM-HS is secured against the replay attack.*

Proof. With the proposed SBAKM-HS, we used the timestamp values TS_1, TS_2, and TS_3 in all of the exchanged messages. The recipient's end also verifies these timestamp values. It is presumed that the received message is new and has not been replayed if the timestamp values can be verified correctly. Thus, SBAKM-HS is able to defend the by replying to messages-related attacks. □

Proposition 2. *SBAKM-HS is secured against man-in-the-middle (MiTM) and impersonation attacks.*

Proof. In proposed SBAKM-HS, we used various timestamps, such as TS_1, TS_2 and TS_3 random secret values, such as rs_1 and rs_2, secret keys, such as k_{HD_i} and k_{PS_j}, and registration timestamps, such as RTS_{HD_i} and RTS_{PS_j}. Because of the use of these values, an attacker $\mathcal{A}$ will find it very challenging to change the values of messages msg_1, msg_2 and msg_3. In addition, $\mathcal{A}$ is unable to generate the right messages by itself. In these conditions, $\mathcal{A}$ loses the ability to carry out MiTM and impersonation attacks on proposed SBAKM-HS. Man-in-the-middle (MiTM) and impersonation attacks are thus protected against by SBAKM-HS. □

Proposition 3. *SBAKM-HS provides anonymity and untraceability properties.*

Proof. Under the proposed SBAKM-HS, none of the identification is transmitted in plaintext. Thus, all personal servers' and devices' identities are hidden. Newly created variables, such as new timestamp values and random secret values, are also used in the calculation of all transmitted and received messages. As a result, the transmitted messages change in each session. Hence, it is impossible for a

potential $\mathcal{A}$ to trace the messages that have been delivered. Further, $\mathcal{A}$ is not able to find out who communicates to whom as the identification information is hidden. Thus the proposed SBAKM-HS offers characteristics that enable device anonymity and untraceability properties. □

Proposition 4. *SBAKM-HS protects against ephemeral secret leakage (ESL) attacks under the CK-adversary model.*

Proof. Generally speaking, assessing the security of an access control or authentication and key establishment technique is a good idea using the CK-adversary model's guiding principles. All of the attributes of the DY model are present to the attacker in this model. In contrast, $\mathcal{A}$ might be able to steal session states. It implies that $\mathcal{A}$ can determine the session key if a freshly developed technique handles it improperly. It is advised that the session keys be computed using both "short-term secrets (such as timestamps and random nonce values) and long-term secrets (like secret keys and many identities)." In SBAKM-HS, session key between HD_i and PS_j is calculated as $SK_{HD_i,PS_j} = h(h(rs_2||\ RID_{PS_j}||\ TC_{PS_j})||\ h(rs_1||\ MK_{HD_i}||\ TC_{HD_i})||\ RID_{HD_i}||\ K_{HD_i,PS_j}||TS_1||\ TS_2)$ It contains both "short term secrets (like, timestamp TS_1, TS_2, TS_3, random secret values rs_1, rs_2)" and "long term secrets (like, secret keys K_{HD_i,PS_j}, k_{HD_i}, k_{TA}, k_{PS_j} and RID_{HD_i}, RID_{PS_j})". It's also important to note that each session starts with the computation and establishment of a new session key. Attacker $\mathcal{A}$ cannot produce the appropriate session key because he or she is unaware of the long and short-term secrets required for precise session key computation. Consequently, it can be stated that the proposed SBAKM-HS is guarded against unauthorized attempts to compute session keys in accordance with the CK-adversary model. □

Proposition 5. *SBAKM-HS is secured against privileged insider attack.*

Proof. After an entity is successfully registered in the SBAKM-HS, the secret registration information (like, k_{HD_i}, RTS_{HD_i}) is deleted from the TA's memory. The insider TA user is, therefore, not aware of these hidden values. The entity's confidential information may be used by an insider user of TA who is acting maliciously to carry out damaging attacks like credential guesswork, MiTM, identity fraud, and unlawful session key calculation. However, these personal settings in SBAKM-HS are not accessible to the insider TA user. As a result, the proposed SBAKM-HS is safeguarded against privileged insider attack. □

Proposition 6. *SBAKM-HS has the ability to mitigate the physical smart healthcare device capture attack.*

Proof. If a security approach is the target of a smart healthcare device capture attack, then the security of the communication will be affected. If session keys or other sensitive data about the devices or users are made public, it is not good from a security perspective. We, therefore, need a method to lessen this onslaught. In SBAKM-HS, the memory of the smart healthcare device does not save the secret data in an unencrypted form. Additionally, if $\mathcal{A}$ manages to take control of a smart healthcare device and utilises an advanced power analysis attack to retrieve data from its memory [24]. In this case, $\mathcal{A}$ is able to obtain the smart healthcare device's session key and not those of the other devices. Each session key is distinct because they are created using a variety of secret and random values. The session keys of other devices cannot be obtained by obtaining just this session key. The remainder of the communication is therefore still safe and secure. Thus, SBAKM-HS is protected from physical attacks that try to capture smart healthcare devices. As a result, SBAKM-HS is safeguarded against the "physical smart healthcare device capture attack."

□

Proposition 7. *SBAKM-HS has ability to defend the stolen verifier attack.*

Proof. In SBAKM-HS, the cloud server's database contains a secure section where all confidential information is kept. It is important to mention that such databases are enabled as "multi-region, multi-master replication and offer extensive data governance with several levels of security, including network isolation and end-to-end encryption [25], [26]." The aforementioned situations prevent attackers from using secret parameters to launch subsequent attacks like "MITM, impersonation, unauthorised computation of session keys," and so on. As a result, $\mathcal{A}$ on SBAKM-HS is safeguarded against the stolen verifier attack. □

VI. FORMAL SECURITY OF THE PROPOSED SBAKM-HS USING ROR MODEL

In this section, we will begin by discussing the Real-Or-Random (ROR) model [27], and we will then proceed to discuss the session key security provided by SBAKM-HS in Theorem 1.

A. ROR MODEL

The formal security proof of SBAKM-HS is carried out in accordance with the ROR model [27]. Participants in SBAKM-HS include the smart healthcare devices HD_i, personal servers PS_j, cloud servers CS_k, and the TA.

Participants. $\Pi^t_{HD_i}$, $\Pi^u_{PS_j}$, $\Pi^v_{CS_k}$ and Π^w_{TA} are for representing instances t, u, v and w of HD_i, PS_j, CS_k and TA, respectively. These are also considered as random oracles [28].

Accepted state. If, after getting the very last expected protocol message, an instance Π^t enters what's known as an accept state, then that instance is said to be in an accepted state. The session identity (sid) is understood to be the ordered concatenation of the exchanged messages, which include both the sent and received messages of an instance Π^t for the session that is presently running.

Partnering. Assume that Π^{t_1} and Π^{t_2} are two different examples. They are considered to be partners if all of the following conditions are met at the same time: 1) Both Π^{t_1} and Π^{t_2} are in an accept state; 2) Both Π^{t_1} and Π^{t_2} mutually authenticate each other, and both are assigned the same sid; and 3) Both Π^{t_1} and Π^{t_2} are partners of the other.

Freshness. $\Pi^{t}_{HD_i}$ or $\Pi^{v}_{PS_j}$ are thought to be in a fresh state if the created session key SK_{ij} between HD_i and PS_j has not been revealed to an opponent $\mathcal{A}$ with the assistance of the reveal query $(Reveal)$ that is defined further down in this paragraph [28], [21].

Adversary. According to the DY model, which is explained in the threat model, $\mathcal{A}$ has complete command over all of the network's communication channels. Therefore, $\mathcal{A}$ is able to not only listen in on conversations but also alter, invent, delete, and introduce new communications into the network. Additionally, in accordance with the ROR, $\mathcal{A}$ will have access to the queries listed below:

- $Execute(\Pi^t, \Pi^u, \Pi^v, \Pi^w)$: After running this query, $\mathcal{A}$ will be able to read the communications that are being passed back and forth between the permitted entities HD_i, PS_j, CS_k, and TA. This inquiry is modeled around an attempt to listen in on conversations (eavesdropping).
- $Send(\Pi^t, Msg)$: Whenever $\mathcal{A}$ runs this query, it has the ability to send a message, denoted by the symbol msg, to a participant instance Π^t and also has the capacity to receive a return message. A query of this nature is treated as an active attack model.
- $Reveal(\Pi^t)$: The currently active session key SK_{ij} that was computed by Π^t (and its partner) is divulged to $\mathcal{A}$ as soon as this query is executed.
- $Test(\Pi^t)$: The semantic security of the session key SK_{ij} that was established between HD_i and PS_j is put to the test by this query. Before beginning the experiment, the first thing that is done is to toss a coin c that has no bearing on the outcome. After then, the result is only kept a secret from $\mathcal{A}$, and it is used to determine the result of the $Test$ query. After executing this query, an instance Π^t will return SK_{ij} when $c = 1$, and SK_{ij} will be either a new number or a random number when $c = 0$; in all other cases, the result will be $\perp$ (null).

According to the threat model that has been presented, it is assumed that both the PS_j and the TA can be relied upon. As a result, it is presumed that $Test(\Pi^t)$ does not have access to any other potentially corrupt queries that are associated with either the PS_j or the TA.

$\mathcal{A}$ is able to have a large number of $Test$ queries directed to either $\Pi^{t}_{HD_i}$ or $\Pi^{u}_{PS_j}$. The outcome of the $Test$ query has to be in line with the random bit c, which is a requirement. At the end of the game, $\mathcal{A}$ will return a guessed bit of c', and if $c' = c$, then he or she will have won the game. If $Succ$ indicates that there is a possibility that $\mathcal{A}$ will win the game, then the advantage $Adv_{\mathcal{A}}^{\text{SBAKM-HS}}$ that $\mathcal{A}$ has over breaking the semantic security of the authenticated key agreement system (SBAKM-HS) in order to derive SK_{ij} from HD_i and PS_j is denoted by

$$Adv_{\mathcal{A}}^{\text{SBAKM-HS}} = |2.Pr[Succ] - 1|.$$

Random oracle. The collision-resistant one-way cryptographic hash function $h(\cdot)$ will be accessible to all the communication entities, including $\mathcal{A}$. This includes all the communicating entities. As described in [28], we model $h(\cdot)$ as a random oracle denoted by $\mathcal{O}_{\mathcal{H}}$.

B. SECURITY PROOF

In Theorem 1, we now give the semantic security of SBAKM-HS.

Theorem 1. *In the ROR model, let $\mathcal{A}$ represent a rival that competes against our SBAKM-HS method while operating in polynomial time t. The advantage that $\mathcal{A}$ brings to the process of breaching the semantic security of SBAKM-HS can be estimated as*

$$Adv_{\mathcal{A}}^{\textit{SBAKM-HS}} \leq \frac{q_{hash}^2}{|Hash|}.$$

where q_{hash} and $|Hash|$ are the number of hash $\mathcal{O}_{\mathcal{H}}$ queries and the range space of $h(\cdot)$, respectively.

Proof. We proceed according to a demonstration of this theorem that is analogous to the one stated in [28], [21]. We will define the four games $Game_j$ $(j = 0, 1, 2)$ in which the event denoted by the variable $Succ_j$ may occur, namely that $\mathcal{A}$ may successfully predict the bit c in $Game_j$ and therefore win that game. $Game_0$ is the actual attack, which kicks off the game, while $Game_2$ is the final step in the process of finishing the game. Below you will find a more in-depth explanation of each of these games.

- $Game_0$**:** It is modeled after the actual assault, in which the bit c has to be selected by $\mathcal{A}$ before anything else can happen. Therefore, it follows that

$$Adv_{\mathcal{A}}^{\text{SBAKM-HS}} = |2.Pr[Succ_0] - 1|. \quad (1)$$

- $Game_1$**:** An attack involving listening in on conversations is represented using this game. $\mathcal{A}$ starts off by $Execute$ querying, then moves on to $Test$ query. After that, $\mathcal{A}$ will need to determine whether the session key SK_{HD_i,PS_j} is an actual number or a made-up one. The computed session key is $SK_{HD_i,PS_j} = h(h(rs_2||\ RID_{PS_j}||\ TC_{PS_j})||\ h(rs_1||\ MK_{HD_i}||\ TC_{HD_i})||\ RID_{HD_i}||\ K_{HD_i,PS_j}||TS_1||\ TS_2)$, so keep that in mind. Let's say that during the authentication and key establishment process, $\mathcal{A}$ manages to get a hold of the messages $msg_1 = \{TID_{HD_i},\ M_1,\ M_2,\ TS_1\}$, $msg_2 = \{M_3,\ M_4,\ M_5,\ TS_2\}$, $msg_3 = \{M_6,\ TS_3\}$. However, the computation of SK_{HD_i,PS_j} is not helped by any of these messages because it is computationally difficult to derive the secret credentials (i.e., rs_1, rs_2, K_{HD_i,PS_j}, k_{HD_i}, k_{TA}, k_{PS_j}, RID_{HD_i}, RID_{PS_j})

```
hashfunction h;
const xor:Function;
const cat:Function;
protocol siddhant(HD,PS)
{
 role HD
 {
   fresh rs1:Nonce;
   const TIDHDi,TIDHDin,RIDHDi,TCHDi,MKHDi,KHDiPSj,RIDPSj,TCPSj,TS1,
   TS2,TS3,rs1,rs2;
   var rs2:Nonce;
   macro M1 = xor(h(cat(RIDHDi,KHDiPSj,TS1)),h(cat(rs1,TCHDi,MKHDi)));
   macro M2 = h(cat(h(cat(rs1,MKHDi,TCHDi)),TS1));
   macro SKHDiPSj = h(cat(h(cat(rs2,RIDPSj,TCPSj),h(cat(rs1,MKHDi,TCHDi)),
   RIDHDi,KHDiPSj,TS1,TS2)));
   macro M6 = h(cat(SKHDiPSj,TS3,TIDHDin));
   send_1(HD,PS,cat(TIDHDi,M1,M2,TS1));
   recv_2(PS,HD,cat(xor(h(cat(rs2,RIDPSj,TCPSj)),h(cat(RIDHDi,KHDiPSj,TS1,TS2))),
   h(cat(h(cat(h(cat(rs2,RIDPSj,TCPSj),h(cat(rs1,MKHDi,TCHDi)),RIDHDi,KHDiPSj,
   TS1,TS2))),RIDHDi,TS1,TS2)),xor(TIDHDin,h(cat(RIDHDi,h(cat(rs2,RIDPSj,TCPSj)),
   TS1,TS2))),TS2));
   send_3(HD,PS,cat(M6,TS3));
   claim_HD1(HD,Secret,(rs1));
   claim_HD3(HD,Secret,(RIDHDi));
   claim_HD4(HD,Secret,(TCHDi));
   claim_HD5(HD,Secret,(MKHDi));
   claim_HD6(HD,Niagree);claim_HD7(HD,Nisynch);
   claim_HD8(HD,Secret,(SKHDiPSj));
   claim_HD9(HD,Secret,(KHDiPSj));
   claim_HD10(HD,Weakagree);claim_HD11(HD,Alive);
 }
```

FIGURE 2: SPDL snippet for the role of a smart healthcare device HD

that create this key. This suggests that this information does not boost $\mathcal{A}$'s chances of winning $Game_1$ through the eavesdropping attack. As a result, both $Game_0$ and $Game_1$ are comparable, and since this is the case, we obtain,

$$Pr[Succ_1] = Pr[Succ_0]. \tag{2}$$

- $Game_2$**:** Within the confines of this game, $\mathcal{A}$ has the ability to query the $Send$ and $\mathcal{O}_{\mathcal{H}}$. $\mathcal{A}$ is able to construct a message on behalf of a participant. To create legal messages msg_1, msg_2 and msg_3. The secret credentials (i.e., rs_1, rs_2, K_{HD_i,PS_j}, k_{HD_i}, k_{TA}, k_{PS_j}, RID_{HD_i}, RID_{PS_j}) are necessary for $\mathcal{A}$. However, using a collision-resistant hash function known as $h(\cdot)$, these secret credentials are concealed alongside the hash results. In addition, each session's messages msg_1, msg_2, and msg_3 are differentiated from the others due to the use of random secret numbers (rs_1, rs_2) and current timestamps (TS_1, TS_2, TS_3). As a result, hash values do not experience collisions. The games $Game_1$ and $Game_2$ are exactly the same except that $Game_2$ simulates the $Send$ and $\mathcal{O}_{\mathcal{H}}$ queries. The following is the conclusion that can be drawn from the birthday paradox:

$$|Pr[Succ_1] - Pr[Succ_2]| \leq \frac{q_{hash}^2}{2.|Hash|}. \tag{3}$$

Since all the queries have been executed by the adversary $\mathcal{A}$, it is only left to guess the correct bit c. It then follows that

$$Pr[Succ_2] = \frac{1}{2}. \tag{4}$$

Eqs. (1)–(4) give

$$\begin{aligned} \frac{1}{2}.Adv_{\mathcal{A}}^{\text{SBAKM-HS}} &= |Pr[Succ_0] - \frac{1}{2}| \\ &= |Pr[Succ_1] - Pr[Succ_2]| \\ &\leq \frac{q_{hash}^2}{2.|Hash|}. \end{aligned} \tag{5}$$

Finally, multiplying both sides of Eq. (5) by a factor of 2, we obtain the required result:

$$Adv_{\mathcal{A}}^{\text{SBAKM-HS}} \leq \frac{q_{hash}^2}{|Hash|}.$$

□

VII. FORMAL SECURITY VERIFICATION OF SBAKM-HS USING SCYTHER TOOL

In this section, we talk about the formal security verification of SBAKM-HS. Via the use of the Scyther tool, formal security of SBAKM-HS is verified [3], [29], [30]. It is a better and more effective tool for falsifying, verifying, and analysing the given security protocol when compared

```
role PS
{
  fresh rs2:Nonce;
  const TIDHDi,TIDHDin,RIDHDi,TCHDi,MKHDi,KHDiPSj,RIDPSj,TCPSj,TS1,
  TS2,TS3,rs1,rs2;
  var rs1:Nonce;
  macro M3 = xor(h(cat(rs2,RIDPSj,TCPSj)),h(cat(RIDHDi,KHDiPSj,TS1,TS2)));
  macro SKPSjHDi = h(cat(h(cat(rs2,RIDPSj,TCPSj),h(cat(rs1,MKHDi,TCHDi)),
  RIDHDi,KHDiPSj,TS1,TS2)));
  macro M4 = h(cat(SKPSjHDi,RIDHDi,TS1,TS2));
  macro M5 = xor(TIDHDin,h(cat(RIDHDi,h(cat(rs2,RIDPSj,TCPSj)),TS1,TS2)));
  recv_1(HD,PS,cat(TIDHDi,xor(h(cat(RIDHDi,KHDiPSj,TS1)),h(cat(rs1,TCHDi,
  MKHDi))),h(cat(h(cat(rs1,MKHDi,TCHDi)),TS1)),TS1));
  send_2(PS,HD,cat(M3,M4,M5,TS2));
  recv_3(HD,PS,cat(h(cat(h(cat(h(cat(rs2,RIDPSj,TCPSj),h(cat(rs1,MKHDi,
  TCHDi)),RIDHDi,KHDiPSj,TS1,TS2))),TS3,TIDHDin)),TS3));
  claim_PS1(PS,Secret,(rs2));
  claim_PS2(PS,Secret,(RIDPSj));
  claim_PS3(PS,Secret,(TCPSj));
  claim_PS4(PS,Niagree);claim_PS5(PS,Nisynch);
  claim_PS6(PS,Secret,(SKPSjHDi));
  claim_PS7(PS,Secret,(KHDiPSj));
  claim_PS8(PS,Weakagree); claim_PS9(PS,Alive);
}
```

FIGURE 3: SPDL snippet for the role of a personal server PS

Claim				Status	Comments
siddhant	HD	siddhant,HD1	Secret rs1	Ok	No attacks within bounds.
		siddhant,HD3	Secret RIDHDi	Ok	No attacks within bounds.
		siddhant,HD4	Secret TCHDi	Ok	No attacks within bounds.
		siddhant,HD5	Secret MKHDi	Ok	No attacks within bounds.
		siddhant,HD6	Niagree	Ok	No attacks within bounds.
		siddhant,HD7	Nisynch	Ok	No attacks within bounds.
		siddhant,HD8	Secret h(cat(h(cat(rs2,RIDPSj,TCPSj),h(cat(rs1,MKH...	Ok	No attacks within bounds.
		siddhant,HD9	Secret KHDiPSj	Ok	No attacks within bounds.
		siddhant,HD10	Weakagree	Ok	No attacks within bounds.
		siddhant,HD11	Alive	Ok	No attacks within bounds.
	PS	siddhant,PS1	Secret rs2	Ok	No attacks within bounds.
		siddhant,PS2	Secret RIDPSj	Ok	No attacks within bounds.
		siddhant,PS3	Secret TCPSj	Ok	No attacks within bounds.
		siddhant,PS4	Niagree	Ok	No attacks within bounds.
		siddhant,PS5	Nisynch	Ok	No attacks within bounds.
		siddhant,PS6	Secret h(cat(h(cat(rs2,RIDPSj,TCPSj),h(cat(rs1,MKH...	Ok	No attacks within bounds.
		siddhant,PS7	Secret KHDiPSj	Ok	No attacks within bounds.
		siddhant,PS8	Weakagree	Ok	No attacks within bounds.

FIGURE 4: Results of security verification using Scyther tool

to existing verification tools like ProVerif and AVISPA. The scyther tool is predicated on optimal suppositions of cryptography. It means that an attacker won't be able to decrypt the data without the secret key. Via the usage of Security Protocol Descriptive Language (SPDL), it simulates user-defined security protocols. In the SPDL specification, each communicating party is represented by a specific role, which is capable of performing a number of functions, like as events, required security claims, and send (it is sending of a message), recv (it is receiving of a message) [31]. The Scyther tool works through the guidelines of the Dolev-Yao (DY) model plus nine more adversarial models, i.e., eCK model and the CK model. The tests offered by Scyther are said to validate security aspects like "secrecy, authentication, synchronisation, aliveness, weak agreement and agreement." In SBAKM-HS, we consider two basic roles for the simulation of authentication and key agreement phase, which are HD (for smart healthcare device) and PS (for personal server). SBAKM-HS is then implemented using the SPDL. The SPDL snippets of SBAKM-HS (for the role of HD) and (for the role of PS) are given in Fig. 2 and Fig. 3. Finally, Fig. 4 shows the results obtained Scyther tool's implementation and analysis. Following examination, it was determined that the SBAKM-HS is protected by the mentioned claims.

VIII. PERFORMANCE COMPARISONS OF SBAKM-HS

This section includes information on numerous comparisons between the SBAKM-HS and other present schemes, like as, the technique of He-Zeadally [32], Deebak and Al-Turjman [33], Jang *et al.* [34], Das and Namasudra [5] and Merabet *et al.* [35] (both protocols). In terms of "communication costs, computation costs, and security and functionality factors," comparisons have been made.

Table 2 compares several key elements of proposed SBAKM-HS to a few recently released schemes in terms of security and functionality attributes. As compared to the mechanisms of He-Zeadally [32], Deebak and Al-Turjman [33], Jang *et al.* [34], Das and Namasudra [5] and Merabet *et al.* [35] (both protocols), it has been shown that proposed SBAKM-HS offers better security and more functionality capabilities.

In order to compare computational costs, T_h and T_{fe} notations are used, which are the time duration of a "one-way hash function (say, SHA-256 hashing algorithm)" and a "fuzzy extractor function $(Gen(\cdot)/Rep(\cdot)$," subsequently. Table 3 lists the execution times for various cryptographic operations utilized in [4].

Then Table 4 compares the cost of computation for several approaches. In the "authentication and key agreement phase", the proposed SBAKM-HS needs $16T_h \approx 5.12$ ms. The computation cost of the SBAKM-HS is better than most of the compared schemes, for example, mechanisms [34], [32], [35] and [33]. Moreover, the scheme of Das and Namasudra [5] has less computation cost, however, it may be approved as SBAKM-HS provides better security and extra factors as compared to the scheme of Das and Namasudra [5]. Also, Das and Namasudra protocol [5] requires less computation, but it can still be chosen because the proposed SBAKM-HS offers superior security and other operational capabilities.

In the "authentication and key agreement phase" of proposed SBAKM-HS, the "messages $msg_1 = \{TID_{HD_i}, M_1, M_2, TS_1\}$, $msg_2 = \{M_3, M_4, M_5, TS_2$ to HD_i and $msg_3 = \{M_6, TS_3\}$ are transmitted between HD_i and PS_j." Suppose an "identity", a "timestamp", a "random number (nonce)" and a "hash output (if SHA-256 hashing algorithm is applied)" take 160 bits, 32 bits, 160 bits and 256 bits. Then, proposed SBAKM-HS's messages msg_1, msg_2 and msg_3 need $(800 + 800 + 288) = 1888$ bits. The entire SBAKM-HS communication cost is thus calculated to be 1888 bits.

The transmission costs of SBAKM-HS and other equivalent schemes are contrasted in Table 5. The information provided makes it evident that SBAKM-HS requires less communication expense than published methods. Nonetheless, it still outperforms Deebak- Al-Turjman [33]. The proposed SBAKM-HS is more acceptable than Deebak-Al-Turjman [33] since it provides greater security and functionality aspects.

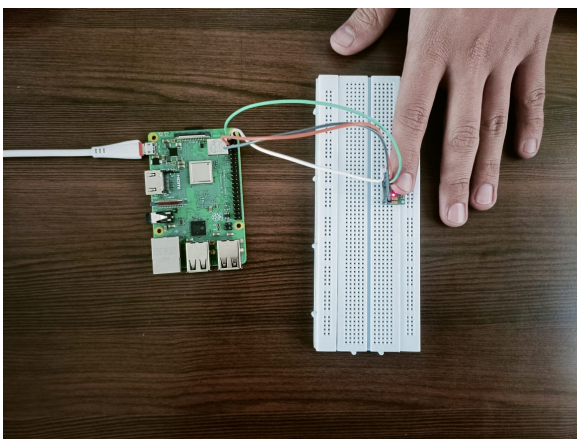

FIGURE 5: View of an implemented testbed of the proposed SBAKM-HS

IX. TESTBED IMPLEMENTATION OF THE PROPOSED SBAKM-HS

We describe the implementation of the testbed for the proposed SBAKM-HS in this section. Information on the several parameters used in the investigations is provided in Table 6. For development, we used a "Raspberry Pi 3 device with Bluetooth 4.1 and Wireless LAN, model number B802.11." The healthcare sensing units were the MLX90614 and MAX30100 (Heartbeat and Obs2). The work was carried out using the Python 3.9.2 programming language. Other tools used included "advance IP Scanner (to get the raspberry pi's IP address), PUTTY (to connect to the raspberry pi on the system), and VNCViewer (to

TABLE 2: Comparative analysis of security and functionality factors

Feature	Merabet *et al.* [35]	Jang *et al.* [34]	He-Zeadally [32]	Deebak-Al-Turjman [33]	Das and Namasudra [5]	SBAKM-HS
ΨFA_1	✓	✓	✓	✓	✓	✓
ΨFA_2	✓	×	✓	✓	✓	✓
ΨFA_3	✓	✓	✓	×	✓	✓
ΨFA_4	✓	✓	✓	✓	✓	✓
ΨFA_5	×	✓	✓	×	✓	✓
ΨFA_6	✓	✓	✓	✓	✓	✓
ΨFA_7	✓	✓	✓	✓	✓	✓
ΨFA_8	×	✓	✓	✓	✓	✓
ΨFA_9	✓	✓	✓	✓	✓	✓
ΨFA_{10}	✓	N/A	✓	N/A	✓	✓
ΨFA_{11}	N/A	N/A	N/A	N/A	✓	✓
ΨFA_{12}	N/A	N/A	N/A	N/A	✓	✓
ΨFA_{13}	×	×	×	×	✓	✓
ΨFA_{14}	×	N/A	✓	N/A	✓	✓
ΨFA_{15}	✓	✓	✓	✓	✓	✓
ΨFA_{16}	×	×	×	×	×	✓
ΨFA_{17}	×	×	×	×	×	✓
ΨFA_{18}	✓	×	×	×	✓	✓

Note: ΨFA_1: "mutual authentication/access control"; ΨFA_2: "anonymity"; ΨFA_3: "untraceability"; ΨFA_4: "session-key agreement"; ΨFA_5: "session key security under CK adversary model"; ΨFA_6: "confidentiality"; ΨFA_7: "integrity"; ΨFA_8: "strong replay attack"; ΨFA_9: "man-in-the-middle attack"; ΨFA_{10}: "efficient login phase"; ΨFA_{11}: "password update phase"; ΨFA_{12}: "biometric update phase"; ΨFA_{13}: "dynamic smart healthcare device addition"; ΨFA_{14}: "protection against stolen smart phone/ mobile device/ programmer attack"; ΨFA_{15}: "protection against impersonation attack"; ΨFA_{16}: "blockchain-based protection"; ΨFA_{17}: "presence of session key verification"; ΨFA_{18}: "formal security verification using Scyther/ AVISPA/ ProVerif 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 3: An approximate time estimate for various cryptography elements [4]

Notation	Characterisation (computation duration)	Approx. computation time (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 4: Comparing of computing requirements

Mechanism	Computation cost (ms)
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
Case-1 of scheme (Merabet *et al.*)	$6T_{ecm} + 6T_h + T_{eca} \approx 108.92$ ms
Case-2 of scheme (Merabet *et al.*)	$4T_{ecm} + 4T_h \approx 69.68$ ms
Deebak- Al-Turjman	$4T_{senc}/T_{sdec} + 8T_h + 1T_{me} \approx 44.16$ ms
Das and Namasudra	$9T_h \approx 2.88$ ms
SBAKM-HS	$16T_h \approx 5.12$ ms

TABLE 5: Comparing of communication costs

Mechanism	No. of messages	Bits exchange
Jang *et al.*	8	5920
He-Zeadally	4	3232
Case-1 of scheme (Merabet *et al.*)	3	1472
Case-2 of scheme (Merabet *et al.*)	3	1472
Deebak- Al-Turjman	3	800
Das and Namasudra	6	3456
SBAKM-HS	3	1888

display the GUI interface of the raspberry pi)." We have used "Google's Android Studio and Java 8 to construct an Android app." The personal server was contemplated using Google Firebase. The developed Android app has capabilities like user registration and registration of various smart healthcare equipment with associated patients. Every time, it also provides login information based on biometrics and credentials. Additionally, it offers secure access to the healthcare sensor's most recent data, a history of healthcare data that was sensed and received, and patient information (such as identification (ID) patient, health condition parameter (physiological parameters), etc.,).

Fig. 5 shows the snapshot of an implemented SBAKM-HS testbed. Fig. 6 depicts the view of the biometric-based login process in the developed Android application (app). After successful biometric login, the user gets the next step, as shown in Fig. 6, where a user has to provide his/ her unique ID and password (credentials) for the successful login process.

After successful login, a user can access the dashboard of the application as shown in Fig. 7, in which the healthcare data in real-time can be viewed as shown in Fig. 7. It has

TABLE 6: Information about the settings used to implement the testbed

Parameter	Description
Specifics of smart healthcare devices	
Raspberry pi 3	With Model B802.11.b/g/n Wireless
	and LAN Bluetooth 4.1
Health data sensing unit	MAX30100 (for heartbeat and oxygen level)
	and MLX90614 (for temperature of body)
Operating system inside raspberry pi	Raspberry OS
Tools in action	Advance IP Scanner (obtaining of Raspberry Pi's IP address)
	PUTTY (for using a system to access the Raspberry Pi)
	VNCViewer (for GUI interface)
Language	Python version 3.9.2
Deployed tools	Google's android studio
Database server deployment	Google's firebase
Functions in developed app	
	Login via credentials and biometrics
	Secure accessing of data of health data sensing unit
	Maintaining of history of healthcare data
	Has patient details, like, ID of patient, health condition of the patient, etc.)

views of patients' details, history of medical records, lab test records, etc. The readings of smart healthcare devices (i.e., sensors) can be seen. For example, the implemented application provides readings like the patient's pulse, temperature, and Spo2 level, which are shown here 70, $36°C$, and 97, respectively.

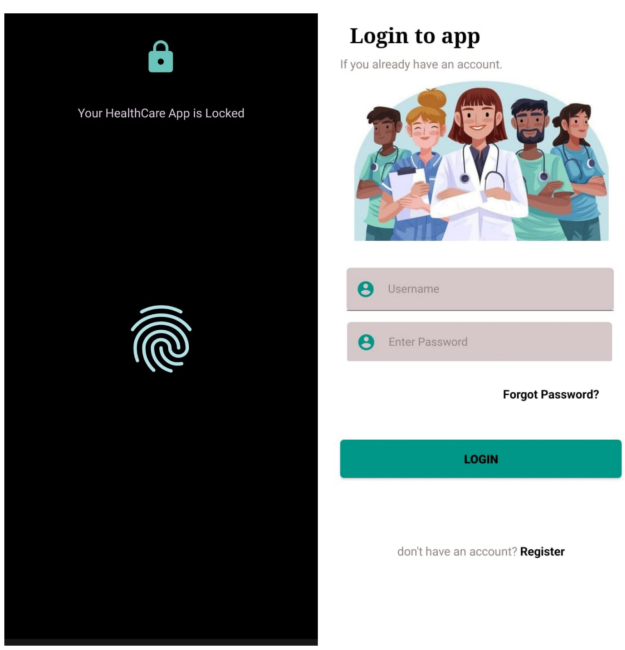

FIGURE 6: View of biometric-based login process in the developed android app

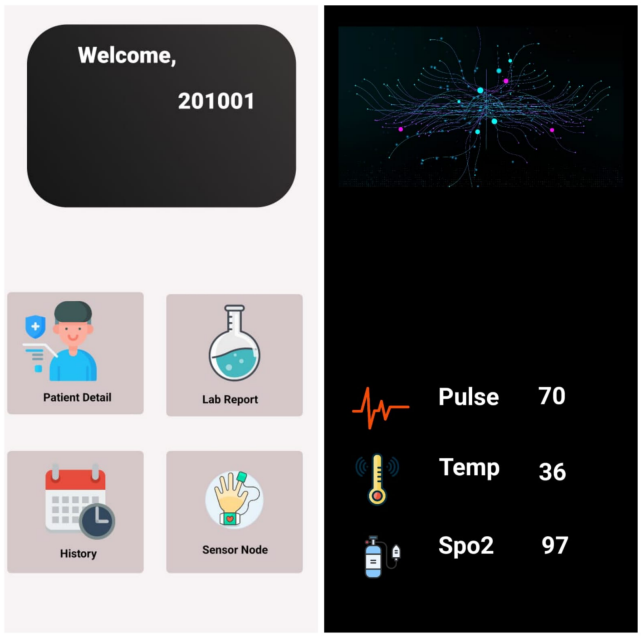

FIGURE 7: View of user in developed Android app and monitored physiological parameters

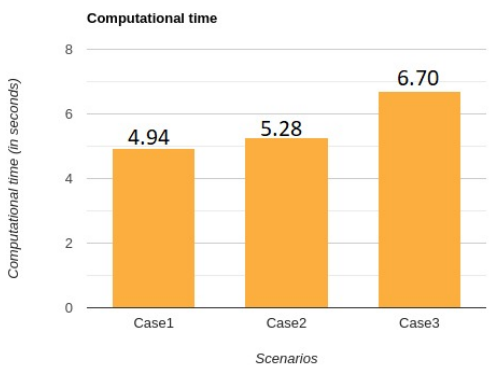

FIGURE 8: Calculations of computational cost

A. IMPLEMENTATION OF BLOCKCHAIN PHASE

In this section, the details of the implementation of the blockchain phase of the proposed SBAKM-HS are provided [8], [36]. In the simulation parts, various cases are considered. The implementation is done on a "Windows 10 64-bit system with an Intel (R) core i5-8250U processor running at 1.60 GHz-1.80 GHz." The random-access memory (RAM)'s size of the system is 8 GB. "Eclipse IDE 2019-12" is used for the platform along with Java programming language. The number of "smart healthcare devices" are considered in each scenario as 10 (in case-1), 20 (in case-2), and 40 (in case-3). We have taken 5 in case-1, 10 in case-2, and 15 in case-3 number blocks in the blockchain's implementation. Total users are considered as 10 in case-1, 20 in case-2, and 40 in case-3, respectively. Further, 4 miner nodes (i.e.,

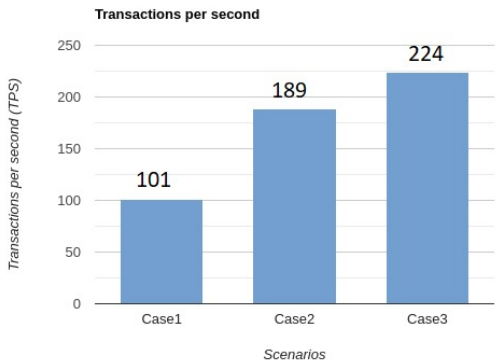

FIGURE 9: Calculations of transactions per second (TPS)

cloud servers) are considered in each case. Following are the estimated specifications of the findings.

- **Calculation of computational time:** In all situations taken into account, the impact of an increase in users and healthcare equipment is calculated in terms of computation time (in seconds). For example, the expected computation times (in seconds) for cases 1, 2, and 3 are 4.94, 5.28, and 6.70, respectively. The results are shown in Figure 8. It's vital to note that when we move from case 1 to case 2 and case 2 to case 3, the computational time increases due to an increase in the number of devices and users. It occurred because new blocks needed to be created and added to the blockchain as a result of those events.
- **Calculation of transactions per second (TPS):** The impact of more users and smart healthcare devices is also evaluated in terms of transactions per second for all scenarios taken into account (TPS). For example, the projected TPS values for cases 1, 2, and 3 are 101, 189, and 224, respectively. The results are shown in Figure 9. It's crucial to keep in mind that as the blockchain expands, more users and devices join it, the number of transactions per second (TPS) rises. It occurred because new blocks needed to be created and added to the blockchain as a result of those events. It progressively increases TPS's value.

X. CONCLUSION

For an IoMT-based smart healthcare system, we presented a blockchain-driven authentication and key management mechanism (in short SBAKM-HS). When the performance was compared, the proposed SBAKM-HS performed better than other existing methods. The formal security verification performed using the Scyther tool and the conducted security analysis demonstrated the proposed SBAKM-HS's security against a variety of potential attacks. Last but not least, the testbed implementation of SBAKM-HS was offered so that its effect on the system's performance could be assessed.

We intend to expand the functionality capabilities of SBAKM-HS in the future.

ACKNOWLEDGEMENTS

The authors would like to thank the "anonymous reviewers and the Associate Editor for their valuable feedback."

REFERENCES

[1] M. A. Ferrag and L. Shu, "The performance evaluation of blockchain-based security and privacy systems for the internet of things: A tutorial," IEEE Internet of Things Journal, vol. 8, no. 24, pp. 17 236–17 260, 2021.
[2] M. Wazid, A. K. Das, and Y. Park, "Blockchain-enabled secure communication mechanism for iot-driven personal health records," Transactions on Emerging Telecommunications Technologies, vol. 33, no. 4, p. e4421, 2022.
[3] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "Lake-iod: Lightweight authenticated key exchange protocol for the internet of drone environment," IEEE Access, vol. 8, pp. 155 645–155 659, 2020.
[4] D. He, N. Kumar, J. H. Lee, and R. S. Sherratt, "Enhanced three-factor security protocol for consumer USB mass storage devices," IEEE Transactions on Consumer Electronics, vol. 60, no. 1, pp. 30–37, 2014.
[5] S. Das and S. Namasudra, "A Lightweight and Anonymous Mutual Authentication Scheme for Medical Big Data in Distributed Smart Healthcare Systems," IEEE/ACM Transactions on Computational Biology and Bioinformatics, pp. 1–12, 2022.
[6] A. Anand and A. K. Singh, "Sdh: Secure data hiding in fused medical image for smart healthcare," IEEE Transactions on Computational Social Systems, vol. 9, no. 4, pp. 1265–1273, 2022.
[7] P. K. Roy, A. Singh, J. V. Desai, and S. K. Singh, "Healthcare data security using lightweight protocol for cyber physical system," IEEE Transactions on Network Science and Engineering, pp. 1–10, 2022.
[8] N. Garg, M. Wazid, A. K. Das, D. P. Singh, J. J. P. C. Rodrigues, and Y. Park, "BAKMP-IoMT: Design of Blockchain Enabled Authenticated Key Management Protocol for Internet of Medical Things Deployment," IEEE Access, vol. 8, pp. 95 956–95 977, 2020.
[9] M. Wazid, A. K. Das, K.-K. R. Choo, and Y. H. Park, "SCS-WoT: Secure Communication Scheme for Web of Things Deployment," IEEE Internet of Things Journal, pp. 1–1, 2021, doi:10.1109/JIOT.2021.3122007.
[10] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," in International Conference on the Theory and Applications of Cryptographic Techniques– Advances in Cryptology (EUROCRYPT 2002), Amsterdam, The Netherlands, 2002, pp. 337–351.
[11] J. Lu, J. Shen, P. Vijayakumar, and B. B. Gupta, "Blockchain-based secure data storage protocol for sensors in the industrial internet of things," IEEE Transactions on Industrial Informatics, vol. 18, no. 8, pp. 5422–5431, 2022.
[12] S. Chatterjee, A. K. Das, and J. K. Sing, "An Enhanced Access Control Scheme in Wireless Sensor Networks," Ad Hoc & Sensor Wireless Networks, vol. 21, no. 1-2, pp. 121–149, 2014.
[13] A. K. Das, "A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications," Networking Science, vol. 2, no. 1, pp. 12–27, 2013.
[14] A. K. Das and B. Bruhadeshwar, "An Improved and Effective Secure Password-Based Authentication and Key Agreement Scheme Using Smart Cards for the Telecare Medicine Information System," Journal of Medical Systems, vol. 37, no. 5, p. 9969, 2013.
[15] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card," Peer-to-Peer Networking and Applications, vol. 9, no. 1, pp. 171–192, 2016.
[16] 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 Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5535–5548, 2020.
[17] W. Xiao, M. Li, B. Alzahrani, R. Alotaibi, A. Barnawi, and Q. Ai, "A blockchain-based secure crowd monitoring system using uav swarm," IEEE Network, vol. 35, no. 1, pp. 108–115, 2021.
[18] R. Paul, N. Ghosh, S. Sau, A. Chakrabarti, and P. Mohapatra, "Blockchain based secure smart city architecture using low resource iots," Computer Networks, vol. 196, p. 108234, 2021.

[19] X. Yang, X. Yang, X. Yi, I. Khalil, X. Zhou, D. He, X. Huang, and S. Nepal, "Blockchain-based secure and lightweight authentication for internet of things," IEEE Internet of Things Journal, vol. 9, no. 5, pp. 3321–3332, 2022.
[20] D. Dolev and A. Yao, "On the security of public key protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198–208, 1983.
[21] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment," IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 2, pp. 391–406, 2020.
[22] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," in Advances in Cryptology (CRYPTO'92), E. F. Brickell, Ed. Santa Barbara, California, USA: Springer Berlin Heidelberg, 1993, pp. 471–486.
[23] 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 Journal of Biomedical and Health Informatics, vol. 22, no. 4, pp. 1299–1309, 2018.
[24] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," IEEE Transactions on Computers, vol. 51, no. 5, pp. 541–552, 2002.
[25] "Databases on AWS," 2021, https://aws.amazon.com/products/databases/. Accessed on July 2021.
[26] M. Wazid, B. Bera, A. K. Das, S. P. Mohanty, and M. Jo, "Fortifying smart transportation security through public blockchain," IEEE Internet of Things Journal, pp. 1–1, 2022, doi:10.1109/JIOT.2022.3150842.
[27] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science (LNCS), vol. 3386, Les Diablerets, Switzerland, 2005, pp. 65–84.
[28] C. C. Chang and H. D. Le, "A Provably Secure, Efficient, and Flexible Authentication Scheme for Ad hoc Wireless Sensor Networks," IEEE Transactions on Wireless Communications, vol. 15, no. 1, pp. 357–366, 2016.
[29] B. Khadem, A. M. Suteh, M. Ahmad, A. Alkhayyat, M. S. Farash, and H. S. Khalifa, "An Improved WBSN Key-Agreement Protocol Based on Static Parameters and Hash Functions," IEEE Access, vol. 9, pp. 78 463–78 473, 2021.
[30] C. J. F. Cremers, "Scyther : semantics and verification of security protocols," https://pure.tue.nl/ws/files/2425555/200612074.pdf. Accessed on November 2022.
[31] M. Adeli, N. Bagheri, and H. R. Meimani, "On the designing a secure biometric-based remote patient authentication scheme for mobile healthcare environments," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 2, pp. 3075–3089, 2021.
[32] D. He and S. Zeadally, "Authentication protocol for an ambient assisted living system," IEEE Communications Magazine, vol. 53, no. 1, pp. 71–77, 2015.
[33] B. D. Deebak and F. Al-Turjman, "Smart Mutual Authentication Protocol for Cloud Based Medical Healthcare Systems Using Internet of Medical Things," IEEE Journal on Selected Areas in Communications, vol. 39, no. 2, pp. 346–360, 2021.
[34] C. S. Jang, D. G. Lee, J.-w. Han, and J. H. Park, "Hybrid Security Protocol for Wireless Body Area Networks," Wireless Communications and Mobile Computing, vol. 11, no. 2, pp. 277–288, 2011.
[35] 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, 2020.
[36] M. Fan and X. Zhang, "Consortium blockchain based data aggregation and regulation mechanism for smart grid," IEEE Access, vol. 7, pp. 35 929–35 940, 2019.

SIDDHANT THAPLIYAL (Student Member, IEEE) "is pursuing PhD CSE degree in the Department of Computer Science and Engineering at Graphic Era Deemed to be University Dehradun, India. He has done M. Tech in Computer Science and Engineering in the Department of Computer Science and Engineering at Graphic Era Deemed to be University Dehradun, India. He has also done has done B. Tech in Computer Science and Engineering in the Department of Computer Science and Engineering at Graphic Era Hill University Dehradun, India. His area of research is information security, authentication, access control, blockchain and machine learning. He has published more than 10 papers in international journals and conferences in the above areas."

MOHAMMAD WAZID (Senior Member, IEEE) "has received his Master of Technology in Computer Network Engineering from Graphic Era University, Dehradun, India, and received a Ph.D. in Computer Science and Engineering from the International Institute of Information Technology, Hyderabad, India. He is currently working as a Professor in the Department of Computer Science and Engineering, at Graphic Era University, Dehradun, India. He is the head of the cybersecurity and IoT research group at Graphic Era University, Dehradun, India. Prior to this, he was an assistant professor in the Department of Computer Science and Engineering at the Manipal Institute of Technology, MAHE, Manipal, India. He was also a postdoctoral researcher in the cyber security and networks lab, at Innopolis University, Innopolis, Russia. His current research interests include security, remote user authentication, the Internet of Things (IoT), and cloud computing. He has published more than 100 papers in international journals and conferences in the above areas. He was a recipient of the University Gold Medal and the Young Scientist Award from UCOST, the Department of Science and Technology, Government of Uttarakhand, India. He is a senior member of IEEE".

DEVESH PRATAP SINGH is currently the Professor and Head of Computer Science and Engineering department at Graphic Era deemed to be University Dehradun India. He has received M. Tech degree in Computer Science and Engineering from Uttarakhand Technical University Dehradun India in 2009. He has also received Ph.D. in 2015. His research interests include Information Security, Wireless Sensor Networks, Internet of Things and Soft Computing. He has published more than 50 research papers in his area of expertise.

ASHOK KUMAR DAS (Senior Member, IEEE) "received a Ph.D. degree in computer science and engineering, an M.Tech. degree in computer science and data processing, and an M.Sc. degree in mathematics from IIT Kharagpur, India. He is currently a full Professor with the Center for Security, Theory and Algorithmic Research, IIIT, Hyderabad, India. He was also a visiting faculty with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA 23435, USA. His research interests include cryptography, system and network security, blockchain, security in the Internet of Things (IoT), Internet of Vehicles (IoV), Internet of Drones (IoD), smart grids, smart city, cloud/fog computing, intrusion detection, AI/ML security, and post-quantum cryptography. He has authored over 360 papers in international journals and conferences in the above areas, including over 305 reputed journal papers. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He has been listed in the Web of Science (Clarivate^{TM}) Highly Cited Researcher 2022 in recognition of his exceptional research performance. He is/was on the editorial board of IEEE Systems Journal, Journal of Network and Computer Applications (Elsevier), Computer Communications (Elsevier), Journal of Cloud Computing (Springer), Cyber Security and Applications (Elsevier), IET Communications, KSII Transactions on Internet and Information Systems, and International Journal of Internet Technology and Secured Transactions (Inderscience). He also severed as one of the Technical Program Committee Chairs of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019, International Conference on Applied Soft Computing and Communication Networks (ACN'20), October 2020, Chennai, India, and second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, October 2020. His Google Scholar h-index is 79 and i10-index is 224 with over 17,600 citations."

SACHIN SHETTY (Senior Member, IEEE) "received the Ph.D. degree in modeling and simulation from Old Dominion University in 2007. He was an associate professor with the Electrical and Computer Engineering Department, Tennessee State University, USA. He is currently a professor with the Virginia Modeling, Analysis and Simulation Center, at Old Dominion University. He holds a joint appointment with the Department of Modeling, Simulation and Visualization Engineering and the Center for Cybersecurity Education and Research. He has authored and co-authored over 200 research articles in journals and conference proceedings and two books. His research interests lie at the intersection of computer networking, network security, and machine learning. He was a recipient of the DHS Scientific Leadership Award. He has served on the Technical Program Committee of ACM CCS, IEEE INFOCOM, IEEE ICDCN, and IEEE ICCCN."

ABDULLAH ALQAHTANI "is an Assistant Professor in Computer Science at the Prince Sattam Bin AbdulAziz University. He received the bachelor degree in computer science from King Saud University, KSA, in 2007, a Master Degree in Advanced Computer Science from the University of Leicester, UK, in 2011, and a PhD degree from the University of Leicester, UK, in 2020. His research interests include big data, IoT Data Engineering and Artificial Intelligence."