AISCM-FH: AI-Enabled Secure Communication Mechanism in Fog Computing-Based Healthcare

Mohammad Wazid, *Senior Member, IEEE*, Ashok Kumar Das, *Senior Member, IEEE*, Sachin Shetty, *Senior Member, IEEE*, Joel J. P. C. Rodrigues, *Fellow, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*— Fog computing-based Internet of Things (IoT) architecture is useful for various types of delay efficient network communications and services, like digital healthcare. However, there are privacy and security issues with the fog computing-based healthcare systems, which can further increase the risk of leakage of sensitive healthcare data. Therefore, a security mechanism, such as access control for fog computing-based healthcare systems, is needed to protect its data against various potential attacks. Moreover, the blockchain technology can be used to solve the digital healthcare's data integrity related problems. The use of Artificial Intelligence (AI) further makes the system more effective in case of prediction of health related diseases. In this paper, an AI-enabled secure communication mechanism in fog computing-based healthcare system (in short, AISCM-FH) has been proposed. The security analysis of the proposed AISCM-FH is provided using the standard random oracle model and also with the heuristic (non-mathematical) security analysis. A pragmatic study determines the impact of the proposed AISCM-FH on key performance indicators. Moreover, we include a detailed performance comparison of AISCM-FH with other relevant existing schemes to show that it has low communication and computation costs, and provides superior security and extra functionality attributes as compared to those for other competing existing approaches.**

***Index Terms*— Smart healthcare, Internet of Things (IoT), fog computing, security, access control and key management, simulation.**

Manuscript received 24 June 2022; revised 25 September 2022; accepted 31 October 2022. Date of publication 9 November 2022; date of current version 7 December 2022. This work was supported in part by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) through the U.S. Army Research Laboratory under Contract W911NF-20-2-0277, in part by FCT/MCTES through National Funds and co-funded EU Funds under Project UIDB/50008/2020, and in part by the Brazilian National Council for Research and Development (CNPq) under Grant 313036/2020-9. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alexey Vinel. *(Corresponding authors: Mohammad Wazid; Ashok Kumar Das.)*

Mohammad Wazid is with the Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248002, India (e-mail: wazidkec2005@gmail.com).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India, and also with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA 23435 USA (e-mail: iitkgp.akdas@gmail.com).

Sachin Shetty is with the Virginia Modeling, Analysis and Simulation Center and the Center for Cybersecurity Education and Research, Department of Modeling, Simulation and Visualization Engineering, Old Dominion University, Suffolk, VA 23435 USA (e-mail: sshetty@odu.edu).

Joel J. P. C. Rodrigues is with the College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266555, China, and also with the Instituto de Telecomunicações, 6201-001 Covilhã, Portugal (e-mail: joeljr@ieee.org).

Mohsen Guizani is with the Machine Learning Department, Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, United Arab Emirates (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/TIFS.2022.3220959

I. Introduction

SMART healthcare is a health service system, which is supported by various tools and technologies, like wearable devices, Internet of Things (IoT) and mobile Internet. It facilitates the dynamic access of information, connect people (i.e., patients, doctors and nursing staff) materials and institutions of healthcare. It actively manages and provides response to medical ecosystem demands in a perceptive manner. It uses various computing platforms, connectivity protocols, software tools and sensors based healthcare devices for their different uses.

IoT is a network consisting of several physical objects that can be assigned to unique addresses, such as Internet Protocol (IP) addresses and can also be accessed over the Internet. The objects can be smart home appliances, smart vehicles and smart healthcare devices. The objects can receive and send data over a network without any assistance from humans by using the unique IP address [1], [2], [3]. IoT communication has a great impact over other technological fields and the innovation across many domains, such as the digital healthcare (i.e., smart healthcare) and automobile sector. Using the IoT communication, data from remotely located smart devices can be accessed in the real time through an Internet connection and smart mobile devices, such as smart phones [4]. Working-class people can get benefit from this because they can conduct their job works while they are also having remote access to smart gadgets, such as smart household appliances [3], [5]. Smart healthcare, self-driving cars, flying IoT (drones), smart surveillance systems, smart cities, industrial IoT (IIoT) and smart farming are some of the possible IoT applications. However, a generic IoT architecture may have issues related to delay and jitter. Therefore, we adopt a fog based IoT communication in this work. Fog computing is an extension of cloud computing that can offer the consumers with delay-free data, storage, computation, and application services. Fog-integrated IoT bridges the gap between remote data centers and IoT devices that can overcome the other concerned problems. The fog servers execute the required computing activities at the fog layer. With the inclusion of fog computing, we get enormous benefits over the traditional architecture, such as increased security, bandwidth reduction, and latency reduction. As a result, it appears to be a viable technology for a variety of IoT applications and associated services [6].

A. Motivation

As mentioned earlier, fog computing-based IoT communications have a wide range of applications in various domains

(i.e., smart healthcare). However, in a fog computing-based IoT context, privacy and security are becoming increasingly important, because it increases the possibility of sensitive data being leaked. It especially matters in case of sensitive healthcare data in smart healthcare. Information mishandling may result in confidential information disclosure to a third party. It is also vulnerable to several forms of security and privacy vulnerabilities, including "privileged-insider, man-in-the-middle, password guessing, replay, impersonation, unlawful session key computation, physical stealing of smart devices, malware injection and many other attacks." As a result, it is critical to safeguard the communication of fog computing-based healthcare systems against these threats. A security mechanism for fog computing-based healthcare system, such as an access control and key management protocol, is required to safeguard its data from any type of assault [2], [3], [7]. Additionally, the blockchain technology can be utilized to improve the security of system. Furthermore, the use of Artificial Intelligence (AI) makes the system more effective in case of prediction of a disease. Thus, this article presents a novel AI-enabled secure communication mechanism in fog computing-based healthcare systems, called AISCM-FH.

B. Research Contributions

The research contributions of the proposed AISCM-FH are summarized as follows.

- The designed AI-enabled secure communication mechanism in fog computing-based healthcare system (AISCM-FH) helps the legitimate devices access other devices in a secure way. The data of smart devices is stored in a Peer-to-Peer (P2P) cloud server network (P2PCS) in a form of a consortium blockchain.
- The system network and threat models associated with the proposed AISCM-FH are provided.
- The formal security of the proposed AISCM-FH is provided through the Real-Or-Random (ROR) oracle model-based random oracle security. Moreover, the security of the AISCM-FH is proved informally using heuristic methods. The offered security studies demonstrate that AISCM-FH is secure against various potential attacks.
- The collation of AISCM-FH with existing schemes shows that AISCM-FH performs significantly better as it incurs less communication and computation costs. In addition, it provides superior security and extra functionality features as compared to those other schemes.
- The pragmatic study of the proposed AISCM-FH is also being done to assess its influence on crucial outcomes.

C. Paper Roadmap

Section II contains a literature evaluation of the related current schemes. The main system models of the proposed AISCM-FH, namely the network model and threat model, are presented in Section III. Various phases associated with the proposed AISCM-FH are described in Section IV. In Section V, various security analyses of AISCM-FH are carried out. The performance of AISCM-FH is compared with other relevant competing schemes in Section VI. In Section VII, we discuss the specifics of AISCM-FH's pragmatic study. Section VIII concludes the paper.

II. Related Work

In this section, we discuss some of the state of the art relevant schemes related to an IoT environment.

Huang et al. [8] proposed a fine-grained data access control technique through the attribute-based signature (ABS) for fog based IoT. Li et al. [9] presented a publicly-verifiable and revocable multi-authority attribute-enabled encryption method to provide a fine-grained access control for the fog computing. Ding et al. [10] further proposed an IoT access control technique based on attributes. They recorded the distribution of attributes via the blockchain technology. It is necessary to avoid some issues, such as data tempering and single point of failure.

Li et al. [11] introduced a scheme for access control and key establishment which is suitable for wireless sensor networks (WSNs) integrated with the IoT environment. A sender belonging to a "certificateless-cryptography (CLC) environment can dispatch a message to a specified recipient under the identity-based cryptography (IBC) environment with the help of their suggested heterogeneous signcryption technique." Their method was drawn through the identity-based access control, which performed bilinear pairing tasks. However, their scheme was heavy as it requires high computation overheads because of the application of the bilinear pairings. Furthermore, a gateway node was used for the access control among two IoT smart devices.

Braeken et al. [12] demonstrated how to use the distributed authentication to control access to an IoT-driven smart home communication. Symmetric encryption and hash functions were used in their method. Despite having a cheap computing cost, their approach had a significant communication cost. In addition, a gateway node was utilized to control access between two IoT smart devices.

Luo et al. [13] introduced a new access control approach for WSNs that are connected to an IoT environment. Their strategy was expensive in terms of computing overheads. In addition, a gateway node was utilized to control access between two IoT smart devices. Furthermore, given the existing *de facto* "Canetti and Krawczyk's adversary (CK-adversary) model" [14], [15], the schemes discussed in [11], [12], [13] are vulnerable to the session key computation attack (as explained in Section III-B). Wang et al. [16] proposed an efficient private comparison protocol based on the additively secret sharing technique, which can be used to realize secure computation of rectified linear activation function (ReLU) without approximation in a semi-honest adversary model.

Later, for secure IoT connectivity, Das et al. [17] suggested a certificate based access-control and session key set-up approach. Their method was designed using the elliptic-curve cryptography (ECC) and a cryptographic hash function. Their approach, on the other hand, does not provide a blockchain-based solution. Finally, Table I shows a summary on various cryptographic techniques used, advantages and limitations of the existing competing schemes.

III. System Models

The system models of a fog based IoT communication environment are discussed in this section.

A. Network Model

The arrangement of network entities of the proposed mechanism is shown in Fig. 1. It can be divided into three layers: a) **end layer** contains all the smart healthcare

TABLE I

TECHNIQUES, ADVANTAGES AND LIMITATIONS OF EXISTING SCHEMES IN IOT ENVIRONMENTS

Scheme	Cryptographic Techniques	Advantages	Drawbacks/Limitations
Luo *et al.* [13]	* ECC * Bilinear pairings * Hash functions * Modular exponentiations	* Mutual authentication * Session key establishment	* Vulnerable to ephemeral secret leakage (ESL) attack under CK-adversary model * High computational cost * Does not support blockchain security solution
Li *et al.* [11]	* Certificateless cryptography (CLC) * Identity-based cryptography (IBC) * Hash functions	* Signcryption * Secure under "indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2) under the gap bilinear Diffie-Hellman (GBDH) problem and existential unforgeability against adaptive chosen messages attacks (EUF-CMA) under the gap Diffie-Hellman (GDH) and computational Diffie-Hellman (CDH) problems"	* No mutual authentication * Does not support blockchain solution
Braeken *et al.* [12]	* Hash functions * Symmetric encryption	* Mutual authentication * Session key establishment	* Vulnerable to ESL attack under the CK-adversary model * Does not support dynamic IoT device addition phase * Does not support blockchain solution
Das *et al.* [17]	* ECC * One-way hash functions * Certificate-based signature	* Mutual authentication * Session key agreement	* Does not support blockchain solution

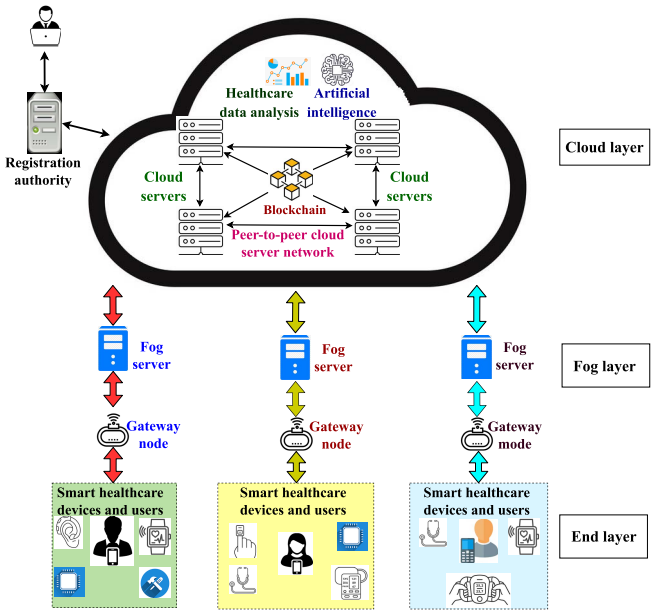

Fig. 1. Network model of AISCM-FH (adapted from [7]).

devices, gateway nodes and users, b) **fog layer** contains the fog servers, and c) **cloud layer** consists of the cloud servers organized in a P2PCS network. A gateway node receives the data from its relevant smart devices and then forwards it to the fog server. After that, the fog server starts preparing the partial blocks with the help of the received data, and later, sends the formed partial blocks to the relevant CS of the P2PCS network. The cloud server in the P2PCS network creates a full block from the received partial block and then calls a consensus process for its addition in the blockchain. The major problem of a fog based IoT communication environment is the existence of various types of attacks. As a result, we require a robust security methods to protect its entire communication [7], [18], [19], [20]. Hence, we aim to design a novel AI-enabled secure communication mechanism in fog computing-based healthcare system. A smart healthcare device can be accessed by a gateway node, and a fog server can be accessed by a gateway node everything in a secure manner. Over the P2PCS network, the healthcare data of a fog computing-based healthcare system is stored in the form of a consortium blockchain. The authorized cloud servers can also perform the task of a health prediction (i.e., possibility of a disease) with the help of collected, processed and stored data. For such tasks, authorized cloud servers can use the various techniques of artificial intelligence (i.e., some machine learning models).

B. Threat Model

To create a safe access control protocol for fog-based IoT communication, the widely recognized Dolev-Yao (DY) threat model is being used [21]. The DY model offers devices to communicate across an unsafe open channel. Smart devices, gateway nodes, fog servers, and cloud servers are examples of terminal devices that cannot be trusted. A potential adversary ($\mathcal{A}$) has the power to eavesdrop, delete, or alter messages that are sent through an insecure channel using the DY model.

The Canetti and Krawczyk's adversary model (CK-adversary model) [14] is another widely-recognized model *de facto* model that is used in the design of the proposed strategy. $\mathcal{A}$ in this model has all the capabilities as the DY model, and in addition, one may gain access to other secrets, such as session states and session keys, which can be used in different sessions.

The gateway nodes are installed under a physical locking system where a physical thievery is not easily possible in this condition [3]. Some smart devices can be physically stolen by $\mathcal{A}$, where through the use of the power analysis attack [22], the compromised smart devices can be utilized to harvest secret credentials from their memory. Other potential attacks, such as "password guessing, impersonation, and unlawful session key computation" attacks can be also launched using the extracted secret data of the compromised devices.

IV. THE PROPOSED SCHEME: AISCM-FH

A secure access control and key management can be achieved among the smart healthcare devices, gateway node(s), fog server(s) and cloud server(s) with the help of the following five important phases, namely i) setup, ii) registration, iii) access control and key management (ACKE), iv) dynamic device addition, v) blockchain implementation and vi) data analytics phase. Various notations and their meanings provided in Table II are used in discussion and analysis of the proposed scheme (AISCM-FH).

TABLE II
NOTATIONS AND THEIR MEANINGS

Notation	Significance
$\mathcal{A}$	An adversary
SD_i and SD_j	i^{th} and j^{th} smart healthcare devices, respectively
GW_k	k^{th} gateway node
FS_l	l^{th} fog server
CS_m	m^{th} cloud server
ID_X, RID_X	Entity X's identity & pseudo identity, respectively
RA, ID_{RA}	Registration authority and its identity, respectively
RID_{RA}	RA's pseudo identity
T_1, T_2, T_3	Various timestamps
ΔT	"Maximum allowed transmission delay"
q	A "sufficiently large prime"
$E_q(\eta, \psi)$	A non-singular elliptic curve
G	A base point in $E_q(\eta, \psi)$
$k.G$	Elliptic curve point (scalar) multiplication, $k \in Z_q^* = \{1, 2, \cdots, q-1\}$ and $k.G = G + G + \cdots + G(k$ times $)$
$h(\cdot)$	A "cryptographic one-way hash function"
SK_{O_1,O_2}	Established session key between entities O_1 and O_2
$\|\|$	Used concatenation operation
$\oplus$	Bitwise XOR operation
(x_O, X_O)	Private key of an entity O, where $X_O = x_O.G$ is corresponding public key of O
mk_X	Master key of an entity X

A. Setup Phase

At this stage, the registration authority (RA) (also known as the trusted authority) chooses important parameters and cryptographic algorithms which will be utilized in the design of the AISCM-FH. The usage of these parameters and methods is mainly necessary in phases, such as "registration, access control, and key agreement, dynamic device addition, and blockchain implementation" phases.

The RA first selects a "non-singular elliptic curve over a Galois (finite) field $GF(q)$ of the type: $E_q(\eta, \psi)$: $y^2 = x^3 + \eta x + \psi$ such that $\eta \in Z_q$ and $\psi \in Z_q$ are two constants with the criteria $4\eta^3 + 27\psi^2 \neq 0 \pmod q$, $q > 3$ is a sufficiently large prime so that the elliptic curve discrete logarithm problem (ECDLP) becomes intractable, $Z_q = \{0, 1, \ldots, q-1\}$ and a point at infinity or zero point as $\mathcal{O}$". The RA then chooses G as a base point in $E_q(\eta, \psi)$ having its order as big as q. Additionally, the RA chooses a "secure one-way hash function $h(\cdot)$ which is collision-resistant in nature" (for example, $h(\cdot)$ can be the "Secure Hash Algorithm (SHA-256) that produces 256 bits hash output on an arbitrary length input").

B. Registration Phase

This phase involves the registration of different devices (i.e., smart healthcare devices (SD_i, SD_j), gateway node GW_k, fog server FS_l, and cloud server (CS_m), which is performed by the fully-trusted registration authority (RA).

1) Registration of Smart Healthcare Devices: The RA does the registration of a smart healthcare device SD_i as per the following steps:

RSD1: The RA selects unique identities ID_{RA} and ID_{SD_i}, unique master keys mk_{RA} and mk_{SD_i} for itself and the SD_i, respectively. The RA then generates unique secret keys $x_{RA} \in Z_q^*$ and $x_{SD_i} \in Z_q^*$, and calculates the public keys as $X_{RA} = x_{RA}.G$ and $X_{SD_i} = x_{SD_i}.G$ for itself and the SD_i, respectively. In addition, the RA generates a certificate random secret key $ck_{SD_i} \in Z_q^*$ and calculates its public key as $CK_{SD_i} = ck_{SD_i}.G$. Furthermore, the RA computes the pseudo identities $RID_{RA} = h(ID_{RA} || mk_{RA})$ and $RID_{SD_i} = h(ID_{SD_i} || mk_{SD_i})$ for itself and for the SD_i, respectively.

RSD2: The RA computes the temporal credential and certificate of SD_i as $TC_{SD_i} = h(ID_{SD_i} || RTS_{SD_i} || RID_{RA} || mk_{SD_i} || mk_{RA} || X_{RA})$ and $CF_{SD_i} = ck_{SD_i} + h(h(RID_{SD_i} || TC_{SD_i}) || X_{SD_i}) * x_{RA} \pmod q$, respectively, where RTS_{SD_i} is the SD_i's registration timestamp. The RA also goes for the generation of a temporary identity and a token identity as TID_{SD_i} and TIN_{SD_i} of the SD_i, respectively.

RSD3: Next, the RA generates a random secret number $n_{SD_i} \in Z_q^*$ for calculating its respective public parameter as $N_{SD_i} = n_{SD_i}.G$. Finally, the RA stores $\{TID_{SD_i}, TIN_{SD_i}, RID_{SD_i}, TC_{SD_i}, CF_{SD_i}, x_{SD_i}, X_{SD_i}, n_{SD_i}, N_{SD_i}, E_q(\eta, \psi), G, h(\cdot)\}$ in the device SD_i's memory. Then, the SD_i will be deployed in the required region. Due to some security issues, the RA performs the deletion of parameters x_{SD_i}, ck_{SD_i}, mk_{SD_i}, n_{SD_i} and RTS_{SD_i} from its database. It then announces the public parameters X_{SD_i}, CK_{SD_i}, X_{RA} and N_{SD_i} to the legitimate entities of the network.

2) Registration of Gateway Node: The RA does the registration of a gateway node GW_k as follows.

RGW1: The RA selects a unique identity ID_{GW_k} and generates a master key mk_{GW_k} for the GW_k. The RA again generates a secret key $x_{GW_k} \in Z_q^*$, and computes the corresponding public key $X_{GW_k} = x_{GW_k}.G$ and the pseudo identity as $RID_{GW_k} = h(ID_{GW_k} || mk_{GW_k})$ for the GW_k. In addition, the RA creates a certificate random secret key $ck_{GW_k} \in Z_q^*$ and calculates its public key as $CK_{GW_k} = ck_{GW_k}.G$.

RGW2: The RA computes a temporal credential and a certificate of GW_k as $TC_{GW_k} = h(ID_{GW_k} || RTS_{GW_k} || RID_{RA} || mk_{GW_k} || mk_{RA} || X_{RA})$ and $CF_{GW_k} = ck_{GW_k} + h(h(RID_{GW_k} || TC_{GW_k}) || X_{GW_k}) * x_{RA} \pmod q$, respectively, where RTS_{GW_k} is GW_k's registration timestamp. The RA also goes for the generation of a temporary identity TID_{GW_k} of GW_k. The RA securely sends the information $\{\{(TIN_{SD_i}, RID_{SD_i}, TC_{SD_i}) | \ i = 1, 2, \cdots, num_{SD}\}, TID_{GW_k}, RID_{GW_k}, TC_{GW_k}, CF_{GW_k}, x_{GW_k}, X_{GW_k}, E_q(\eta, \psi), G, h(\cdot)\}$ to the GW_k, where num_{SD} is the total number of smart devices under the gateway node GW_k.

RGW3: The GW_k stores $\{\{(TIN_{SD_i}, RID_{SD_i}, TC_{SD_i}) | \ i = 1, 2, \cdots, num_{SD}\}, TID_{GW_k}, RID_{GW_k}, TC_{GW_k}, CF_{GW_k}, x_{GW_k}, X_{GW_k}, E_q(\eta, \psi), G, h(\cdot)\}$ in its secure memory. Finally, the gateway node GW_k generates its own random secret number n_{GW_k} and calculates the corresponding public parameter as $N_{GW_k} = n_{GW_k}.G$, and stores (n_{GW_k}, N_{GW_k}) in its secure memory. The GW_k publishes N_{GW_k} as public. The GW_k will be then deployed in the required region. Due to some security issues, the RA performs deletion of parameters x_{GW_k}, ck_{GW_k}, mk_{GW_k} and RTS_{GW_k} from its database. It then proceeds for the announcement of public parameters to the legitimate entities of the network. It is essential to mention that "all gateway nodes are under the physical security in a locking system". It is required to mitigate the malicious task of the physical gateway node stolen attacks.

3) Registration of Fog Server: The RA does the registration of a fog server FS_l as per the following steps:

RFS1: The RA selects a unique identity ID_{FS_l} for the FS_l, generates a master key mk_{FS_l} for FS_l and secret key $x_{FS_l} \in Z_q^*$, and calculates public keys as $X_{FS_l} = x_{FS_l}.G$ for FS_l. RA computes pseudo identity $RID_{FS_l} = h(ID_{FS_l} || mk_{FS_l})$ for FS_l. The RA also creates a certificate random secret key $ck_{FS_l} \in Z_q^*$ and calculates the respective public key as $CK_{FS_l} = ck_{FS_l}.G$.

RFS2: The RA computes a temporal credential of FS_l as $TC_{FS_l} = h(ID_{FS_l} || RTS_{FS_l} || RID_{RA} || mk_{FS_l} ||$

$mk_{RA}||\ X_{RA})$ and certificate of FS_l as $CF_{FS_l} = ck_{FS_l} + h(h(RID_{FS_l}||\ TC_{FS_l})||\ X_{FS_l}) * x_{RA}\ (\text{mod}\ q)$, where RTS_{FS_l} is FS_l's registration timestamp value. The RA securely sends the credentials $\{\{(TID_{GW_k},\ RID_{GW_k},\ TC_{GW_k})|\ i = 1,\ 2, \cdots,\ num_{GW}\},\ RID_{FS_l},\ TC_{FS_l},\ CF_{FS_l},\ x_{FS_l},\ X_{FS_l},\ E_q(\eta, \psi),\ G,\ h(\cdot)\}$ to FS_l, where num_{GW} is the total number of gateway nodes under a fog server FS_l.

RFS3: FS_l stores $\{\{(TID_{GW_k},\ RID_{GW_k},\ TC_{GW_k})|\ i = 1,\ 2, \cdots,\ num_{GW}\},\ RID_{FS_l},\ TC_{FS_l},\ CF_{FS_l},\ x_{FS_l},\ X_{FS_l},\ E_q(\eta, \psi),\ G,\ h(\cdot)\}$ in its secured region of database. The FS_l also proceeds to generate its own random secret $n_{FS_l} \in Z_q^*$ and calculates the public key as $N_{FS_l} = n_{FS_l}.G$ to store (n_{FS_l}, N_{FS_l}) in its secure database. The FS_l publishes N_{FS_l} as public. Due to some security reasons, the RA performs the deletion of parameters x_{FS_l}, ck_{FS_l}, mk_{FS_l} and RTS_{FS_l} from its database. It then announces the public parameters to the legitimate entities of the network. It is worth noting that the database's secured area is utilized to combat hacking attempts against the database's secret stored values. Thus, other associated attacks are prevented, such as the use of a stolen verifier, impersonation, guessing secret credentials, and so on.

4) Registration of Cloud Server: The RA does the registration of a cloud server CS_m by executing the following steps:

RCS1: The RA first picks a unique identity ID_{CS_m} for the CS_m. Next, the RA generates a master key mk_{CS_m} and the pseudo identity $RID_{CS_m} = h(ID_{CS_m}||\ mk_{CS_m})$ for CS_m. The RA then securely sends $\{RID_{CS_m},\ E_q(\eta, \psi),\ G,\ h(\cdot)\}$ to CS_m.

RCS2: The CS_m stores the received information in its secured region of database. It also generates its own secret key $x_{CS_m} \in Z_q^*$, estimates the corresponding public key as $X_{CS_m} = x_{CS_m}.G$, stores (x_{CS_m}, X_{CS_m}) in its secure database and publishes X_{CS_m} as public. To mitigate the security problems, the RA proceeds for the deletion of mk_{CS_m} from its own database. Further, it makes the announcement of public values to different legitimate entities. The sensitive values are stored in the secured region of the database to mitigate the potential hacking attempts against them. It further prevents attacks like "stolen verifier, impersonation and secret credentials guessing" attacks.

All the above discussed registration phases of different network entities are summarized in Fig. 2.

Registration of Smart Healthcare Devices SD_i
Store the credentials $\{TID_{SD_i}, TIN_{SD_i}, RID_{SD_i}, TC_{SD_i}, CF_{SD_i}, (x_{SD_i}, X_{SD_i}), (n_{SD_i}, N_{SD_i}), E_q(\eta, \psi), G, h(\cdot)\}$ in its memory.
Registration of Gateway Node GW_k
Stores the credentials $\{\{(TIN_{SD_i}, RID_{SD_i}, TC_{SD_i})\| i = 1, 2, \cdots, num_{SD}\}, TID_{GW_k}, RID_{GW_k}, TC_{GW_k}, CF_{GW_k}, (x_{GW_k}, X_{GW_k}), (n_{GW_k}, N_{GW_k}), E_q(\eta, \psi), G, h(\cdot)\}$ in its secure database.
Registration of Fog Server FS_l
Stores the credentials $\{\{(TID_{GW_k}, RID_{GW_k}, TC_{GW_k})\| i = 1, 2, \cdots, num_{GW}\}, RID_{FS_l}, TC_{FS_l}, CF_{FS_l}, (x_{FS_l}, X_{FS_l}), (n_{FS_l}, N_{FS_l}), E_q(\eta, \psi), G, h(\cdot)\}$ in its secure database.
Registration of Cloud Server CS_m
Stores the credentials $\{RID_{CS_m}, (x_{CS_m}, X_{CS_m}), E_q(\eta, \psi), G, h(\cdot)\}$ in its secure database.

Fig. 2. Summary of the registration phases.

C. ACKE Phase

A smart device can access other smart devices in a secure way by using the steps of the following presented ACKE process. Similar procedures are followed for the secure access control between the smart device and the gateway node, and also between the gateway node and the fog server.

1) ACKE Between Smart Healthcare Devices: A legitimate smart device SD_i can access another legitimate smart healthcare device SD_j securely by making the use of the following steps:

ACSD1: The process starts by the SD_i with the generation of a random nonce $r_{SD_i} \in Z_q^*$ and the current timestamp T_1. The SD_i computes $M_1 = r_{SD_i} \oplus h(TID_{SD_i}||\ x_{SD_i}.\ X_{SD_j}||\ T_1)$, $M_2 = h(RID_{SD_i}||\ TC_{SD_i}) \oplus h(r_{SD_i}||\ T_1||\ X_{SD_i})$ and $R_{SD_i} = h(CF_{SD_i}||\ T_1||r_{SD_i}||\ x_{SD_i}).G$. Furthermore, the SD_i generates its signature as $SG_{SD_i} = h(CF_{SD_i}||\ T_1||\ r_{SD_i}||\ x_{SD_i}) + h(h(RID_{SD_i}||\ TC_{SD_i})||\ r_{SD_i}||\ x_{SD_i}.X_{SD_j}||\ T_1) * n_{SD_i}\ (\text{mod}\ q)$. On the completion of these estimations, the SD_i sends the message $MS_1 = \{TID_{SD_i}, M_1, M_2, R_{SD_i}, CF_{SD_i}, SG_{SD_i}, T_1\}$ to the SD_j using an open (insecure) channel.

ACSD2: Upon the arrival of the MS_1 from the SD_i, the SD_j first performs the verification of the timestamp T_1 through the condition, $|T_1 - T_1^*| \leq \Delta T$, where "$\Delta T$ denotes the maximum transmission delay and T_1^* denotes the message MS_1's receiving time." If the criterion is met, the SD_j calculates the $r_{SD_i} = M_1 \oplus h(TID_{SD_i}||\ x_{SD_j}.\ X_{SD_i}||\ T_1)$, $h(RID_{SD_i}||\ TC_{SD_i}) = M_2 \oplus h(r_{SD_i}||\ T_1||\ X_{SD_i})$. Then, the SD_j verifies the certificate of the SD_i through $CF_{SD_i}.G = CK_{SD_i} + h(h(RID_{SD_i}||\ TC_{SD_i})||\ X_{SD_i}).\ X_{RA}$. The matching in both sides of this equation proves the genuineness of the certificate CF_{SD_i} of the SD_i. After this SD_j proceeds for the signature verification of SD_i through $SG_{SD_i}.G = R_{SD_i} + h(h(RID_{SD_i}||\ TC_{SD_i})||\ r_{SD_i}||\ x_{SD_j}.\ X_{SD_i}).\ N_{SD_i}$. If it holds, SG_{SD_i} of SD_i is successfully verified by the SD_j.

ACSD3: The smart healthcare device SD_j generates a random nonce $r_{SD_j} \in Z_q^*$ and current timestamp T_2. Then, the SD_j computes $M_3 = r_{SD_j} \oplus h(TID_{SD_j}||\ x_{SD_j}.\ X_{SD_i}||\ T_2)$, $M_4 = h(RID_{SD_j}||\ TC_{SD_j}) \oplus h(r_{SD_j}||\ T_2||\ X_{SD_j})$ and $R_{SD_j} = h(CF_{SD_j}||\ T_2||r_{SD_j}||\ x_{SD_j}).G$. After this SD_j computes session key $SK_{SD_j,SD_i} = h(h(RID_{SD_i}||\ TC_{SD_i})||\ h(RID_{SD_j}||\ TC_{SD_j})||\ r_{SD_i}||\ r_{SD_j}||\ CF_{SD_i}||\ CF_{SD_j}||\ x_{SD_j}.\ X_{SD_i}||\ T_1||\ T_2)$. Further, the SD_j generates its signature as $SG_{SD_j} = h(CF_{SD_j}||\ T_2||\ r_{SD_j}||\ x_{SD_j}) + h(SK_{SD_j,SD_i}||\ T_1||\ T_2) * n_{SD_j}\ (\text{mod}\ q)$. On the completion of these estimations, the SD_j sends a message $MS_2 = \{TID_{SD_j}, M_3, M_4, R_{SD_j}, CF_{SD_j}, SG_{SD_j}, T_2\}$ to the SD_i using an insecure (open) channel.

ACSD4: Upon the arrival of MS_2 from SD_j, SD_i first performs the verification of timestamp T_2 through the condition, $|T_2 - T_2^*| \leq \Delta T$, where T_2^* denotes the message MS_2's receiving time. If the criterion is met, the SD_i calculates $r_{SD_j} = M_3 \oplus h(TID_{SD_j}||\ x_{SD_j}.\ X_{SD_i}||\ T_2)$ and $h(RID_{SD_j}||\ TC_{SD_j}) = M_4 \oplus h(r_{SD_j}||\ T_2||\ X_{SD_j})$. Then, the SD_i verifies the certificate of SD_j through $CF_{SD_j}.G = CK_{SD_j} + h(h(RID_{SD_j}||\ TC_{SD_j})||\ X_{SD_j}).\ X_{RA}$. The matching in both sides of this equation proves the genuineness of the certificate CF_{SD_j} of SD_j. Next, SD_i computes the session key $SK_{SD_i,SD_j} = h(h(RID_{SD_i}||\ TC_{SD_i})||\ h(RID_{SD_j}||\ TC_{SD_j})||\ r_{SD_i}||\ r_{SD_j}||\ CF_{SD_i}||\ CF_{SD_j}||\ x_{SD_i}.\ X_{SD_j}||\ T_1||\ T_2)$ shared with SD_j. After this, the SD_i proceeds

SD_i	SD_j
Generate $r_{SD_i} \in Z_q^*$, T_1 Compute $M_1 = r_{SD_i} \oplus h(TID_{SD_i} \| x_{SD_i}.X_{SD_j} \| T_1)$, $M_2 = h(RID_{SD_i} \| TC_{SD_i}) \oplus h(r_{SD_i} \| T_1 \| X_{SD_i})$, $R_{SD_i} = h(CF_{SD_i} \| T_1 \| r_{SD_i} \| x_{SD_i}).G$, $SG_{SD_i} = h(CF_{SD_i} \| T_1 \| r_{SD_i} \| x_{SD_i}) + h(h(RID_{SD_i} \| TC_{SD_i}) \| r_{SD_i} \| x_{SD_i}.X_{SD_j} \| T_1) * n_{SD_i} \pmod q$. $MS_1 = \{TID_{SD_i}, M_1, M_2, R_{SD_i}, CF_{SD_i}, SG_{SD_i}, T_1\}$ $\xrightarrow{\text{(via open channel)}}$	Check $\lvert T_1 - T_1^* \rvert \le \Delta T$. If so, compute r_{SD_i} $= M_1 \oplus h(TID_{SD_i} \| x_{SD_j}.X_{SD_i} \| T_1)$, $h(RID_{SD_i} \| TC_{SD_i}) = M_2 \oplus h(r_{SD_i} \| T_1 \| X_{SD_i})$. Verify if $CF_{SD_i}.G = CK_{SD_i} + h(h(RID_{SD_i} \| TC_{SD_i}) \| X_{SD_i}).X_{RA}$? If so, verify $SG_{SD_i}.G = R_{SD_i} + h(h(RID_{SD_i} \| TC_{SD_i}) \| r_{SD_i} \| x_{SD_j}.X_{SD_i}).N_{SD_i}$. If it holds, generate $r_{SD_j} \in Z_q^*$, T_2. Compute $M_3 = r_{SD_j} \oplus h(TID_{SD_j} \| x_{SD_j}.X_{SD_i} \| T_2)$, $M_4 = h(RID_{SD_j} \| TC_{SD_j}) \oplus h(r_{SD_j} \| T_2 \| X_{SD_j})$, $R_{SD_j} = h(CF_{SD_j} \| T_2 \| r_{SD_j} \| x_{SD_j}).G$, $SK_{SD_j,SD_i} = h(h(RID_{SD_i} \| TC_{SD_i}) \| h(RID_{SD_j} \| TC_{SD_j}) \| r_{SD_i} \| r_{SD_j} \| CF_{SD_i} \| CF_{SD_j} \| x_{SD_j}.X_{SD_i} \| T_1 \| T_2)$. $SG_{SD_j} = h(CF_{SD_j} \| T_2 \| r_{SD_j} \| x_{SD_j}) + h(SK_{SD_j,SD_i} \| T_1 \| T_2) * n_{SD_j} \pmod q$. $MS_2 = \{TID_{SD_j}, M_3, M_4, R_{SD_j}, CF_{SD_j}, SG_{SD_j}, T_2\}$ $\xleftarrow{\text{(via open channel)}}$
Check $\lvert T_2 - T_2^* \rvert \le \Delta T$. If so, compute $r_{SD_j} = M_3 \oplus h(TID_{SD_j} \| x_{SD_j}.X_{SD_i} \| T_2)$, $h(RID_{SD_j} \| TC_{SD_j}) = M_4 \oplus h(r_{SD_j} \| T_2 \| X_{SD_j})$. Verify $CF_{SD_j}.G = CK_{SD_j} + h(h(RID_{SD_j} \| TC_{SD_j}) \| X_{SD_j}).X_{RA}$. If so, compute $SK_{SD_i,SD_j} = h(h(RID_{SD_i} \| TC_{SD_i}) \| h(RID_{SD_j} \| TC_{SD_j}) \| r_{SD_i} \| r_{SD_j} \| CF_{SD_i} \| CF_{SD_j} \| x_{SD_i}.X_{SD_j} \| T_1 \| T_2)$. Verify $SG_{SD_j}.G = R_{SD_j} + h(SK_{SD_i,SD_j} \| T_1 \| T_2).N_{SD_j}$. If so, generate T_3 and compute $M_5 = h(SK_{SD_i,SD_j} \| T_3)$ $MS_3 = \{M_5, T_3\}$ $\xrightarrow{\text{(via open channel)}}$	Check $\lvert T_3 - T_3^* \rvert \le \Delta T$. compute $M_5' = h(SK_{SD_j,SD_i} \| T_3)$ Check if $M_5' = M_5$? If it matches, then session key is legitimate.
Both SD_i and SD_j store $SK_{SD_j,SD_i} = SK_{SD_i,SD_j}$	

Fig. 3. ACKE phase between SD_i and SD_j.

for the signature verification of the SD_j through $SG_{SD_j}.G = R_{SD_j} + h(SK_{SD_i,SD_j} \| T_1 \| T_2).N_{SD_j}$. If it holds, the SG_{SD_j} of the SD_j is also successfully checked by the SD_i. The SD_i generates another timestamp T_3, computes a session key verifier as $M_5 = h(SK_{SD_i,SD_j} \| T_3)$ and sends the message $MS_3 = \{M_5, T_3\}$ to the SD_j through an insecure (open) channel.

ACSD5: When the message MS_3 is received, the SD_j first performs the verification of timestamp T_3 through the condition, $\lvert T_3 - T_3^* \rvert \le \Delta T$, where T_3^* denotes the message MS_3's receiving time. If the criterion is met, the SD_j calculates $M_5' = h(SK_{SD_j,SD_i} \| T_3)$ and checks if $M_5' = M_5$? If it matches, it affirms that the session key computed by the SD_i is correct. Then, both the SD_i and the SD_j establish $SK_{SD_j,SD_i} = SK_{SD_i,SD_j}$ for their secure communications.

The above discussed steps of the access control and the key establishment phase between the SD_i and the SD_j are summarized in Fig. 3.

2) ACKE Between Smart Healthcare Devices and Gateway Node: This phase explains the procedure of secure access control between a legitimate smart device SD_i and its associated gateway node GW_k. The following steps needs to be executed:

ACSG1: The smart device SD_i being the initiator starts with the generation of a random nonce $rn_{SD_i} \in Z_q^*$ and a fresh timestamp TS_1. The SD_i estimates $M_1 = rn_{SD_i} \oplus h(RID_{SD_i} \| x_{SD_i}.X_{SD_j} \| TS_1)$ and $RN_{SD_i} = h(TC_{SD_i} \| CF_{SD_i} \| TS_1 \| rn_{SD_i} \| x_{SD_i}).G$. Moreover, the SD_i generates its signature as $SIG_{SD_i} = h(TC_{SD_i} \| CF_{SD_i} \| TS_1 \| rn_{SD_i} \| x_{SD_i}) + h(RID_{SD_i} \| rn_{SD_i} \| x_{SD_i}.X_{GW_k} \| TS_1) * n_{SD_i} \pmod q$. On the completion of these estimations, the SD_i sends the message $MSG_1 = \{TIN_{SD_i}, M_1, RN_{SD_i}, CF_{SD_i}, SIG_{SD_i}, TS_1\}$ to GW_k through an insecure (open) channel.

ACSG2: Upon the arrival of MSG_1 from the SD_i, the GW_k first performs the verification of timestamp TS_1 through the condition, $\lvert TS_1 - TS_1^* \rvert \le \Delta T$. Here, the TS_1^* is the time when the message MSG_1 was received. If the condition holds, the GW_k retrieves RID_{SD_i} and TC_{SD_i} corresponding to the received TIN_{SD_i} from its memory. The GW_k computes the $rn_{SD_i} = M_1 \oplus h(RID_{SD_i} \| x_{GW_k}.X_{SD_i} \| TS_1)$. Next, the GW_k verifies the certificate of SD_i through $CF_{SD_i}.G = CK_{SD_i} + h(h(RID_{SD_i} \| TC_{SD_i}) \| X_{SD_i}).X_{RA}$. The matching proves the genuineness of the certificate CF_{SD_i} of the SD_i. After this, the GW_k proceeds for the signature verification of SD_i through $SIG_{SD_i}.G = RN_{SD_i} + h(RID_{SD_i} \| rn_{SD_i} \| x_{GW_k}.X_{SD_i} \| TS_1).N_{SD_i}$. If it holds it means that the SIG_{SD_i} of the SD_i is successfully checked by the GW_k.

ACSG3: The GW_k generates a random nonce $rn_{GW_k} \in Z_q^*$ and a fresh timestamp TS_2. The GW_k estimates $M_2 = rn_{GW_k} \oplus h(RID_{SD_i} \| x_{GW_k}.X_{SD_i} \| TS_2)$, $M_3 = h(RID_{GW_k} \| TC_{GW_k}) \oplus h(rn_{GW_k} \| TS_2 \| X_{GW_k})$ and $RN_{GW_k} = h(CF_{GW_k} \| TS_2 \| rn_{GW_k} \| x_{GW_k}).G$, and also the session key $SK_{GW_k,SD_i} = h(h(RID_{SD_i} \| TC_{SD_i}) \| h(RID_{GW_k} \| TC_{GW_k}) \| rn_{SD_i} \| rn_{GW_k} \| CF_{SD_i} \| CF_{GW_k} \| x_{GW_k}.X_{SD_i} \| TS_1 \| TS_2)$. Furthermore, the GW_k generates its signature as $SIG_{GW_k} = h(CF_{GW_k} \| TS_2 \| rn_{GW_k} \| x_{GW_k}) + h(SK_{GW_k,SD_i} \| TS_1 \| TS_2) * n_{GW_k} \pmod q$. Again, the GW_k generates a new token identity $TIN_{SD_i}^{new}$ of SD_i and computes $M_4 = TIN_{SD_i}^{new} \oplus h(TC_{SD_i} \| h(RID_{GW_k} \| TC_{GW_k}) \| TS_1 \| TS_2)$. On the completion of these estimations, the GW_k sends the message $MSG_2 = \{M_2, M_3, M_4, RN_{GW_k}, CF_{GW_k}, SIG_{GW_k}, TS_2\}$ to SD_i via the open channel.

ACSG4: Upon the arrival of MSG_2 from the GW_k, the SD_i first performs the verification of timestamp TS_2 through the condition: $\lvert TS_2 - TS_2^* \rvert \le \Delta T$, where TS_2^* denotes the message MSG_2's receiving time. If the criterion is met, the SD_i calculates the $rn_{GW_k} = M_2 \oplus h(RID_{SD_i} \| x_{SD_i}.X_{GW_k} \| TS_2)$ and $h(RID_{GW_k} \| TC_{GW_k}) = M_3 \oplus h(rn_{GW_k} \| TS_2 \| X_{GW_k})$. The SD_i verifies the certificate of GW_k through $CF_{GW_k}.G = CK_{GW_k} + h(h(RID_{GW_k} \| TC_{GW_k}) \| X_{GW_k}).X_{RA}$. The matching ensures the genuineness of the certificate CF_{GW_k} of the GW_k. Next, the SD_i computes the session

key $SK_{SD_i,GW_k} = h(h(RID_{SD_i}||\ TC_{SD_i})||\ h(RID_{GW_k}||\ TC_{GW_k})||\ rn_{SD_i}||\ rn_{GW_k}||\ CF_{SD_i}||\ CF_{GW_k}||\ x_{SD_i}.\ X_{GW_k}||\ TS_1||\ TS_2)$. After this SD_i proceeds for the signature verification of the GW_k through $SIG_{GW_k}.G = RN_{GW_k} + h(SK_{SD_i,GW_k}||\ TS_1||\ TS_2).N_{GW_k}$. If it holds, the SIG_{GW_k} of the GW_k is successfully verified by the SD_i. SD_i computes a new token identity through $TIN_{SD_i}^{new} = M_4 \oplus h(TC_{SD_i}||\ h(RID_{GW_k}||\ TC_{GW_k})||\ TS_1||\ TS_2)$, and replaces the TIN_{SD_i} with the $TIN_{SD_i}^{new}$ in its memory. The SD_i generates another timestamp TS_3 and computes a session key verifier $M_5 = h(SK_{SD_i,GW_k}||\ TIN_{SD_i}^{new}||\ TS_3)$ and sends the message $MSG_3 = \{M_5,\ TS_3\}$ to GW_k through an insecure (open) channel.

ACSG5: When the MSG_3 is received, the GW_k first verifies the timestamp TS_3. If the condition holds, the GW_k computes $M_5' = h(SK_{GW_k,SD_i}||\ TIN_{SD_i}^{new}||\ TS_3)$ and checks if $M_5' = M_5$? If it matches, the GW_k agrees that the session key computed by the SD_i is correct and it has also updated its new taken identity successfully. Both the SD_i and the GW_k establish $SK_{GW_k,SD_i} = SK_{SD_i,GW_k}$ for their secure communications.

Remark 1 (Protection against the synchronisation attack): For protection of the synchronisation attack, one can assume a scenario where the communicated message MSG_2 *may be tampered or it could be due to a communication error in the channel. Because of this issue, a smart device* SD_i *may not receive the parameter* M_4 *including the new token identity* $TIN_{SD_i}^{new}$*. To sort this problem out, one can follow a method that is discussed in [23], where a smart device requires to keep a set of* l *shadow identities, say* $SID = \{s_{id_1}, s_{id_2}, \cdots, s_{id_l}\}$*. In worst cases, when the smart device* SD_i *can not get the message* MSG_2 *within a pre-specific time interval, it requires to choose one of the shadow identities that are used so far, say* $s_{id_i} \in SID$*. It can send this identity within the message* MSG_1*. Once the gateway* GW_k *receives the* s_{id_i}*, it can create a new token identity and dispatch it to* SD_i *through a secure channel. Thus, the synchronization attacks are mitigated using the steps of aforementioned mechanism in the proposed approach. The same mechanism can be also useful in our proposed scheme to prevent the synchronisation attack between the* GW_k *and the* FS_l*.*

3) ACKE Between the Gateway Node and the Fog Server: This phase explains the procedure of a secure access control between a legitimate GW_k and its associated fog server FS_l. The following steps need to be executed:

ACGF1: The process starts by the gateway node GW_k with the generation of a random nonce value $rc_{GW_k} \in Z_q^*$ and the current timestamp t_1. The GW_k computes $M_1 = rc_{GW_k} \oplus h(RID_{GW_k}||\ x_{GW_k}.\ X_{FS_l}||\ t_1)$ and $RC_{GW_k} = h(TC_{GW_k}||\ CF_{GW_k}||\ t_1||rc_{GW_k}||\ x_{GW_k}).G$. Furthermore, GW_k generates its signature as $\text{sin}_{GW_k} = h(TC_{GW_k}||\ CF_{GW_k}||\ t_1||\ rc_{GW_k}||\ x_{GW_k}) + h(RID_{GW_k}||\ rc_{GW_k}||\ x_{GW_k}.X_{FS_l}||\ t_1) * n_{GW_k} \pmod q$. On the completion of these estimations, the GW_k sends the message $msg_1 = \{TID_{GW_k},\ M_1,\ RC_{GW_k},\ CF_{GW_k},\ \text{sin}_{GW_k},\ t_1\}$ to FS_l through an insecure (open) channel.

ACGF2: Upon the arrival of msg_1 from GW_k, the FS_l first performs the verification of timestamp t_1. If the timestamp is valid, the FS_l retrieves RID_{GW_k} and TC_{GW_k} corresponding to the received TID_{GW_k} from its memory. The FS_l computes $rc_{GW_k} = M_1 \oplus h(RID_{GW_k}||\ x_{FS_l}.\ X_{GW_k}||\ t_1)$. Then, it verifies the certificate of the GW_k through $CF_{GW_k}.G = CK_{GW_k} + h(h(RID_{GW_k}||\ TC_{GW_k})||\ X_{GW_k}).\ X_{RA}$. The matching in both sides of this equation proves the genuineness of the certificate CF_{GW_k} of the GW_k. After this, the FS_l proceeds for the signature verification of the GW_k through $\text{sin}_{GW_k}.G = RC_{GW_k} + h(RID_{GW_k}||\ rc_{GW_k}||\ x_{FS_l}.X_{GW_k}||\ t_1).N_{GW_k}$. If it holds, the sin_{GW_k} of the GW_k is successfully verified by the FS_l.

ACGF3: The FS_l goes for the generation of a random nonce $rc_{FS_l} \in Z_q^*$ with a fresh timestamp t_2. The FS_l estimates $M_2 = rc_{FS_l} \oplus h(RID_{GW_k}||\ x_{FS_l}.\ X_{GW_k}||\ t_2)$, $M_3 = h(RID_{FS_l}||\ TC_{FS_l}) \oplus h(rc_{FS_l}||\ t_2||\ X_{FS_l})$ and $RC_{FS_l} = h(CF_{FS_l}||\ t_2||rc_{FS_l}||\ x_{FS_l}).G$. After these computations, the FS_l computes the session key $SK_{FS_l,GW_k} = h(h(RID_{FS_l}||\ TC_{FS_l})||\ h(RID_{GW_k}||\ TC_{GW_k})||\ rc_{FS_l}||\ rc_{GW_k}||\ CF_{FS_l}||\ CF_{GW_k}||\ x_{FS_l}.\ X_{GW_k}||\ t_1||\ t_2)$. Furthermore, the FS_l generates its signature as $\text{sin}_{FS_l} = h(CF_{FS_l}||\ t_2||\ rc_{FS_l}||\ x_{FS_l}) + h(SK_{FS_l,GW_k}||\ t_1||\ t_2) * n_{FS_l} \pmod q$. Again, the FS_l creates a new temporary identity $TID_{GW_k}^{new}$ of the GW_k to compute $M_4 = TID_{GW_k}^{new} \oplus h(TC_{GW_k}||\ h(RID_{FS_l}||\ TC_{FS_l})||\ t_1||\ t_2)$. On the completion of these estimations, the FS_l sends the message $msg_2 = \{M_2,\ M_3,\ M_4,\ RC_{FS_l},\ CF_{FS_l},\ \text{sin}_{FS_l},\ t_2\}$ to GW_k using an insecure (open) channel.

ACGF4: Upon the arrival of msg_2 from FS_l, the GW_k first performs the verification of timestamp t_2. If the criterion is met, the GW_k calculates $rc_{FS_l} = M_2 \oplus h(RID_{GW_k}||\ x_{GW_k}.\ X_{FS_l}||\ t_2)$ and $h(RID_{FS_l}||\ TC_{FS_l}) = M_3 \oplus h(rc_{FS_l}||\ t_2||\ X_{FS_l})$. The GW_k verifies the certificate of FS_l through $CF_{FS_l}.G = CK_{FS_l} + h(h(RID_{FS_l}||\ TC_{FS_l})||\ X_{FS_l}).\ X_{RA}$. The matching in both sides of this equation proves the genuineness of the certificate CF_{FS_l} of FS_l. After this, the GW_k computes the session key $SK_{GW_k,FS_l} = h(h(RID_{GW_k}||\ TC_{GW_k})||\ h(RID_{FS_l}||\ TC_{FS_l})||\ rc_{GW_k}||\ rc_{FS_l}||\ CF_{GW_k}||\ CF_{FS_l}||\ x_{GW_k}.\ X_{FS_l}||\ t_1||\ t_2)$. Next, the GW_k proceeds for the signature verification of FS_l through $\text{sin}_{FS_l}.G = RC_{FS_l} + h(SK_{GW_k,FS_l}||\ t_1||\ t_2).N_{FS_l}$. If it holds, it means that the sin_{FS_l} of the FS_l is successfully checked by the GW_k. The GW_k also derives its new temporary identity through $TID_{GW_k}^{new} = M_4 \oplus h(TC_{GW_k}||\ h(RID_{FS_l}||\ TC_{FS_l})||\ t_1||\ t_2)$ and replaces TID_{GW_k} with $TID_{GW_k}^{new}$. The GW_k generates another timestamp t_3 and computes a session key verifier $M_5 = h(SK_{GW_k,FS_l}||\ TID_{GW_k}^{new}||\ t_3)$ and sends the message $msg_3 = \{M_5,\ t_3\}$ to the FS_l through the insecure (open) channel.

ACGF5: Upon the arrival of msg_3 from the GW_k, the FS_l performs the verification of timestamp t_3. If the criterion is met, the FS_l calculates $M_5' = h(SK_{FS_l,GW_k}||\ TID_{GW_k}^{new}||\ t_3)$ and checks if $M_5' = M_5$? If it matches, the FS_l agrees that the session key computed by the GW_k is correct and it has also updated its new temporary identity successfully. Thus, both FS_l and GW_k establish $SK_{FS_l,GW_k} = SK_{GW_k,FS_l}$ for their secure communication.

Remark 2: FS_l *and* CS_m *are resource-rich devices in the network. For secure communication,* FS_l *and* CS_m *can employ their ECC-based private-public key pairs.*

D. Dynamic Device Addition Phase

The facility of new devices' addition is essential to provide in an access control or authentication scheme. In the proposed AISCM-FH, new devices like smart devices or gateway nodes can be added in the network. The RA can add a new smart device SD_i^{new} in the network using the following steps:

DASD1: The RA selects the identity $ID_{SD_i}^{new}$ for SD_i^{new}, generates a unique master key $mk_{SD_i}^{new}$ for SD_i^{new}, a secret key $x_{SD_i}^{new} \in Z_q^*$ and its corresponding public key $X_{SD_i}^{new} = x_{SD_i}^{new}.G$ for SD_i^{new}. The RA then generates its pseudo-identity as

Block Header	
Identity of block	BID
Hash of previous block	H_{PB}
Merkle Tree Root	MTR
Timestamp	TS
Owner of Block	OB
Public Key of Owner	X_{FS_l}
Block Payload (Encrypted Transactions)	
Encrypted Transaction #1 Encrypted Transaction #2 $\vdots$ Encrypted Transaction $\#n_t$	$E_{X_{FS_l}}(Tx_1)$ $E_{X_{FS_l}}(Tx_2)$ $\vdots$ $E_{Pub_{FS_l}}(Tx_{n_t})$
Current Block Hash Signature on Block using ECDSA	H_{CB} SIG_{BLK}

Fig. 4. Layout of a block.

$RID_{SD_i}^{new} = h(ID_{SD_i}^{new} || mk_{SD_i}^{new})$. In addition, the RA generates a certificate random secret key $ck_{SD_i}^{new} \in Z_q^*$ and calculates its public key as $CK_{SD_i}^{new} = ck_{SD_i}^{new}.G$, and publishes $CK_{SD_i}^{new}$ as public.

DASD2: The RA computes the temporal credential and certificate as $TC_{SD_i}^{new} = h(ID_{SD_i}^{new} || RTS_{SD_i}^{new} || RID_{RA} || mk_{SD_i}^{new} || mk_{RA} || X_{RA})$, $CF_{SD_i}^{new} = ck_{SD_i}^{new} + h(h(RID_{SD_i}^{new} || TC_{SD_i}^{new}) || X_{SD_i}^{new}) * x_{RA} \pmod q$, respectively, for SD_i^{new}, where $RTS_{SD_i}^{new}$ is SD_i^{new}'s registration timestamp. The RA also generates a temporary identity and a token identity $TID_{SD_i}^{new}$, $TIN_{SD_i}^{new}$ for SD_i^{new}.

DASD3: The RA generates a random secret number $n_{SD_i}^{new}$ and its corresponding public parameter as $N_{SD_i}^{new} = n_{SD_i}^{new}.G$. Finally, the RA stores the credentials $\{TID_{SD_i}^{new}, TIN_{SD_i}^{new}, RID_{SD_i}^{new}, TC_{SD_i}^{new}, CF_{SD_i}^{new}, (x_{SD_i}^{new}, X_{SD_i}^{new}), (n_{SD_i}^{new}, N_{SD_i}^{new}), E_q(\eta, \psi), G, h(\cdot)\}$ in the memory of SD_i^{new}. SD_i^{new} can be deployed in the required region of the network. Due to some security issues, the RA does the deletion of values $x_{SD_i}^{new}$, $ck_{SD_i}^{new}$, $mk_{SD_i}^{new}$, $n_{SD_i}^{new}$ and $RTS_{SD_i}^{new}$ from its database. The RA announces the public values to the parties publicly. It also securely communicates the SD_i^{new}'s registration information to the existing associated gateway node. In a similar way, the addition of new gateway nodes can be done in the network.

E. Blockchain Implementation Phase

In this step, we give the details of a block creation and its inclusion in the blockchain. For this purpose, the following procedure is required to execute. The layout of a block of the blockchain is given in Table 4.

BI1: When a fog server, say FS_l, receives data from its corresponding gateway node GW_k securely through the established session key, it converts the received information in the form of n_t transactions and encrypts them through its own public key to create $\alpha_{ETx_i} = E_{X_{FS_l}}(Tx_i))$, where $i = 1, 2, \cdots, n_t$. Then, the FS_l computes the Merkle tree root $MTR_{\alpha_{ETx}}$ on the encrypted transactions α_{ETx_i} by building the corresponding Merkle tree. After that, the FS_l creates a partial block that contains information like the public key of owner X_{FS_l}, information of owner of the block OB, Merkle tree root $MTR_{\alpha_{ETx}}$ and encrypted transactions α_{ETx_i}. The FS_l sends the constructed partial blocks to the corresponding cloud server CS_m by encrypting these using the public key X_{CS_m} of the CS_m.

BI2: Upon the arrival of the partial block at the cloud server, say CS_m, it makes the full blocks (say, BLK_i) from them by introducing other essential data fields such as identity of the block, previous block's hash, timestamp value, current block's hash and signature on the block using the signature generation of "elliptic curve digital signature algorithm (ECDSA)." After the that, a full block BLK_i is passed to other mining entities of the peer-to-peer cloud servers (P2P CS) network. The averment and inclusion of BLK_i will be performed with the help of the selected consensus algorithm. In case, BLK_i is received by the P2P CS network, a miner will be selected as the leader (L) from the P2PCS network through the steps of leader selection technique [24]. In the proposed AISCM-FH, we use the "Ripple Protocol Consensus Algorithm (RPCA)" [25] for the purpose of block's averment and inclusion via a voting method. Note that each cloud server CS_m in the P2P-CS network has a pair of ECC-enabled private-public keys (x_{CS}, X_{CS}), where $X_{CS} = x_{CS} \cdot G$. After the execution of all the steps described in Algorithm 1, the concerned block will be prosperously included in the blockchain.

F. Data Analytics Phase

The data analysis of the collected and processed digital healthcare data is performed in this phase through the machine learning models. The data analysis occurs at the authorized cloud servers of the P2PCS network. This procedure is required to draw useful decision making from the collected healthcare data (for example, predictions of getting a heart attack, chances of getting early diabetes, etc.).

DAP1: The healthcare data collected through the smart devices in sensitive nature and it should not revealed to updated by any adversary in any case. For the secure exchange of information in between the gateway nodes and fog servers, and between devices, the aforementioned steps of access control and key establishment can be utilized. For the secure communication of fog server FS_l and cloud server CS_m, the FS_l can send the computed partial block to CS_m in a secure way by encrypting it using the public key X_{CS_m} of CS_m. The partial blocks contain the health related data in the form of encrypted transactions.

DAP2: For the data analysis purpose, an authorized cloud server collects data from all fog servers and then performs the data aggregation over that. For such needs, the fog servers can provide the data to the authorized cloud server CS_m in the decrypted form in a secure way using the public key X_{CS_m} of CS_m. After the data aggregation, the other essential steps like data analysis, data visualization and prediction are performed over the data. For the data analysis and prediction purpose, various machine learning models under classification, clustering, or deep learning can be used. The final outcomes will be in the form of some useful results as discussed earlier [20], [26].

V. Security Analysis

A. Formal Security Analysis

We prove the security of the proposed AISCM-FH through the "standard Real-Or-Random (ROR) model" [15]. We use the imperative properties like one-way hash function and "Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP)".

Definition 1 (Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP)): Given a "quadruple $(P, l_1.P, l_2.P, l_3.P)$ *where* $P \in E_q(\eta, \psi)$ *is a point on a non-singular elliptic curve* $E_p(a, b)$*, determine if* $l_3 = l_1 l_2$ *or a uniform value", where* $l_1, l_2, l_3 \in Z_q^*$.

Algorithm 1 Block Averment and Inclusion

1: Assume that a leader node L is elected from the peer nodes of P2PCS network.
2: Let L have a full block BLK_i that is ready for consensus process.
3: L sets $NV = 0$, where NV denotes the counter of valid votes.
4: L sets $flag_{CS_l} = 0$, for all $l = 1, 2, \cdots, n_{cs}$ with $L \neq CS_m$, where n_{cs} is the total number of cloud servers.
5: L procreates a random nonce RN_l and a fresh timestamp t_l.
6: L creates the encrypted messages for other cloud servers (CS_m) as $E_{X_{CS_m}}(RN_l, t_l)$.
7: L broadcasts message $bmg_l = \{BLK_i, E_{X_{CS_m}}(RN_l, t_l), t_l\}$ to other miner nodes.
8: Upon the arrival of bmg_l, each peer CS_m verifies the validity of timestamp t_l.
9: **if** (t_l verified successfully) **then**
10: Merkle tree root MTR^* on encrypted transactions $\alpha_{ETx_i} = E_{X_{FS_l}}(Tx_i))$, where $i = 1, 2, \cdots, n_t$, is computed.
11: **if** ($MTR^* \neq MTR$) **then**
12: Consensus is terminated.
13: **end if**
14: Received block's hash value $H(BLK_i)^*$ is computed.
15: **if** ($H(BLK_i)^* = H(BLK_i)$) **then**
16: CS_m performs verification of signature on BLK_i.
17: **if** (signature verification is successful) **then**
18: CS_m creates $rbmg_l$ as voting reply message including RN_l and sends securely to the leader L.
19: **for** each valid $rbmg_l$ received from peer nodes **do**
20: L sets $NV = NV + 1$ and $flag_{CS_l} = 1$.
21: **if** (NV is less than pre-defined opted threshold value, i.e., 75% NV) **then**
22: L continues the process.
23: **else**
24: L sends commitment of addition of the BLK_i to other peer nodes.
25: BLK_i is added into the the blockchain BC_{HS}.
26: **end if**
27: **end for**
28: **end if**
29: **end if**
30: **end if**

The ECDDHP proves to be computationally infeasible if q is chosen as a large prime number. For the intractability of ECDDHP, q should be selected at least 160-bit of prime as the essential requirement.

The semantic security of the AISCM-FH is proven through the ROR model. Under the standard procedure of ROR model, we prove that the AISCM-FH is able to provide the session key security (SK-security). We first discuss the ROR model and then prove the SK-security of the proposed AISCM-FH in Theorem 1. According to ROR model, an attacker $\mathcal{A}$ is connected with the t^{th} instance of the working participant, i.e., the communicating party as $\mathcal{P}^t$. In AISCM-FH, the participants are devices SD_i/SD_j or GW_k, FS_l, or CS_m that are assumed as $\mathcal{P}^t$. Let $\mathcal{P}_{SD_i}^{t_1}$ and $\mathcal{P}_{SD_j}^{t_2}$ express the t_1^{th} and t_2^{th} instances of smart devices SD_i and SD_j, respectively. Similar expressions can be written for the communications happen in between SD_i and GW_k, GW_k and FS_l, and FS_l and CS_m. Table III discusses the various queries like *Execute*, *Reveal* and *Test*. A "one-way hash function $h(\cdot)$" is modeled as a random oracle, say $Hash$. $Hash$ is accessible to all the legitimate parties including the adversary $\mathcal{A}$.

TABLE III
VARIOUS QUERIES AND THEIR PURPOSES

Query	Explanation
$Execute$	$\mathcal{A}$ can intercept the messages exchanged between two entities through this query.
$Reveal$	This query can be executed for the revealing of the calculated session key between $\mathcal{P}^t$ and its partner to the adversary $\mathcal{A}$.
$Test$	$\mathcal{A}$ argues $\mathcal{P}^t$ for the calculated session key and $\mathcal{P}^t$ answers with a probabilistic outcome of a flipped unbiased coin c.

To prove Theorem 1, we first discuss Lemma 1.

Lemma 1 (Difference Lemma): Assume that we have three events E_1, E_2 and E_3 that are defined in some probability distribution. If $E_1 \wedge \neg E_3 \Leftrightarrow E2 \wedge \neg E_3$, $|Pr[E_1] - Pr[E_2]| \leq Pr[E_3]$, where $Pr[E]$ denotes the probability of an event E.

Theorem 1: Let $\mathcal{A}$ run in polynomial time t against the proposed scheme (AISCM-FH), and q_{hash}, $|Hash|$ and $Adv_{\mathcal{A}}^{ECDDHP}(t)$ represent the "number of hash queries", the "range space of one-way hash function $h(\cdot)$" and "$\mathcal{A}$'s advantage in breaking ECDDHP", respectively. Let $Adv_{\mathcal{A}}^{AISCM-FH}(t)$ denotes $\mathcal{A}$'s advantage for breaking of the semantic security of AISCM-FH in time t to the illegal obtaining of session key SK_{SD_i,SD_j} between the participants SD_i and SD_j during the access control process discussed in Section IV-C1. Then,

$$Adv_{\mathcal{A}}^{AISCM-FH}(t) \leq \frac{q_{hash}^2}{|Hash|} + 2\ Adv_{\mathcal{A}}^{ECDDHP}(t).$$

Proof: This theorem has a similar proof as explained in other schemes [3], [27]. We assume the three games, say GM_j, $j \in [0, 2]$, where $Succ_{\mathcal{A}}^{GM_j}$ represents an event wherein $\mathcal{A}$ has a guessing of a random bit c in GM_j correctly. The advantage of $\mathcal{A}$ in winning the game GM_j is given by $Adv_{\mathcal{A},GM_j}^{AISCM-FH} = Pr[Succ_{\mathcal{A}}^{GM_j}]$. The details of these games GM_j, $j \in [0, 2]$ are as follows.

Game GM_0: GM_0 is a real attack performed by $\mathcal{A}$ against the proposed AISCM-FH under the ROR model. Initially, in GM_0, a bit c is selected in a random way. The following result is obtained using the semantic security of the AISCM-FH as

$$Adv_{\mathcal{A}}^{AISCM-FH}(t) = |2.Adv_{\mathcal{A},GM_0}^{AISCM-FH} - 1| \quad (1)$$

Game GM_1: This game is corresponding to an eavesdropping attack in which $\mathcal{A}$ has the capability to intercept all the communicated messages $MS_1 = \{TID_{SD_i}, M_1, M_2, R_{SD_i}, CF_{SD_i}, SG_{SD_i}, T_1\}$, $MS_2 = \{TID_{SD_j}, M_3, M_4, R_{SD_j}, CF_{SD_j}, SG_{SD_j}, T_2\}$ and $MS_3 = \{M_5, T_3\}$ during the access control procedure (Section IV-C1) through the $Execute$ query provided in Table III. At the end of GM_1, $\mathcal{A}$ needs to execute the $Reveal$ and $Test$ queries to check if the derived session key SK_{SD_i,SD_j} between devices SD_i and SD_j is real

or random. It is essential to highlight that the session key established in between the SD_i and SD_j is $SK_{SD_j,SD_i} = h(h(RID_{SD_i}||\ TC_{SD_i})||\ h(RID_{SD_j}||\ TC_{SD_j})||\ r_{SD_i}||\ r_{SD_j}||\ CF_{SD_i}||\ CF_{SD_j}||\ x_{SD_j}.\ X_{SD_i}||\ T_1||\ T_2)$. To calculate the session key, $\mathcal{A}$ should know the short term secret values like the random secrets as well as the "long term secrets like different pseudo identities, temporal credentials and secret keys". Therefore, it is observed that only eavesdropping of the messages MS_1, MS_2 and MS_3 is not helping $\mathcal{A}$ for increasing the winning probability of the game GM_1. The following relation is then obtained using the fact that both games GM_0 and GM_1 are indistinguishable:

$$Adv_{\mathcal{A},GM_1}^{AISCM-FH} = Adv_{\mathcal{A},GM_0}^{AISCM-FH} \tag{2}$$

Game GM_2: Here, the $Hash$ query is used in the modeling of an active attack. In the messages MS_1, MS_2 and MS_3, the secret information is covered under one-way hash function $H(\cdot)$. Furthermore, the taping of $X_{SD_i} = x_{BS_i}.G$ is not helpful for $\mathcal{A}$ because it is a "computationally infeasible problem" for $\mathcal{A}$ to deduce the private key x_{SD_i} due to "intractability property of ECDDHP" (Definition 1). Other related parameters are also secured. Therefore, $\mathcal{A}$ should have a clear knowledge of these discussed parameters for the correct computation of the session key SK_{SD_i,SD_j}. It is a difficult task for $\mathcal{A}$ as it has to solve ECDDHP in polynomial time t. Besides the computations of the secret parameters used in messages MS_1, MS_2 and MS_3 are difficult, even if MS_1, MS_2 and MS_3 are eavesdropped by $\mathcal{A}$. The different random secret numbers, distinct identities and different secret keys are used in all exchanged messages among various entities of the network. It is an important observation that games GM_1 and GM_2 are indistinguishable except the admittance of the simulation of the $Hash$ query in GM_2. After applying the results obtained via "birthday paradox and intractability of ECDDHP," and also the difference lemma defined in Definition 1, the following result is obtained:

$$\begin{aligned} &|Adv_{\mathcal{A},GM_1}^{AISCM-FH} - Adv_{\mathcal{A},GM_2}^{AISCM-FH}| \\ &\quad \leq \frac{q_{hash}^2}{2|Hash|} + Adv_{\mathcal{A}}^{ECDDHP}(t) \end{aligned} \tag{3}$$

The various queries are now simulated by $\mathcal{A}$. $\mathcal{A}$ is remaining with the guessing of bit c in game GM_2. It then follows that

$$Adv_{\mathcal{A},GM_2}^{AISCM-FH} = \frac{1}{2} \tag{4}$$

Eqs. (1), (2) and (4) provide

$$\begin{aligned} \frac{1}{2}.Adv_{\mathcal{A}}^{AISCM-FH}(t) &= |Adv_{\mathcal{A},GM_0}^{AISCM-FH} - \frac{1}{2}| \\ &= |Adv_{\mathcal{A},GM_1}^{AISCM-FH} - Adv_{\mathcal{A},GM_2}^{AISCM-FH}| \end{aligned} \tag{5}$$

Eqs. (3) and (5) give

$$\frac{1}{2}.Adv_{\mathcal{A}}^{AISCM-FH}(t) \leq \frac{q_{hash}^2}{2|Hash|} + Adv_{\mathcal{A}}^{ECDDHP}(t) \tag{6}$$

Simplifying Eq. (6), we finally arrive to the final result: $Adv_{\mathcal{A}}^{AISCM-FH}(t) \leq \frac{q_{hash}^2}{|Hash|} + 2\ Adv_{\mathcal{A}}^{ECDDHP}(t)$. □

Remark 3: Like Theorem 1, one can also prove the session key security of the proposed AISCM-FH under the ROR model against the session keys derivation during the ACKE phase between a smart device and a gateway node, and also the ACKE phase between a gateway node and a fog server.

B. Informal Security Analysis

1) Replay Attack: In AISCM-FH, we have used freshly generated timestamp values which are verified by the receiving parties upon the arrival of the messages. If the timestamp validation condition holds, the received message is considered to be fresh; otherwise, the receiving party discards the message. Thus, an adversary $\mathcal{A}$ is not able to replay the old messages. Hence, AISCM-FH is robust against replay attacks.

2) Man-in-the-Middle (MiTM) Attack: Let there be an adversary $\mathcal{A}$, that eavesdrops on the exchanged messages and then modifies them in order to send them to a different communicating party for launching of the MiTM attack. Suppose $\mathcal{A}$ generates a temporary identity $TID_{SD_i}^a$, a timestamp T_1^a and a random secret $r_{SD_i}^a$. Next, $\mathcal{A}$ tries to calculate $M_1^a = r_{SD_i}^a \oplus h(TID_{SD_i}^a||\ x_{SD_i}.\ X_{SD_j}||\ T_1^a)$, $M_2 = h(RID_{SD_i}||\ TC_{SD_i}) \oplus h(r_{SD_i}^a||\ T_1^a||\ X_{SD_i})$, $R_{SD_i}^a = h(CF_{SD_i}||\ T_1^a||r_{SD_i}^a||\ x_{SD_i}).G$ and signature $SG_{SD_i}^a = h(CF_{SD_i}||\ T_1^a||\ r_{SD_i}^a||\ x_{SD_i}) + h(h(RID_{SD_i}||\ TC_{SD_i})||\ r_{SD_i}^a||\ x_{SD_i}.X_{SD_j}||\ T_1^a) * n_{SD_i} \pmod q$. Here, $X_{SD_i} = x_{SD_i}.G$, $RID_{RA} = h(ID_{RA}||\ mk_{RA})$, $RID_{SD_i} = h(ID_{SD_i}||\ mk_{SD_i})$, $TC_{SD_i} = h(ID_{SD_i}||\ RTS_{SD_i}||\ RID_{RA}||\ mk_{SD_i}||\ mk_{RA}||\ X_{RA})$ and RTS_{SD_i} is the SD_i's registration timestamp. On the completion of these estimations, $\mathcal{A}$ tries to create the message $MS_1^a = \{TID_{SD_i}^a,\ M_1^a,\ M_2^a,\ R_{SD_i}^a,\ CF_{SD_i},\ SG_{SD_i}^a,\ T_1^a\}$ so that it can send this message to other legitimate smart device SD_j using an open (insecure) channel. However, $\mathcal{A}$ can not create a legitimate MS_1 because it does not know the secret values x_{SD_i}, n_{SD_i}, mk_{SD_i}, mk_{RA}, RTS_{SD_i}, RID_{SD_i} and RID_{TA}. Therefore, $\mathcal{A}$ can not modify MS_1 or any other transmitted messages. Thus, $\mathcal{A}$ can not launch the MiTM attack.

3) Impersonation Attacks: In such a malicious act, an active adversary $\mathcal{A}$ tries to impersonate like a genuine entity of the network. Let there be an adversary $\mathcal{A}$, that tries to create some messages in order to send them to another communicating party in order to launch an impersonation attack. Suppose $\mathcal{A}$ generates a temporary identity $TID_{SD_i}^I$, a timestamp T_1^I and a random secret $r_{SD_i}^I$. Then, $\mathcal{A}$ tries to calculate $M_1^I = r_{SD_i}^I \oplus h(TID_{SD_i}^I||\ x_{SD_i}.\ X_{SD_j}||\ T_1^I)$, $M_2 = h(RID_{SD_i}||\ TC_{SD_i}) \oplus h(r_{SD_i}^I||\ T_1^I||\ X_{SD_i})$ and $R_{SD_i}^I = h(CF_{SD_i}||\ T_1^I||r_{SD_i}^I||\ x_{SD_i}).G$ and a signature $SG_{SD_i}^I = h(CF_{SD_i}||\ T_1^I||\ r_{SD_i}^I||\ x_{SD_i}) + h(h(RID_{SD_i}||\ TC_{SD_i})||\ r_{SD_i}^I||\ x_{SD_i}.X_{SD_j}||\ T_1^I) * n_{SD_i} \pmod q$. Here, $X_{SD_i} = x_{SD_i}.G$, $RID_{RA} = h(ID_{RA}||\ mk_{RA})$, $RID_{SD_i} = h(ID_{SD_i}||\ mk_{SD_i})$, $TC_{SD_i} = h(ID_{SD_i}||\ RTS_{SD_i}||\ RID_{RA}||\ mk_{SD_i}||\ mk_{RA}||\ X_{RA})$ and RTS_{SD_i} is the SD_i's registration timestamp. Now, $\mathcal{A}$ tries to create the message $MS_1^I = \{TID_{SD_i}^I,\ M_1^I,\ M_2^I,\ R_{SD_i}^I,\ CF_{SD_i},\ SG_{SD_i}^I,\ T_1^I\}$ on behalf of a legitimate smart device SD_i. However, $\mathcal{A}$ can not create the exact value of MS_1 because it does not know the secret credentials. In a similar argument, $\mathcal{A}$ can not also create other valid messages on behalf of the other entities. As a result, $\mathcal{A}$ can not launch an impersonation attack.

4) Anonymity: The exchanged messages in between the smart devices and gateway node, and in between the "gateway node and the fog server" used only the freshly created timestamps and random secrets. Any identity is not transmitted in plaintext in any of the transmitted messages. In addition to this, a temporary/token identity is used in AISCM-FH, which gets updated in each session. Hence, the AISCM-FH supports the anonymity property.

5) Ephemeral Secret Leakage (ESL) Attack: In the AISCM-FH, a session key between entities like SD_i and SD_j

is calculated as $SK_{SD_j,SD_i} = h(h(RID_{SD_i}||\ TC_{SD_i})||\ h(RID_{SD_j}||\ TC_{SD_j})||\ r_{SD_i}||\ r_{SD_j}||\ CF_{SD_i}||\ CF_{SD_j}||\ x_{SD_j}.\ X_{SD_i}||\ T_1||\ T_2)$, where $X_{SD_i} = x_{SD_i}.G$, $RID_{RA} = h(ID_{RA}||\ mk_{RA})$, $RID_{SD_i} = h(ID_{SD_i}||\ mk_{SD_i})$, $TC_{SD_i} = h(ID_{SD_i}||\ RTS_{SD_i}||\ RID_{RA}||\ mk_{SD_i}||\ mk_{RA}||\ X_{RA})$, $CF_{SD_i} = ck_{SD_i} + h(h(RID_{SD_i}||\ TC_{SD_i})||\ X_{SD_i}) * x_{RA} \pmod q$. The session keys in between the other entities in the network also involve the long-term secrets, such as the secret keys and various real/pseudo identities, and short-term secrets, such as random secrets. Furthermore, a new session key is established for each new session. $\mathcal{A}$ is unaware of these secret credentials. As a result, $\mathcal{A}$ will be unable to calculate the session keys as it requires both types of secrets (short-term and long-term secret credentials). This shows that the AISCM-FH is capable of thwarting an ESL attack under the CK-adversary model.

6) Synchronization and Associated Attacks: In the AISCM-FH, messages like $MSG_1 = \{TIN_{SD_i},\ M_1,\ RN_{SD_i},\ CF_{SD_i},\ SIG_{SD_i},\ TS_1\}$, $MSG_2 = \{M_2,\ M_3,\ M_4,\ RN_{GW_k},\ CF_{GW_k},\ SIG_{GW_k},\ TS_2\}$ and $MSG_3 = \{M_5,\ TS_3\}$ are exchanged between the smart device and the gateway node. During this exchange, the last message MSG_3 contains M_5, and TS_3, where $M_5 = h(SK_{SD_i,GW_k}||\ TIN^{new}_{SD_i}||\ TS_3)$. The use of this mechanism forces the GW_k to cross check the updated value of the token identity $TIN^{new}_{SD_i}$. If condition $M'_5 = M_5$ holds at the GW_k's end, then it is considered that SD_i updates the value of its token identity correctly. During the communication between the gateway node and the fog server, a similar mechanism can be used to update the temporary identity of the gateway node. Furthermore, all of the network's heterogeneous devices, i.e., "smart healthcare devices, gateway nodes, fog servers, and cloud servers, are considered to be synced with their clocks." For the purpose of resolving the synchronisation problem, these entities also commit on "maximum transmission delay (ΔT)." Apart from that, we have used other techniques for the mitigation of the synchronisation problem in between the entities.

7) Stolen Verifier Attack: In the proposed AISCM-FH, the registration information of SD_j, GW_k, FS_l, and CS_m is stored in their secure memory. For instance, the FS_l stores $\{\{(TID_{GW_k},\ RID_{GW_k},\ TC_{GW_k})|\ i = 1, 2, \cdots, num_{GW}\}, RID_{FS_l}, TC_{FS_l}, CF_{FS_l}, x_{FS_l}, X_{FS_l}, n_{FS_l}, N_{FS_l}, E_q(\eta, \psi), G, h(\cdot)\}$ in a secured region of its database. Here, it is important to mention that the same mechanism is also used in cases of other traditional cryptosystems like RSA, ECC and AES based secure communication systems, where it is presumed that all sensitive secret keys are kept in the database's safe area (memory). Unauthorized access to the confidential information is then prevented. As a result, in AISCM-FH, $\mathcal{A}$ does not have access to useful information for launching additional potential attacks like impersonation, and unlawful session key computation attacks. The gateway nodes are installed under the physical locking system. Thus, their physical stealing is not possible. Therefore, $\mathcal{A}$ can not launch other attacks which are related to the physical stealing of gateway nodes. As a result, the AISCM-FH has the potential to stop a stolen verifier attack.

8) Privileged-Insider Attack: There is a possibility that a trusted authority's privileged insider user will have access to entities SD_i, GW_k, FS_l and CS_m registration information. However, because the insider user does not know the secret values such as secret keys, registration timestamps, etc. after the registration process is over, it cannot launch other related attacks like impersonation, unlawful session key computation attacks on AISCM-FH. This shows that AISCM-FH can protect against privileged insider attack.

9) Physical Device Capture Attack: Let a potential adversary $\mathcal{A}$ physically capture a smart healthcare device and further tries to launch other associated attacks on the proposed AISCM-FH, like unauthorized session key computation attack. However, execution of such malicious task is not feasible on AISCM-FH as the session keys are calculated with the use of different random and secret values. This mechanism produces different session keys in all distinct sessions for different entities. Therefore, the physically capturing of a smart device may cause the revealing of all the credentials stored in its memory using the power analysis attacks [22]. It is worth noticing that the secrets given to each smart device are completely distinct and unique. Thus, the compromised secret credentials will not help in compromising the secret credentials stored in other smart devices. This implies that using a physically captured smart device, adversary $\mathcal{A}$ can not perform other attacks like the impersonation attack on behalf of the other non-compromised smart devices. This property is known as the *unconditional security against device capture attack* [28]. Hence, the AISCM-FH can mitigate the physical device capture attack.

10) Mitigation of 51% Attack and Selfish Mining: It is already revealed that 51% attack and selfish mining attack are possible on the blockchain based networks. Such attacks may occur in case if $\mathcal{A}$ has a large amount of hashing power [29]. The 51% attack needs that $\mathcal{A}$ should carry more than half of the hashing power. Particularly, 51% attack may be conducted against the cryptocurrencies in which $\mathcal{A}$ performs unauthorised tasks, like double-spending. Furthermore, the selfish mining is another well discovered security flaw in the blockchain based networks, which can be executed by malicious miners to theft block rewards. Recent exploitation proves that the "Proof-of-Work (PoW) consensus algorithm" is unsafe against the 51% attack. In the proposed AISCM-FH, the voting-based "Ripple Protocol Consensus Algorithm (RPCA)" has been integrated, which is secured against these attacks. Thus, the proposed AISCM-FH is safe against 51% attack and selfish mining.

11) Other Potential Attacks Through Blockchain: The AISCM-FH can help to protect against other types of threats. To thwart potential hacking efforts, the entities like FS_l and CS_m store confidential information in their secured region of their databases. The AISCM-FH employs a blockchain-based method, which makes it more secure and tamper-proof. As a result, it can defend against a variety of attacks, including denial-of-service (DoS) attack, and various sorts of data poisoning, alterating and leaking attacks.

VI. Comparative Study

In this section, we provide the comparisions among the proposed AISCM-FH and other related existing schemes, such as the schemes of Luo et al. [13], Li et al. [11], Braeken et al. [12] and Das et al. [17] in terms of communication costs, security & functionality features, and computation costs. In the AISCM-FH, we consider the following three cases:

- **Case I.** ACKE phase between two smart healthcare devices SD_i and SD_j
- **Case II.** ACKE phase between a smart healthcare device SD_i and its associated gateway node GW_k
- **Case III.** ACKE phase between a gateway node GW_k and its associated fog server FS_l

TABLE IV
COMPARATIVE STUDY – COMMUNICATION COSTS

Protocol	No. of messages	Total cost (in bits)
AISCM-FH (Case I)	3	2496
AISCM-FH (Case II)	3	2336
AISCM-FH (Case III)	3	2336
Luo *et al.* [13]	2	3040
Li *et al.* [11]	2	3488
Braeken *et al.* [12]	3	3552
Das *et al.* [17]	3	3296

A. Communication Costs Comparison

For the communication costs comparison and analysis, we assume the sizes of an "identity is 160 bits", a "random number is 160 bits", the "hash output when SHA-1 technique is considered [30] is 160 bits", and a "timestamp is 32 bits". An elliptic curve point, say $P = (P_x, P_y)$, with P_x and P_y representing the x and y co-ordinates, respectively, requires $(160 + 160) = 320$ bits. This is considered by assuming the fact that "the security provided by an 160-bit ECC is almost the same as that for an 1024-bit RSA-based public key cryptosystem" [31].

In the AISCM-FH, messages like $MS_1 = \{TID_{SD_i}, M_1, M_2, R_{SD_i}, CF_{SD_i}, SG_{SD_i}, T_1\}$, $MS_2 = \{TID_{SD_j}, M_3, M_4, R_{SD_j}, CF_{SD_j}, SG_{SD_j}, T_2\}$ and $MS_3 = \{M_5, T_3\}$ are exchanged between the smart devices SD_i and SD_j which requires the sizes of MS_1 as 1152 bits, MS_2 as 1152 bits and MS_3 as 192 bits. The total communication cost of the AISCM-FH in Case I is then $1152 + 1152 + 192 = 2496$ bits. Similarly, for Case II and Case III, the communication costs involved in AISCM-FH are respectively 2336 and 2336 bits for an exchange of three messages. The comparison of communication costs of different schemes shown in Table IV concludes that the total communication costs required for the schemes of Luo et al. [13], Li et al. [11], Braeken et al. [12] and Das et al. [17] are 3040 bits, 3488 bits, 3552 bits and 3296 bits, respectively. It is important to highlight that the proposed AISCM-FH needs less communication cost as compared to other existing techniques.

B. Computation Costs Comparison

We use the following symbols in the calculations of computation costs for the proposed AISCM-FH and other compared schemes: T_h – the time for executing a "one-way cryptographic hash function," T_{ecm} – the time for executing an "elliptic curve point (scalar) multiplication," T_{eca} – the time for executing an "elliptic curve point addition," T_{sed} – the time for executing a "symmetric encryption/decryption function". In the ACKE phase between two smart devices of AISCM-FH, a smart device SD_i requires $7T_{ecm} + 11T_h + 2T_{eca}$ computation cost whereas its neighbor smart device SD_j needs $6T_{ecm} + 11T_h + 2T_{eca}$ computation cost. Thus, the maximum computation required for a smart device for the ACKE phase in Case I is $7T_{ecm} + 11T_h + 2T_{eca}$. Similarly, for the other ACKE phases of AISCM-FH (Case II and Case III) are also computed and tabulated in Table V.

For the estimation of rough computation time (in milliseconds), we apply the experimental results available in [32]: "$T_{ecm} \approx 13.405$ ms, $T_{eca} \approx 0.081$ ms, $T_h \approx 0.056$ ms, $T_{bp} \approx 32.713$ ms, $T_{me} \approx 2.249$ ms". It is assumed that $T_{sed} \approx T_h$. Under this environment, the computation cost in the AISCM-FH for a smart device under Case I becomes $7T_{ecm} + 10T_h \approx 94.395$ ms. The computation costs for the AISCM-FH and other schemes are listed in Table V. It is worth to note that the AISCM-FH requires less computation cost in Case I as compared with other schemes, such as the schemes of Luo et al. [13], Li et al. [11] and Das et al. [17], for the smart devices. On the other hand, the scheme of Braeken et al. [12] has less computation cost than the proposed AISCM-FH. It is because Braeken et al.'s scheme [12] uses a symmetric encryption/decryption and hash function computations. Braeken's method, on the other hand, lacks essential security and functionality attributes as stated in Table VI as compared to those for the proposed AISCM-FH.

TABLE V
COMPARATIVE STUDY – COMPUTATION COSTS

Protocol	Smart (sensing) device cost	Total cost	Rough estimation (in milliseconds)
AISCM-FH (Case I)	$7T_{ecm} + 11T_h + 2T_{eca}$ or $6T_{ecm} + 10T_h + 2T_{eca}$	$7T_{ecm} + 11T_h + 2T_{eca}$	94.613
AISCM-FH (Case II)	$10T_h + 9T_{ecm} + 2T_{eca}$	$23T_h + 17T_{ecm} + 4T_{eca}$	229.497
AISCM-FH (Case III)	$10T_h + 6T_{ecm} + 2T_{eca}$	$21T_h + 12T_{ecm} + 4T_{eca}$	162.36
Luo *et al.* [13]	$T_{bp} + T_h$	$3T_{ecm} + 4T_{bp} + 4T_h + T_{eca} + T_{me}$	173.621
Li *et al.* [11]	$T_{bp} + T_h$	$3T_{ecm} + 5T_{bp} + 2T_h + 2T_{eca}$	204.054
Braeken *et al.* [12]	$11T_h + T_{sed}$	$23T_h + 2T_{sed}$	1.400
Das *et al.* [17]	$7T_{ecm} + T_h + 3T_{eca}$	$7T_{ecm} + 6T_h + 3T_{eca}$	94.414

C. Security and Functionality Attributes Comparison

A comparison of functionality and security attributes of the proposed AISCM-FH and other related schemes is given in Table VI. It is essential to highlight that the other existing methods [11], [12], [13], [17] do not support or preserve the attributes like ϕS_3, ϕS_9, ϕS_{10} and ϕS_{11}. The existing compared schemes do not also provide "blockchain based security." On the other hand, AISCM-FH is based on blockchain and supports all essential functionality & security features ranging from ϕS_1 to ϕS_{11}.

VII. PRACTICAL DEMONSTRATION OF AISCM-FH

A. Blockchain Simulation Setup and Results

The details of different parameters utilized for simulation of AISCM-FH are given in Table VII. The three different scenarios for example, Scenario-1 (Σc-1), Scenario-2 (Σc-2) and Scenario-3 (Σc-3) were taken in the experimentation. The "Windows 10 64-bit OS with Intel (R) core i5-8250U, 1.60 GHz-1.80 GHz processor" platform was used to perform the experiments. The "eclipse IDE 2019-12 with Java language" programming environment was considered for the performing the coding part. In the experimentation, we have taken 50, 100 and 150 smart devices for Σc-1, Σc-2, and Σc-3, respectively. Moreover, we have taken 5, 10 and 15 gateway nodes for Σc-1, Σc-2, and Σc-3, respectively. We have also taken 5, 10 and 15 fog servers for Σc-1, Σc-2, and Σc-3,

TABLE VI

COMPARATIVE STUDY – FUNCTIONALITY & SECURITY ATTRIBUTES

Feature	Luo *et al.* [13]	Li *et al.* [11]	Braeken *et al.* [12]	Das *et al.* [17]	AISCM-FH
ϕS_1	✓	✓	✓	✓	✓
ϕS_2	✓	✓	✓	✓	✓
ϕS_3	×	×	×	✓	✓
ϕS_4	✓	✓	✓	✓	✓
ϕS_5	✓	✓	✓	✓	✓
ϕS_6	✓	✓	✓	✓	✓
ϕS_7	✓	✓	✓	✓	✓
ϕS_8	✓	✓	✓	✓	✓
ϕS_9	×	×	×	✓	✓
ϕS_{10}	×	×	×	✓	✓
ϕS_{11}	×	×	×	×	✓

ϕS_1: "protection against replay attack"; ϕS_2: "protection against man-in-the-middle attack"; ϕS_3: "provides mutual authentication"; ϕS_4: "provides session key agreement"; ϕS_5: "protection against device impersonation attack"; ϕS_6: "protection against malicious device deployment attack"; ϕS_7: "resilience against device physical capture attack"; ϕS_8: "formal security analysis"; ϕS_9: "requires involvement of gateway node during the authentication (access control) phase"; ϕS_{10}: "ESL attack under the CK-adversary model"; ϕS_{11}: "blockchain based security".
✓: "a scheme is secure or it supports a functionality feature"; ×: "a scheme is insecure or it does not support a functionality feature".

TABLE VII

DETAILS OF BLOCKCHAIN IMPLEMENTATION PARAMETERS

Parameter	Value
Considered platform	Windows 10 OS with 64-bit
Processor utilized	Intel (R) core (TM), i5-8250U, 1.60 GHz-1.80 GHz
Configuration of random-access memory (RAM)	8 GB
Considered coding environment	Java with Eclipse IDE 2019-12
Scenarios taken	Scenario-1 (Σc-1), Scenario-2 (Σc-2), Scenario-3 (Σc-3)
Number of smart devices	50 (in Σc-1), 100 (in Σc-2), 150 (in Σc-3)
Number of gateway nodes	5 (in Σc-1), 10 (in Σc-2), 15 (in Σc-3)
Number of fog servers	5 (in Σc-1), 10 (in Σc-2), 15 (in Σc-3)
Considered miner nodes	4 in all scenarios
Estimated size of a block	65,632 bits

respectively. Moreover, four miner nodes (cloud servers) were considered. The "voting based technique along with the steps of RPCA consensus algorithm" were adopted in the blockchain mining. A leader L was elected which initiates the consensus process as per the steps of Algorithm 1 for the addition of a block BLK_i in the blockchain BC_{HS} after it is committed by the other miner nodes.

A block has following different fields:

* *Block version:* This field contains information about a block's version. We interpreted it as 32 bits.

* *Previous block's hash value:* It carries the previous block's hash value information. We assumed it was 256 bits (in case SHA256 hash algorithm was taken).

* *Merkle tree root:* On encrypted transactions, this value is calculated. We assumed it was 256 bits (in case of SHA-256 hash algorithm).

* *Timestamp:* It contains information regarding the timestamp. We interpreted it as 32-bits.

* *Block owner:* It contains the block owner's details. We interpreted it as 160-bits.

* *Public key of owner:* It holds the owner's public key information. We interpreted it as 320-bits (under the consideration of ECC).

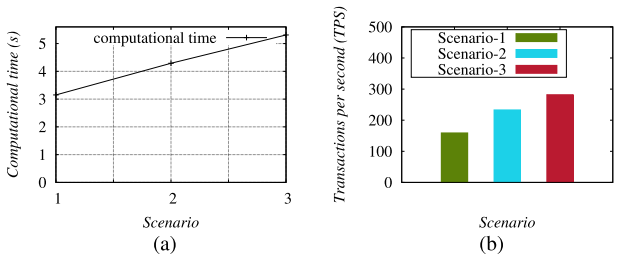

Fig. 5. Results on (a) computational time in seconds (b) transactions per second (TPS).

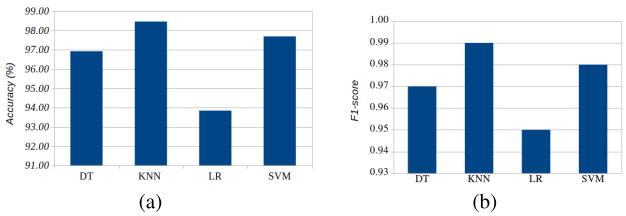

Fig. 6. Results under different methods (a) accuracy (b) F1-score.

* *Digital data in the form of encrypted transactions:* It contains information about current transactions. It is worth noting that the size of each encrypted transaction is determined by the ECC-based ciphertext. As a result, it requires $(320 + 320) = 640$ bits. The payload of the block was determined as $(100 \times 640) = 64,000$ bits, which is the sum of 100 encrypted transactions.

* *Current block's hash value:* It contains information about the current block's hash value. We assumed it was 256-bits (for SHA256 hash algorithm).

* *Block's signature:* It holds the signature data for a specific block. We interpreted it as 320-bits (under the consideration of ECC algorithm).

1) Analysis on Computational Time: The influence of an increasing number of smart devices and gateway nodes on the performance of the AISCM-FH should be measured. We calculated and studied the computation time (in seconds) for the three cases we considered: Scenario-1 (Σc-1), Scenario-2 (Σc-2) and Scenario-3 (Σc-3). The computation times for Σc-1, Σc-2, and Σc-3 are 3.15 seconds, 4.29 seconds, and 5.31 seconds, respectively. Fig. 5(a) shows a similar set of data. It is important to note that the computation cost rises as the number of smart healthcare devices and gateway nodes increases from Σc-1 to Σc-2 and Σc-2 to Σc-3, respectively, because the increase in these devices leads to the creation and addition (mining) of more blocks in the blockchain.

2) Analysis on Transactions per Second (TPS): We calculated and analysed the impact on transactions per second (TPS) based on the numerous situations studied. The TPS values for Σc-1, Σc-2 and Σc-3 are 159, 233, and 282, respectively. Fig. 5(b) shows that the value of a transaction per second (TPS) increases as the number of smart healthcare devices and gateway nodes increases from Σc-1 to Σc-2 and Σc-2 to Σc-3, respectively, because the increase in these devices causes the production and addition (mining) of more blocks in the blockchain.

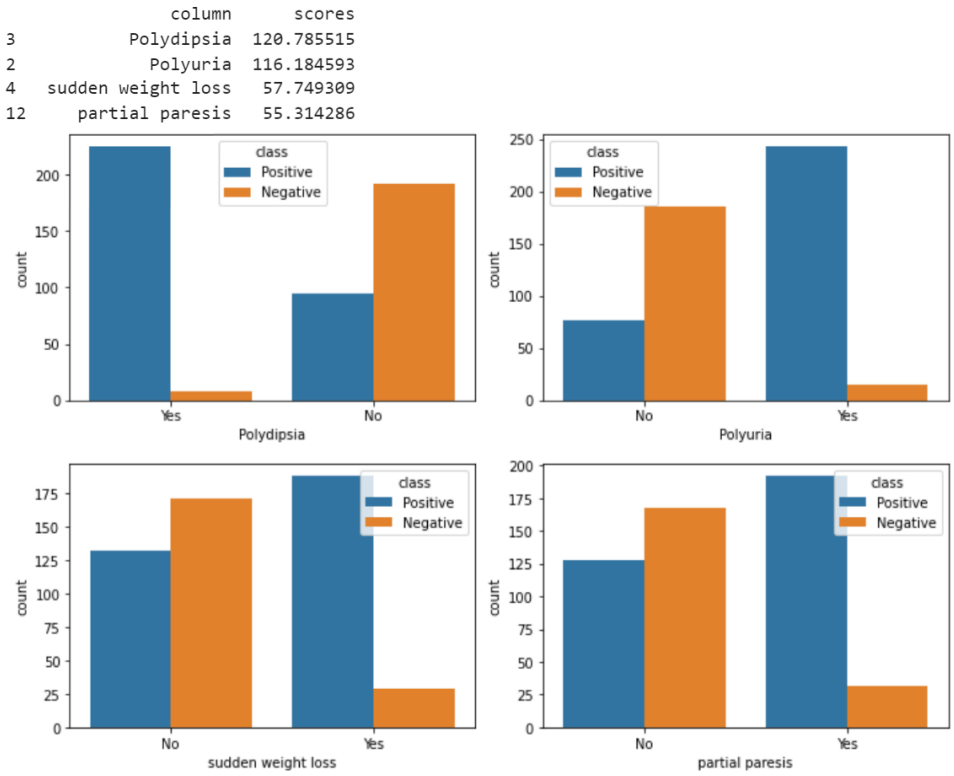

Fig. 7. Factors that can cause high possibility of diabetes.

B. Big Data Analytics Experiments and Results

In this section, we provide the details of data analytics phase, which is done through the machine learning (ML) models.

For the data analysis part, we have utilized "early stage diabetes risk prediction dataset," which is available on the "UCI machine learning repository" [33], [34]. The data analysis is essential to draw important decision making from the collected, processed and analysed health related data. To deal with this, we have applied the four most related classifiers: 1) "Decision Tree (DT)," 2) "K-nearest neighbors (KNN)," 3) "Logistic Regression (LR)" and 4) "Support Vector Machine (SVM)," for the estimation of various important parameters, like accuracy and F1-score. The following specifications are considered.

Hardware specifications: We have used "$i5$ 11^{th} generation processor along with 8 GB RAM", having 2 GB Nvidia MX 450 graphic card and 1TB HDD/256 Gb SSD".

Platforms and libraries utilized: We have used the "Google colab platform." Moreover, the "pandas library" has been used for importing and reading the data. Furthermore, "seaborne and matplotlib libraries" are considered for the "data visualization." In addition, "scikit learn library" has been used for "data pre-processing to perform ML based analysis."

Data: We have utilized "early stage diabetes risk prediction dataset," which is available on the "UCI machine learning repository" [33], [34].

The processed healthcare data related to diabetes is provided to four most related classifiers, i.e., decision tree, K-nearest neighbors (KNN), logistic regression and support vector machine (SVM). The following outcomes are produced as a result of the assessments.

1) Analysis on Accuracy: Accuracy is a performance criterion for an ML algorithm. It is a measure for assessing classification models. It is the fraction of the "number of correct predictions" to the "total number of predictions." The accuracy for "prediction of getting diabetes" under different ML models has been estimated. The different accuracy values are obtained (for instance, 96.92%, 98.46%, 93.84%, and 97.69% are the accuracy values for decision tree, KNN, logistic regression and SVM techniques, respectively), which are presented in Fig. 6(a). From the estimated results, it is clear that we have obtained a good accuracy value (i.e., ≈98%) for the KNN algorithm.

2) Analysis on F1-Score: It is another important performance parameter, which should be estimated for the considered ML models. The F-score or F1-score is another accuracy parameter of an ML model for a particular dataset. The binary classification systems are first evaluated. They are classified into "positive" or "negative." In other words, we can say that the F1-score is a way of combining of precision and recall parameters. It is formulated as the "harmonic mean of precision and recall values of the model." The F1-score values for different considered models are estimated and shown in Fig. 6(b). We have obtained various F1-score values (for instance, 0.97, 0.99, 0.95, and 0.98 for decision tree, KNN, logistic regression and SVM techniques, respectively). From the obtained results, it is clear that, we have obtained a good F1-score value (i.e., ≈0.99) for KNN algorithm too.

Based on the results of the evaluation, the KNN technique appears to be more effective in terms of accuracy and F1-score, because it has a high accuracy and an excellent F1-score value. We have also provided an impact diagram, which predicts the various factors that may cause the high possibility of diabetes. For example, polydipsia and polyuria have positive impact on diabetes rate, whereas partial paresis increases the diabetes rate. Furthermore, a sudden weight loss could be used as a major factor for early diabetes detection. The obtained outcomes are presented in Fig. 7.

VIII. Conclusion

We designed a new AI-enabled secure communication scheme in a fog computing-based healthcare system (AISCM-FH). The legitimate devices can securely access the associated data under the proposed scheme. The proposed AISCM-FH's security was assessed using the standard random oracle model (ROR model)-based formal security proof as well as non-mathematical (informal) security methods. The security analyzes proved that AISCM-FH is secure against various potential attacks. The pragmatic study of the AISCM-FH is conducted to measure its impact on the important outcomes. The performance of AISCM-FH is compared with other relevant existing schemes to show that AISCM-FH outperforms other protocols because it requires less communication cost, computation cost, and provides superior security as well as more functionality features as compared to those for the considered schemes. As a result, AISCM-FH appears to be a good fit for the delay-efficient services, such as smart healthcare systems.

Acknowledgment

The authors would like to thank the anonymous reviewers and the associate editor for providing their valuable suggestions and comments which helped them to improve the paper significantly.

References

[1] S. Challa, M. Wazid, A. K. Das, and M. K. Khan, "Authentication protocols for implantable medical devices: Taxonomy, analysis and future directions," *IEEE Consum. Electron. Mag.*, vol. 7, no. 1, pp. 57–65, Jan. 2018.

[2] 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.

[3] 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, Dec. 2020.

[4] RedAlkemi. (2018). *Pros & Cons of Internet of Things*. Accessed: Oct. 2019. [Online]. Available: https://www.redalkemi.com/blog/post/pros-cons-of-internet-of-things

[5] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors J.*, vol. 16, no. 1, pp. 254–264, Jan. 2016.

[6] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.

[7] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Gener. Comput. Syst.*, vol. 91, pp. 475–492, Feb. 2019.

[8] Q. Huang, Y. Yang, and L. Wang, "Secure data access control with ciphertext update and computation outsourcing in fog computing for Internet of Things," *IEEE Access*, vol. 5, pp. 12941–12950, 2017.

[9] D. Li, J. Liu, Q. Wu, and Z. Guan, "Efficient CCA2 secure flexible and publicly-verifiable fine-grained access control in fog computing," *IEEE Access*, vol. 7, pp. 11688–11697, 2019.

[10] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for IoT," *IEEE Access*, vol. 7, pp. 38431–38441, 2019.

[11] F. Li, Y. Han, and C. Jin, "Practical access control for sensor networks in the context of the Internet of Things," *Comput. Commun.*, vols. 89–90, pp. 154–164, Sep. 2016.

[12] A. Braeken, P. Porambage, M. Stojmenovic, and L. Lambrinos, "eDAAAS: Efficient distributed anonymous authentication and access in smart Homes," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 12, pp. 1–11, Dec. 2016.

[13] M. Luo, Y. Luo, Y. Wan, and Z. Wang, "Secure and efficient access control scheme for wireless sensor networks in the cross-domain context of the IoT," *Secur. Commun. Netw.*, vol. 2018, pp. 1–10, Aug. 2018, doi: 10.1155/2018/6140978.

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

[15] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptogr. (PKC)*, in Lecture Notes in Computer Science, vol. 3386. Cham, Switzerland: Springer, 2005, pp. 65–84.

[16] J. Wang, D. He, A. Castiglione, B. B. Gupta, M. Karuppiah, and L. Wu, "PCNNCEC: Efficient and privacy-preserving convolutional neural network inference based on cloud-edge-client collaboration," *IEEE Trans. Netw. Sci. Eng.*, early access, May 26, 2022, doi: 10.1109/TNSE.2022.3177755.

[17] A. K. Das, M. Wazid, A. R. Yannam, J. J. P. C. Rodrigues, and Y. Park, "Provably secure ECC-based device access control and key agreement protocol for IoT environment," *IEEE Access*, vol. 7, pp. 55382–55397, 2019.

[18] M. Wazid, A. K. Das, and A. V. Vasilakos, "Authenticated key management protocol for cloud-assisted body area sensor networks," *J. Netw. Comput. Appl.*, vol. 123, pp. 112–126, Dec. 2018.

[19] M. Wazid, A. K. Das, S. Shetty, and M. Jo, "A tutorial and future research for building a blockchain-based secure communication scheme for Internet of Intelligent Things," *IEEE Access*, vol. 8, pp. 88700–88716, 2020.

[20] N. Garg, R. Petwal, M. Wazid, D. P. Singh, A. K. Das, and J. J. P. C. Rodrigues, "On the design of an AI-driven secure communication scheme for Internet of Medical Things environment," *Digit. Commun. Netw.*, Apr. 2022, doi: 10.1016/j.dcan.2022.04.009.

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

[22] 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.

[23] P. Gope, J. Lee, and T. Q. S. Quek, "Resilience of dos attacks in designing anonymous user authentication protocol for wireless sensor networks," *IEEE Sensors J.*, vol. 17, no. 2, pp. 498–503, Jan. 2017.

[24] 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.

[25] X. L. 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.

[26] M. Wazid, A. K. Das, K.-K.-R. Choo, and Y. Park, "SCS-WoT: Secure communication scheme for web of things deployment," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10411–10423, Jul. 2022.

[27] 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. 2017.

[28] A. K. Das, "A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks," *Int. J. Inf. Secur.*, vol. 11, no. 3, pp. 189–211, Jun. 2012.

[29] S. Sayeed and H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Appl. Sci.*, vol. 9, no. 9, p. 1788, Apr. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/9/1788

[30] Secure Hash Standard. (Apr. 1995). *FIPS PUB 180-1, National Institute of Standards and Technology (NIST)*. U.S. Department of Commerce. Accessed: Mar. 2022. [Online]. Available: http://www.umich.edu/~x509/ssleay/fip180/fip180-1.htm

[31] E. Barker. (2014). *Recommendation for Key Management*. Accessed: May 2018. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final

[32] L. Wu, J. Wang, K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 319–330, Feb. 2019.

[33] (2020). *Early Stage Diabetes Risk Prediction Dataset*. Accessed: Jun. 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset

[34] M. M. F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, "Likelihood prediction of diabetes at early stage using data mining techniques," in *Computer Vision and Machine Intelligence in Medical Image Analysis*, M. Gupta, D. Konar, S. Bhattacharyya, and S. Biswas, Eds. Singapore: Springer, 2020, pp. 113–125.

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 IIIT, Hyderabad, India. He is currently working as an Associate Professor with the Department of Computer Science and Engineering, Graphic Era Deemed to be University. His current research interests include security, authentication, the Internet of Things (IoT), cloud computing, and blockchain. He has published more than 100 papers in international journals and conferences in the above areas.

Ashok Kumar Das (Senior Member, IEEE) received the M.Tech. degree in computer science and data processing, the M.Sc. degree in mathematics from IIT Kharagpur, India, and the Ph.D. degree in computer science and engineering. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, IIIT, Hyderabad, India. He was a Visiting Faculty with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA, USA. His current research interests include cryptography, system and network security, blockchain, and AI/ML security. He has authored over 325 papers in international journals and conferences in the above areas, including over 275 reputed journal articles. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He was/is 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 Chair of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, in June 2019, International Conference on Applied Soft Computing and Communication Networks (ACN'20), Chennai, India, in October 2020, and second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, in October 2020. His Google Scholar H-index is 72 and i10-index is 204 with over 14,200 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, Old Dominion University. He holds a joint appointment with the Center for Cybersecurity Education and Research and the Department of Modeling, Simulation and Visualization Engineering. He has authored or coauthored over 125 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.

Joel J. P. C. Rodrigues (Fellow, IEEE) is currently a Professor at the College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China, and a Senior Researcher at the Instituto de Telecomunicações, Portugal. He is also the Leader of the Next Generation Networks and Applications Research Group (CNPq), the Director for Conference Development—IEEE ComSoc Board of Governors, IEEE Distinguished Lecturer, the Technical Activities Committee Chair of the IEEE ComSoc Latin America Region Board, the President of the Scientific Council at ParkUrbis—Covilhã Science and Technology Park, the Past Chair of the IEEE ComSoc Technical Committee on eHealth, the Past Chair of the IEEE ComSoc Technical Committee on Communications Software, a Steering Committee Member of the IEEE Life Sciences Technical Community and the Publications Co-Chair, and a Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the Editor-in-Chief of the *International Journal on E-Health and Medical Communications* and an editorial board member of several high-reputed journals. He has been the General Chair and the TPC Chair of many international conferences, including IEEE ICC, IEEE GLOBECOM, IEEE HEALTHCOM, and IEEE LatinCom. He has authored or coauthored over 1000 papers in refereed international journals and conferences, three books, two patents, and one ITU-T recommendation. He had been awarded several Outstanding Leadership and Outstanding Service Awards by IEEE Communications Society and several best papers awards. He is a member of the Internet Society and a Senior Member of ACM.

Mohsen Guizani (Fellow, IEEE) received the B.S. (Hons.) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor with the Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Masdar City, Abu Dhabi, United Arab Emirates (UAE). Previously, he has worked in different academic and administrative positions at Qatar University, University of Idaho, Western Michigan University, the University of West Florida, the University of Missouri-Kansas City, the University of Colorado-Boulder, and Syracuse University. He is the author of nine books and more than 1100 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is also a Senior Member of ACM. Throughout his career, he received three teaching awards and four research awards. He was a recipient of the 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, the 2018 Ad-Hoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks, and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award for Outstanding Contributions to the Technological Advancement of Security. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He has served as a member, the chair, and the general chair of a number of international conferences. He is also the Editor-in-Chief of the *IEEE Network*. He serves on the editorial boards for several international technical journals. He serves the Founder and the Editor-in-Chief for *Wireless Communications and Mobile Computing* journal (Wiley). He has served as the IEEE Computer Society Distinguished Speaker. He is also the IEEE ComSoc Distinguished Lecturer. He is an Associate Editor of various journals, including *Ad Hoc Networks*, *IEEE Internet of Things Magazine*, IEEE NETWORKING LETTERS, *IET Networks*, and *IET Quantum Communications*.