Cost-effective Authenticated Solution (CAS) for 6G-enabled Artificial Intelligence of Medical Things (AIoMT)

Khalid Mahmood *Senior Member, IEEE*, Mohammad S. Obaidat *Life Fellow of IEEE*, Salman Shamshad, Muhammad Asad Saleem, Gulshan Kumar, Mohammad Hossein Anisi, *Senior Member, IEEE* and Mauro Conti *Fellow, IEEE*

***Abstract*—The Internet of Things (IoT) is a network of interconnected objects, which congregate and exchange gigantic amounts of data. Usually, pre-deployed embedded sensors sense this massive data. Soon several applications of IoT are anticipated to exploit emerging 6G technology. Healthcare is one of them, where the 6G-inspired paradigm may facilitate the users to exchange information through hundreds of sensors under the assumption of Artificial Intelligence of Things (AIoT). Integration of medical sensors with AIoT is known as Artificial Intelligence of Medical Things (AIoMT). The secure and seamless interactions among 6G-enabled AIoMT users should be the primary challenge. Furthermore, resource-constrained wearable sensing devices, with their inability to execute complex security solutions, provide an ideal attraction for malicious entities to launch diverse attacks. These challenges have motivated us to design a cost-effective authenticated solution (CAS) for 6G-enabled AIoMT healthcare applications.**

Our CAS protocol not only prevents cyber threats like impersonation session key secrecy, it can also prevent physical threats like hardware tampering. We observe formal and informal security validations to endorse its robustness and effectiveness. Performance comparison reveals that CAS protocol offers maximum security enrichment. Moreover, CAS is cost-effective as it has achieved 8% and 52% reduction in computation and communication cost, respectively, compared to contemporary competing related protocols.

***Index Terms*—Authentication, Authentication Protocol, Mutual Authentication, Key Agreement Protocol**

Khalid Mahmood is with the Graduate School of Intelligent Data Science, National Yunlin University of Science and Technology, Yunlin, ROC, Taiwan (e-mail:khalidm.research@gmail.com)

Mohammad S. Obaidat is with the King Abdullah II School of Information Technology, The University of Jordan, Amman 11942, Jordan, also with the School of Communication and Computing Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with the Amity School of Engineering and Technology, Amity University, Noida, Uttar Pradesh 201301, India (e-mail: msobaidat@gmail.com)

Gulshan Kumar, Mauro Conti are with Department of Mathematics, University of Padua, 35131 Padua, Italy. (emails: gulshan3971@gmail.com; mauro.conti@unipd.it)

Muhammad Asad Saleem is with School of Computer Science and Engineering, University of Electronic Science and Technology of China, 12599 Chengdu, Sichuan, China, (e-mail: masadsaleem123@gmail.com

Salman Shamshad is with the Department of Software Engineering, University of Lahore, Lahore 54590, Pakistan (e-mail:salmanshamshad01@gmail.com)

Mohammad Hossein Anisi is with School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, United Kingdom. (email: m.anisi@essex.ac.uk)

(Corresponding Author: Khalid Mahmood)

I. Introduction

The incredible success of 3G and 4G networks has influenced the launch of 5G networks. The unprecedented growth of 5G has encouraged the proliferation of next-generation networks. This includes 6G and Internet of Things (IoT), aiming to offer end-users seamless networking capabilities at higher data rates. IoT facilitates the interconnection of tiny sensing devices through the modern communication system to develop a top-notch plan adept at sensing, monitoring, analyzing, and exchanging invaluable information. Such systems can help us manage business solutions with nimbler efficiency and efficacy for industrial organizations [1].

The 6G-enabled IoT infrastructure users can be individuals with their gadgets, sensing devices, or systems with nimbler enabling technologies in their diverse applications. Artificial Internet of Medical Things (AIoMT) is one of the critical applications of 6G enabled IoT. AIoMT defines the process of sensing, processing, and intelligently communicating biomedical data via remote access. This application aims at maintaining the medical history/records of patients over the cloud so that physicians can access them at their convenience as per their authorization. If we narrow down the infrastructure, Wireless Medical Sensor Networks (WMSNs), a variant of Wireless Sensor Networks (WSNs), facilitate the healthcare application in the 6G-enabled AIoMT. WMSNs enable hospitals or concerned physicians to monitor the vitals of the patients under observation, like their pulse rate, blood pressure, temperature, etc., [2], [3].

The growing use of industrial procedures in the healthcare ecosystem invites new challenges, vulnerabilities, and risks for patients and physicians. Eventually, the probability of malicious attempts is high, where an adversary can attempt to breach the cyber and physical security of IoT devices connected to the Internet (i.e., open channel). Therefore, it is imperative to deploy robust security mechanisms to defend against such adversarial threats. No doubt, plenty of cyber-security solutions exist in the literature. Still, they are not directly applicable to IoT-based healthcare systems since they require ample network resources or fail to defend manifold attacks. For instance, Jangirala et al. [4] developed a cloud-assisted authentication solution for the remote health monitoring ecosystem. Unfortunately, their protocol [4] was unable to protect impersonation and verifier attacks. He et al. [5] also

constructed a lightweight Cross-Domain Handshake (CDHS) protocol for mobile healthcare systems. Nevertheless, the protocol in [5] is defenseless against the physical capturing of devices and anonymity violation attacks. Li et al. [6] introduced a key negotiation scheme based on a chaotic map for the mobile healthcare system. Madhusudhan-Nayak [7] later disclosed that the scheme of [6] is defenseless against impersonation and guessing threats. After that, [7] came up with robust and improved security features. Similarly, Qiu et al. [8] developed a key establishment protocol using Elliptic Curve Cryptographic (ECC) techniques for e-healthcare systems. Later, Kumari and Renuka [9] identified numerous security loopholes, including guessing, impersonation, and anonymity violation attacks risks. In 2021, Barman et al. [10] designed an identity-based secure access control mechanism for remote patients in multiserver-assisted healthcare systems. Moreover, they claimed the robustness of their protocol against well-known attacks. Jia et al. [11] also suggested a fog-assisted bilinear-based protocol for the healthcare ecosystem. The authors of [12] investigated their protocol and revealed its susceptibilities such as fog node masquerading, ESL, and anonymity violations attacks. Later, the authors in [13] argued that the protocol in [10] has no resistance against a stolen verifier, masquerading and ESL attacks. Madhusudhan and Nayak [7] presented another security solution for healthcare [7]. Nonetheless, Sureshkumar et al. [14] demonstrated the shortcoming of [7] and argued that the work of [7] is prone to traceability, replay, and masquerading threats. The authors of [15] and [16] recently also designed access control security mechanisms to achieve desired security of the e-healthcare system. Unfortunately, their protocols are defenseless against verifier and device tampering/cloning attacks.

A. Motivation and Contributions

In 6G enabled AIoMT healthcare applications, thousands or even millions of users are assumed to exchange critical real-time information directly from distinct sources. The exchange of sensitive data in such infrastructure demands a suitable access control mechanism for promising privacy and security requirements. Therefore, we can assume that the primary challenge is a secure and seamless exchange of information among users. Moreover, resource-constrained wearable sensing devices with their ineptitude to implement and execute classic complex security solutions provide an ideal point of attraction for malicious users to launch severe security threats. These challenges have motivated us to design and develop a cost-effective authenticated solution for 6G-enabled AIoMT healthcare applications. In anticipation of the challenges discussed, our significant contributions are as summarized below:

- We design and develop a cost-effective authenticated solution for 6G enabled IoT healthcare applications. CAS protocol exploits trivial primitives of cryptography such as bit-wise XoR, string concatenation, and hash function to minimize the development complexity.
- Our solution not only prevents cyber threats, it is also able to prevent physical threats like hardware tampering by employing physically unclonable function
- Since sensing device can not transmit its sensed data directly toward the user due to its limited communication range, therefore, a cloud of things server can act as an intermediary entity between user and sensing device to negotiate SK for secure data transmission.
- Performance comparison has revealed that CAS protocol offers maximum security enrichment. Moreover, it is cost-effective as it has achieved 8% and 52% reduction in computation and communication cost, respectively, compared to contemporary competing related protocols.

II. Network and Threat Models

In order to solicit the implementation and operation of CAS protocol, we briefly explain network and threat models as follows.

A. Network Model

Fig. 1 outlines the network model of our designed protocol for a 6G-enabled AIoMT-based healthcare system, which mainly encompasses three entities, including Cloud of Things Server (CTS), communication interface, and various health monitoring sensors. The CTS plays a significant role in storing health information collected from sensors installed in a patient's body. The healthcare physicians (i.e., a medical advisor or a doctor) can access the real-time health information from the health monitoring sensors of patients using a communication interface to write a prescription. Therefore, due to the sensitivity of collected data, this study focuses on securing the communication between health monitoring sensors and communication interfaces.

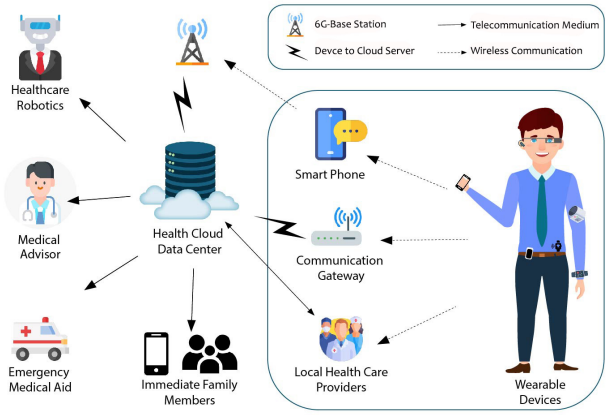

Fig. 1: 6G-enabled AIoMT-based Healthcare System

B. Threat Model

This article has adopted a globally-recognized DY threat model [17] to solicit the scrutiny of the developed protocol. According to the DY model, $\mathfrak{A}$ has full access and control over the communication channel. In addition to the DY model, we also followed the CK [18] threat model, which is considered more robust than the DY model. In accordance with the CK model, $\mathfrak{A}$ can launch compromise session key security through the leakage of permanent and temporary secrets.

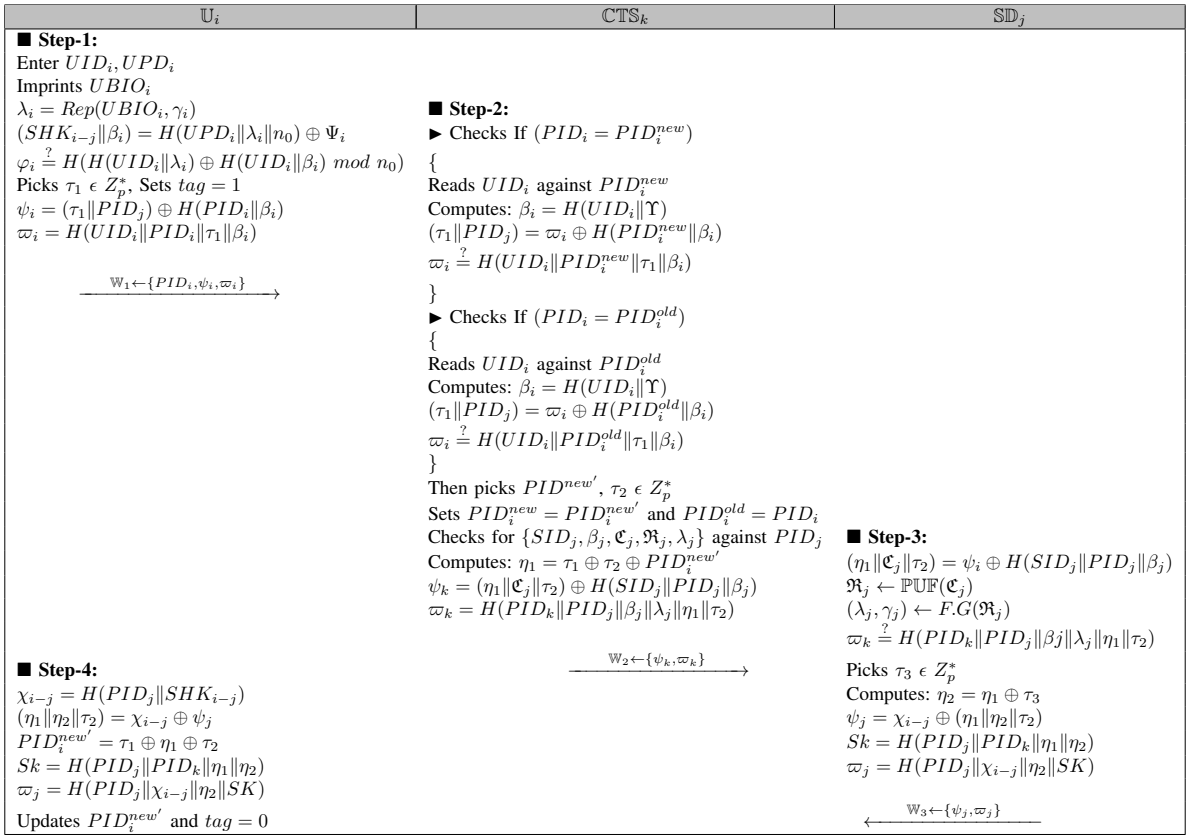

Fig. 2: Authentication Key Establishment Phase of CAS

TABLE I: Notations Guide

Notation	*Description*
$\mathbb{CTS}_k$	Cloud of Things Server
Υ, PID_k	Pseudonym and Secret key of $\mathbb{CTS}_k$
$\mathbb{U}_i$	i^{th} User
UID_i, PID_i	$\mathbb{U}_i$'s Real and Pseudo Identities
$UPD_i, UBIO_i$	$\mathbb{U}_i$'s Password and Biometric
$\mathbb{SD}_j$	jth Wearable Sensing Device
SID_j	$\mathbb{SD}_j$'s Unique Identity
SHK_{i-j}	Shared key between $\mathbb{U}_i$ and $\mathbb{SD}_j$
PID_j	$\mathbb{SD}_j$'s Pseudonym
$\mathbb{PUF}_j$	Physically Unclonable Function
$< \mathfrak{C}_j, \mathfrak{R}_j >$	Challenge-Response Pairs
$\mathfrak{A}$	Adversary

III. CAS Protocol

This section devises CAS protocol developed for 6G-enabled AIoMT, which primarily consists of three entities including: $\mathbb{U}_i$, $\mathbb{CTS}_k$ and $\mathbb{SD}_j$. To get the patient's real-time information through his $\mathbb{SD}_j$, $\mathbb{U}_i$ needs to login at the terminal device located in the hospital. $\mathbb{SD}_j$ can not transmit its sensed data directly toward $\mathbb{U}_i$ due to its limited communication range. Therefore, $\mathbb{CTS}_k$ acts as an intermediary entity between $\mathbb{U}_i$ and $\mathbb{SD}_j$ to negotiate SK between them for secure data transmission. Table I outlines notations and their meaning we used throughout the design of CAS protocol.

A. System Setup

This is the initial phase of our developed protocol, which is performed in an offline manner. The cloud of things server $\mathbb{CTS}_k$ firstly generates a hash function $H : \{1,0\}^* \rightarrow \{1,0\}^{256}$ and chooses its secret key Υ. $\mathbb{CTS}_k$ then chooses random number generation function Z_p^*, concatenation ($\|$) and exclusive-OR ($\oplus$) operations. Finally, $\mathbb{CTS}_k$ secretly holds Υ and publicly shares $\{Z_p^*, \|, \oplus, H(.)\}$ system parameters.

B. User Registration Phase

In order to access the real-time healthcare information from $\mathbb{SD}_j$, the user $\mathbb{U}_i$ (i.e., health professional/ physician) needs to submit registration request to $\mathbb{CTS}_k$. The detailed description about $\mathbb{U}_i$ registration is given in the following steps:

1: $\mathbb{U}_i$ initially picks UID_i and UPD_i. Next, $\mathbb{U}_i$ randomly generates $n_0\ \epsilon\ Z_p^*$ and imprints his $UBIO_i$ on the interface of his terminal device. $\mathbb{U}_i$ then submits UID_i toward $\mathbb{CTS}_k$ as a registration request message.

2: Upon receiving $UID_i, \mathbb{CTS}_k$ picks PID_i and sets $PID_i^{new} = PID_i, PID_i^{old} = NULL$. At the same time $\mathbb{CTS}_k$ computes: $\beta_i = H(UID_i \| \Upsilon)$. Therefore, $\mathbb{CTS}_k$ writes $\{PID_i^{new} = PID_i, PID_i^{old} = NULL, UID_i\}$ in $\mathbb{U}_i$'s identity table against PID_i and encodes them with Υ. At the end, $\mathbb{CTS}_k$ chooses SHK_{i-j} for all $\mathbb{SD}_j$ and transmits $\{\beta_i, SHK_{i-j}, PID_i\}$ to $\mathbb{U}_i$.

3: $\mathbb{U}_i$ receives $\{\beta_i, SHK_{i-j}, PID_i\}$ and computes: $\lambda_i = Rep(UBIO_i, \gamma_i)$, $\Psi_i = H(UPD_i \| \lambda_i \| n_0) \oplus (SHK_{i-j} \| beta_i)$ and $\varphi_i = H(H(UID_i \| \lambda_i) \oplus H(UID_i \| \beta_i)\ mod\ n_0)$. Next, $\mathbb{U}_i$ selects a Boolean variable tag and assigns it with 0. Finally, $\mathbb{U}_i$ securely holds $\{\Psi_i, \varphi_i, n_0\}$ for later use.

C. Pre-Deployment

The detailed $\mathbb{U}_i$ pre-deployment phase comprises the following steps:

1: $\mathbb{SD}_j$ chooses his unique SID_j at first, and sends it to $\mathbb{CTS}_k$ over unreliable channel.

2: Upon getting SID_j, $\mathbb{CTS}_k$ checks whether the received SID_j exists in $\mathbb{SD}_j$'s identity table or not. If it is not found there, $\mathbb{CTS}_k$ generates a pseudonym PID_j and challenge message $\mathfrak{C}_j$. Thereafter, $\mathbb{CTS}_k$ computes: $\beta_j = H(SID \| \Upsilon)$ and $\chi_{i-j} = H(PID_j \| SHK_{i-j})$. $\mathbb{CTS}_k$ finally transmits $\{\mathfrak{C}_j, \chi_{i-j}, PID_j\}$ to $\mathbb{SD}_j$.

3: Whenever $\mathbb{SD}_j$ gets $\{\mathfrak{C}_j, \chi_{i-j}, PID_j\}$ from $\mathbb{CTS}_k$, $\mathbb{SD}_j$ calculates: $\mathfrak{R}_j \leftarrow \mathbb{PUF}_j(\mathfrak{C}_j)$, $(\lambda_j, \gamma_j) \leftarrow F.G(\mathfrak{R}_j)$ and sends $\{\mathfrak{R}_j, \lambda_j\}$ back to $\mathbb{CTS}_k$. Meanwhile, $\mathbb{SD}_j$ also writes $\{SID_j, \beta_j, \chi_{i-j}\}$ in its memory.

4: On getting $\{\mathfrak{R}_j, \lambda_j\}$, $\mathbb{CTS}_k$ records $\{SID_j, \mathfrak{C}_j, \mathfrak{R}_j, \lambda_j\}$ in $\mathbb{SD}_j$'s identity table against PID_j and encodes them with Υ.

D. Authentication and Key Negotiation

To securely access real-time health information directly from $\mathbb{SD}_j$, $\mathbb{U}_i$ must negotiate session key with $\mathbb{SD}_j$ via $\mathbb{CTS}_k$. The detailed description of key negotiation between $\mathbb{SD}_j$ and $\mathbb{U}_i$ are given in the following steps:

1: Initially, $\mathbb{U}_i$ inputs his UID_i, UPD_i and $UBIO_i$ into terminal device. The device then computes: $\lambda_i = Rep(UBIO_i, \gamma_i)$, $(SHK_{i-j} \| \beta_i) = H(UPD_i \| \lambda_i \| n_0) \oplus \Psi_i$ and compares $\varphi_i \stackrel{?}{=} H(H(UID_i \| \lambda_i) \oplus H(UID_i \| \beta_i) \ mod \ n_0)$. If they are equal, $\mathbb{U}_i$ picks $\tau_1 \epsilon Z_p^*$ and sets $tag = 1$. Thereafter, $\mathbb{U}_i$ computes: $\psi_i = (\tau_1 \| PID_j) \oplus H(PID_i \| \beta_i)$, $\varpi_i = H(UID_i \| PID_i \| \tau_1 \| \beta_i)$ and transmits $\mathbb{W}_1 \leftarrow \{PID_i, \psi_i, \varpi_i\}$ toward $\mathbb{CTS}_k$.

2: On receiving $\mathbb{W}_1$ from $\mathbb{U}_i$, $\mathbb{CTS}_k$ reads its identity table to check whether $PID_i = PID_i^{new}$ or $PID_i = PID_i^{old}$.

- If the received PID_i matches with PID_i^{new}, then $\mathbb{CTS}_k$ perceives that pseudonym of $\mathbb{U}_i$ was updated in the last session. Thereafter, $\mathbb{CTS}_k$ reads UID_i against PID_i^{new}, computes: $\beta_i = H(UID_i \| \Upsilon)$, $(\tau_1 \| PID_j) = \varpi_i \oplus H(PID_i^{new} \| \beta_i)$ and validates the authenticity of $\mathbb{U}_i$ through $\varpi_i \stackrel{?}{=} H(UID_i \| PID_i^{new} \| \tau_1 \| \beta_i)$. If there is a match, then $\mathbb{CTS}_k$ believes that $\mathbb{U}_i$ is a legal user. Else, ends the session.
- If the received PID_i matches with PID_i^{old}, then $\mathbb{CTS}_k$ perceives that pseudonym of $\mathbb{U}_i$ was not updated in the last session. Therefore, $\mathbb{CTS}_k$ reads UID_i against PID_i^{old}, computes: $\beta_i = H(UID_i \| \Upsilon)$, $(\tau_1 \| PID_j) = \varpi_i \oplus H(PID_i^{old} \| \beta_i)$ and validates the authenticity of $\mathbb{U}_i$ through $\varpi_i \stackrel{?}{=} H(UID_i \| PID_i^{old} \| \tau_1 \| \beta_i)$. If there is a match, then $\mathbb{CTS}_k$ believes that $\mathbb{U}_i$ is a legitimate. Elseways, ends the session.
- If the received PID_i does not match with PID_i^{old} or PID_i^{new}, then $\mathbb{CTS}_k$ immediately terminates the session.

$\mathbb{CTS}_k$ then picks $PID^{new'}$, $\tau_2 \epsilon Z_p^*$ updates $PID_i^{new} = PID_i^{new'}$ and $PID_i^{old} = PID_i$. Next, $\mathbb{CTS}_k$ reads $\{SID_j, \beta_j, \mathfrak{C}_j, \mathfrak{R}_j, \lambda_j\}$ against PID_j through Υ and computes: $\eta_1 = \tau_1 \oplus \tau_2 \oplus PID_i^{new'}$, $\psi_k = (\eta_1 \| \mathfrak{C}_j \| \tau_2) \oplus H(SID_j \| PID_j \| \beta_j)$ and $\varpi_k = H(PID_k \| PID_j \| \beta_j \| \lambda_j \| \eta_1 \| \tau_2)$. At the end, $\mathbb{CTS}_k$ transmits $\mathbb{W}_2 \leftarrow \{\psi_k, \varpi_k\}$ to $\mathbb{SD}_j$.

3: Upon getting $\mathbb{W}_2$, $\mathbb{SD}_j$ computes: $(\eta_1 \| \mathfrak{C}_j \| \tau_2) = \psi_i \oplus H(SID_j \| PID_j \| \beta_j)$ and determines $\mathfrak{R}_j \leftarrow \mathbb{PUF}(\mathfrak{C}_j)$. To avoid $\mathfrak{R}_j$ from noise, $\mathbb{SD}_j$ further gets $(\lambda_j, \gamma_j) \leftarrow F.G(\mathfrak{R}_j)$. Next, $\mathbb{SD}_j$ verifies $\varpi_k \stackrel{?}{=} H(PID_k \| PID_j \| \beta j \| \lambda_j \| \eta_1 \| \tau_2)$. If it is valid, $\mathbb{SD}_j$ picks $\tau_3 \epsilon Z_p^*$ and computes: $\eta_2 = \eta_1 \oplus \tau_3$, $\psi_j = \chi_{i-j} \oplus (\eta_1 \| \eta_2 \| \tau_2)$, $Sk = H(PID_j \| PID_k \| \eta_1 \| \eta_2)$ and $\varpi_j = H(PID_j \| \chi_{i-j} \| \eta_2 \| SK)$. Finally, $\mathbb{SD}_j$ sends $\mathbb{W}_3 \leftarrow \{\psi_j, \varpi_j\}$ toward $\mathbb{U}_i$.

4: After receiving $\mathbb{W}_3$, $\mathbb{U}_i$ computes: $\chi_{i-j} = H(PID_j \| SHK_{i-j})$, $(\eta_1 \| \eta_2 \| \tau_3) = \chi_{i-j} \oplus \psi_j$, $PID_i^{new'} = \tau_1 \oplus \eta_1 \oplus \tau_2$, $Sk = H(PID_j \| PID_k \| \eta_1 \| \eta_2)$ and matches $\varpi_j \stackrel{?}{=} H(PID_j \| \chi_{i-j} \| \eta_2 \| SK)$. If the condition does not hold true, $\mathbb{U}_i$ ends the session. Elseway, accepts the session key SK. At the end, $\mathbb{U}_i$ updates $PID_i^{new'}$ and sets $tag = 0$.

E. Dynamic Device Addition Phase

Our designed protocol allows $\mathbb{CTS}_k$ to dynamically add a new wearable sensor $\mathbb{SD}_j^{new}$ in the existing network. The dynamic $\mathbb{SD}_j^{new}$ addition phase is described in the following steps:

1: $\mathbb{SD}_j^{new}$ chooses his unique SID_j^{new} at first, and sends it towards $\mathbb{CTS}_k$ over unreliable channel.

2: Upon getting SID_j^{new}, $\mathbb{CTS}_k$ checks whether the received SID_j^{new} exists in $\mathbb{SD}_j$'s identity table or not. If it is not found there, $\mathbb{CTS}_k$ generates a pseudonym PID_j^{new} and a challenge message $\mathfrak{C}_j^{new}$. Thereafter, $\mathbb{CTS}_k$ computes: $\beta_j^{new} = H(SID_i^{new} \| \Upsilon)$ and $\chi_{i-j}^{new} = H(PID_j^{new} \| SHK_{i-j})$. $\mathbb{CTS}_k$, finally transmits $\{\mathfrak{C}_j^{new}, \chi_{i-j}^{new}, PID_j^{new}\}$ to $\mathbb{SD}_j^{new}$.

3: Whenever $\mathbb{SD}_j^{new}$ gets $\{\mathfrak{C}_j^{new}, \chi_{i-j}^{new}, PID_j^{new}\}$ from $\mathbb{CTS}_k$, $\mathbb{SD}_j^{new}$ calculates: $\mathfrak{R}_j^{new} \leftarrow \mathbb{PUF}_j(\mathfrak{C}_j^{new})$, $(\lambda_j^{new}, \gamma_j^{new}) \leftarrow F.G(\mathfrak{R}_j^{new})$ and sends $\{\mathfrak{R}_j^{new}, \lambda_j^{new}\}$ back to $\mathbb{CTS}_k$. Meanwhile, $\mathbb{SD}_j^{new}$ also writes $\{SID_j^{new}, \beta_j^{new}, \chi_{i-j}^{new}\}$ in its memory.

4: On getting $\{\mathfrak{R}_j^{new}, \lambda_j^{new}\}$, $\mathbb{CTS}_k$ records $\{SID_j^{new}, \mathfrak{C}_j^{new}, \mathfrak{R}_j^{new}, \lambda_j^{new}\}$ in $\mathbb{SD}_j$'s identity table against PID_j^{new} and encodes them with Υ.

IV. Security Evaluation

In this section, we investigate the security strength of our developed protocol through informal and formal security evaluation.

A. Formal Security Evaluation

The developed protocol is evaluated formally under the Random or Real (RoR) security model to prove its semantic security. According to RoR, the n^{th} instance of an entity

$\mathfrak{E}$ is denoted as $\mathfrak{R}_n$. The cloud of things server $\mathbb{CTS}_k$, the user $\mathbb{U}_i$ and wearable sensing device $\mathbb{SD}_j$ are symbolized as the entities $\mathfrak{E}_{\mathbb{CTS}_k}$, $\mathfrak{E}_{\mathbb{U}_i}$ and $\mathfrak{E}_{\mathbb{SD}_j}$, whereas their instances are denotes as: n_1^{th}, n_2^{th} and n_3^{th}, respectively. The hash function $H(.)$ is simulated as $HASH$ and assumed to be publically know to all other entities. Moreover, the set of queries for $\mathfrak{A}$ in modeling attack is summarized as follows

- $EXECUTE(\mathfrak{E}_{\mathbb{CTS}_k}, \mathfrak{E}_{\mathbb{U}_i}, \mathfrak{E}_{\mathbb{SD}_j})$: This helps $\mathfrak{A}$ to listen the public communicated messages among $\mathfrak{E}_{\mathbb{CTS}_k}$, $\mathfrak{E}_{\mathbb{U}_i}$ and $\mathfrak{E}_{\mathbb{SD}_j}$.
- $SEND(\mathfrak{E}^n, \mathcal{W})$: By this query, $\mathfrak{A}$ can submit message $\mathcal{W}$ to $\mathfrak{E}^n$ in order to get response from $\mathfrak{E}^n$.
- $CORRUPT(\mathfrak{E}^{n_1}_{\mathbb{CTS}_k})$: This query allows $\mathfrak{A}$ to extract the datum stored inside the device of $\mathbb{U}_i$.
- $REVEAL_{SK}(\mathfrak{E}^{n_1})$: $\mathfrak{A}$ executes this to reveal SK shared between $\mathbb{U}_i$ and $\mathbb{SD}_j$.
- $TEST(\mathfrak{E}^n)$: Through this query, $\mathfrak{A}$ flips an unbiased coin z.

We also employ Zipf's law [19] to show the security of the developed protocol. Theorem 1 presents the detailed proof as follows: **Theorem 1:** *Suppose $ADVT^{\mathcal{P}}_{\mathfrak{A}}(tme)$ denotes the advantage of $\mathfrak{A}$ in breaking the semantic security of developed protocol AKA in polynomial time tme. Let, $|HASH|$, q_{read}, and q_{hash} signify the length of hash function, read query and hash queries, respectively. In addition, s and C symbolizes Zipf's parameters, and ln represents the length of $\mathbb{U}_i$'s biometric key in the following equation:*

$$ADVT^{\mathcal{P}}_{\mathfrak{A}}(tme) \leq \frac{q^2_{hash}}{|HASH|} + 2max\{C.q^s_{read}, \frac{q_{read}}{s^{ln}}\}$$

Proof: We follow the identical proof as described in [4]. The solicited semantic security using series of four games, denoted as $GAME_p$, where $p = [1, 4]$, while $SUCCE^{GAME_p}_{\mathfrak{A}}$ as the as the advantage of $\mathfrak{A}$ in guessing the output of a flipped z. Moreover, we symbolizes $\mathfrak{A}$'s advantage in winning $GAME_p$ as $ADVT^{GAME_p}_{\mathfrak{A}} = Pro[SUCCE^{GAME_p}_{\mathfrak{A}}]$. Each $GAME_p$ is described in detailed as follows. • $GAME_1$: This game is analogous to an active attack correspondent to $GAME_1$. At the beginning, the output of a flipped z is chosen arbitrarily:

$$ADVT^{GAME_p}_{\mathfrak{A}} = |2.ADVT^{GAME_1}_{\mathfrak{A}} - 1| \tag{1}$$

• $GAME_2$: This game corresponds to sniffing attack. Here, $\mathfrak{A}$ tries to listen communicated messages: $\mathbb{W}_1 \leftarrow \{PID_i, \psi_i, \varpi_i\}$, $\mathbb{W}_2 \leftarrow \{\psi_k, \varpi_k\}$ and $\mathbb{W}_3 \leftarrow \{PID_i, \psi_j, \varpi_j\}$ among $\mathbb{U}_i$, $\mathbb{CTS}_k$ and $\mathbb{SD}_j$ through $EXECUTE$ query. Thereafter, $\mathfrak{A}$ models $REAVEL$ and $TEST$ queries to verify whether SK is a random nonce or valid session key. Referring to $SK = H(PID_j \| PID_k \| \eta_1 \| \eta_2)$, it can be noticed that the construction SK encompasses with random nonce and long term keys which are unavailable to $\mathfrak{A}$. Consequently, $\mathfrak{A}$ can not determine real SK from random nonce. As, $GAME_1$ and $GAME_2$ are identical, then it follows:

$$ADVT^{GAME_1}_{\mathfrak{A}} = ADVT^{GAME_2}_{\mathfrak{A}} \tag{2}$$

• $GAME_3$: This games can assist A to execute "'active attack"' by modeling $HASH$ oracle. The parameters $\{PID_i, \psi_i, \varpi_i\}$, $\{\psi_k, \varpi_k\}$ and $\{PID_i, \psi_j, \varpi_j\}$ in network messages $\mathbb{W}_1$, $\mathbb{W}_2$ and $\mathbb{W}_3$, receptively are safe under the irreversible hash function $H(.)$. Therefore, it is infeasible to determine the pre-image. Additionally, the composition of these messages includes random nonce, which helps to make these messages indistinguishable. It is clear that $GAME_2$ and $GAME_3$ are similar except for the fact that $GAME_3$ includes $HASH$ oracle. Therefore, from the birthday paradox of hash function, we have,

$$ADVT^{GAME_2}_{\mathfrak{A}} = ADVT^{GAME_3}_{\mathfrak{A}} \leq \frac{q^2_{hash}}{2|HASH|} \tag{3}$$

• $GAME_4$: In this game, $\mathfrak{A}$ attempts to physically tamper to mobile device of $\mathbb{U}_i$ by modeling $CORRUPT$ query. Assume that $\mathfrak{A}$ possessed the device and successfully reveled its datum $\{\psi_k, \varpi_k, n_0\}$. However, it is computationally infeasible to construct authentication message $\mathbb{W}_1$ without having UID_i, UPD_i and λ_i. Additionally, the feasibility of guessing λ_i of ln bits if almost $\frac{1}{s^{ln}}$. It can be noticed that both $GAME_3$ and $GAME_4$ are indistinguishable except the involvement of biometric/password guessing. Consequently, referring to the Zipf's Law, it is as follows:

$$ADVT^{GAME_3}_{\mathfrak{A}} = ADVT^{GAME_4}_{\mathfrak{A}} \leq max\{C.q^s_{read}, \frac{q_{read}}{2^{ln}}\} \tag{4}$$

Now, the advantage of $\mathfrak{A}$ in guessing z's output by modeling games $GAME_p, p \in [1, 4]$ is $ADVT^{GAME_4}_{\mathfrak{A}} = \frac{1}{2}$. From the Equation (1) and (2), we obtain:

$$\frac{1}{2}.ADVT^{AKA}_{\mathfrak{A}}(tme) = |ADVT^{GAME_1}_{\mathfrak{A}} - \frac{1}{2}|$$
$$= |ADVT^{GAME_2}_{\mathfrak{A}} - \frac{1}{2}| = |ADVT^{GAME_2}_{\mathfrak{A}} - ADVT^{GAME_4}_{\mathfrak{A}}| \tag{5}$$

Adopting the triangular inequality and Equations (3), (4) & (5), we get:

$$\tfrac{1}{2}.ADVT^{AKA}_{\mathfrak{A}}(tme) = |ADVT^{GAME_2}_{\mathfrak{A}} - ADVT^{GAME_4}_{\mathfrak{A}}|$$
$$|ADVT^{GAME_2}_{\mathfrak{A}} - ADVT^{GAME_3}_{\mathfrak{A}}| + |ADVT^{GAME_3}_{\mathfrak{A}} - ADVT^{GAME_4}_{\mathfrak{A}}|$$
$$\leq \frac{q^2_{hash}}{2|HASH|} + max\{C.q^s_{read}, \frac{q_{read}}{ln}\} \tag{6}$$

At the end, multiplying either side of Equation (6) by 2, we obtained:

$$ADVT^{\mathcal{P}}_{\mathfrak{A}}(tme) \leq \tfrac{q^2_{hash}}{|HASH|} + 2max\{C.q^s_{read}, \tfrac{q_{read}}{s^{ln}}\}$$

B. Informal Security Evaluation

In this subsection, we test the competency of our developed protocol against potential security attacks.

C. Anonymity and Untraceability

In order to gain the actual identity UID_i of $\mathbb{U}_i$, $\mathfrak{A}$ can intercept public messages. However, in the developed protocol, $\mathbb{U}_i$ exchanges its pseudonym PID_i instead of UID_i over unreliable channel. Moreover, each parameter in every public communicated message encompasses random nonce, making them distinguishable in all sessions. Thus, $\mathfrak{A}$ can never know the real identity of $\mathbb{U}_i$ nor can trace $\mathbb{U}_i$ through traffic analysis. Consequently, CAS protocol protects trace attacks by achieving $\mathbb{U}_i$'s anonymity.

D. User Impersonation Attack

$\mathfrak{A}$ can attempt to compose a real $\mathbb{W}_1 \leftarrow \{PID_i, \psi_i, \varpi_i\}$ in order to impersonate $\mathbb{U}_i$. Referring to Fig. 2, it is clear that $\mathfrak{A}$ must have the knowledge about $\mathbb{U}_i$'s secrets (i.e., UID_i, UPD_i and λ_i which are unavailable to $\mathfrak{A}$. In the absence of $\mathbb{A}_i$'s secrets, $\mathfrak{A}$ can never compose valid $\mathbb{W}_1$ and ultimately can never pass the verification check $\varpi_i \stackrel{?}{=}$. Hence, CAS protocol is immune to $\mathbb{U}_i$ impersonation attack.

E. Cloud of Things Server Impersonation Attack

Suppose $\mathfrak{A}$ attempts to initiate a forged $\mathbb{W}_2 \leftarrow \{\psi_k, \varpi_k\}$ into the network. However, such an attempt will be gone in vain since the construction of $\mathbb{W}_2$ requires $\mathbb{CTS}_k$'s secret key Υ which is inaccessible to $\mathfrak{A}$. Consequently, it is clear that $\mathfrak{A}$ can never compromise the system through $\mathbb{CTS}_k$ impersonation attack.

F. Resistance to a physical attack on $\mathbb{SD}_j$

The sensing devices deployed in the healthcare system are not tamper-proof, and the risk of breaching the physical security of devices is high. In case $\mathfrak{A}$ gets physical access to $\mathbb{SD}_j$ then he can easily read the secret keys stored within captured $\mathbb{SD}_j$. However, in our developed protocol, such an attempt will be thwarted in foil because it employs $\mathbb{PUF}$ which offers a layered immunity against physical threats. Therefore, if $\mathfrak{A}$ wants to launch tampering attacks, it will immediately modify the essential outcome of PUF. Ultimately, with the compromised $\mathbb{SD}_j$, $\mathfrak{A}$ will remain unsuccessful in passing the check $\varpi_j = H(PID_j \| \chi_{i-j} \| \eta_2 \| SK)$. In a nutshell, the designed solution is unaffected from physical attacks on $\mathbb{SD}_j$.

G. Sensing Device Impersonation Attack

Due to the public openness of communication medium, $\mathfrak{A}$ can intercept $\mathbb{W}_3$ during the execution of the key negotiation phase. Later, $\mathfrak{A}$ can attempt to reproduce the valid message using $\mathbb{W}_1$ to fool $\mathbb{U}_i$. However, we can see that the composition of $\mathbb{W}_3$ requires β_j which is available inside the memory of $\mathbb{SD}_j$. As discussed earlier in Section IV-F, $\mathfrak{A}$ can neither compromise $\mathbb{SD}_j$ nor read data from it. Thus, in the absence of necessary parameters, $\mathfrak{A}$ can never produce $\mathbb{W}_3$. Consequently, CAS protocol is immune to $\mathbb{SD}_j$ masquerading attacks.

TABLE II: Running Time of Primitive Cryptographic Operations

Operation	Execution Time		
	Mobile Device	Desktop System	Arduino Device
E_h	0.705 ms	0.036 ms	1.718 ms
E_{pm}	0.292 ms	0.123 ms	0.510 ms
$E_{e/d}$	0.615 ms	0.041 ms	0.925 ms
E_{fe}	0.805 ms	0.030 ms	1.705 ms
T_p	1.365 ms	0.0314 ms	2.423 ms

V. Performance Comparison

This section exhibits the comparative analysis of CAS and competing related protocols Ali et al. [4], Liu et al. [21] and Srinivas et al. [20] in terms of their performance.

TABLE III: Implementation Setup

Attribute	Mobile Device	Desktop System	Arduino Device
Platform	Android OS	Linux	–
System	–	Intel Core i4	Microcontroller:ATmega328
RAM	4 GB	16 GB	SRAM: 2 KB (ATmega328)
Processing Power	1.8 GHZ	2.9 GHZ	16 MHz
IDE	PyCharm	PyCharm	Arduino IDE

A. Computation Cost

The 6G-enabled AIoMT healthcare applications primarily consists of three entities including: $\mathbb{U}_i$, $\mathbb{CTS}_k$ and $\mathbb{SD}_j$. To observe the execution time of the protocols, we have executed the exploited primitive operations for $\mathbb{U}_i$, $\mathbb{CTS}_k$ and $\mathbb{SD}_j$ on specified devices such as mobile, desktop and Arduino, respectively. Table III presents the specification of each device. Whereas assumed specific notations and obtained execution time in millisecond (ms) for each cryptographic primitive are given in Table II. The notations E_h, E_{pm}, $E_{e/d}$, E_{fe} and T_p in Table II represents the execution time of hash function, point multiplication, symmetric encryption/decryption, fuzzy extraction and bi-linear pairing, respectively.

It can be observed that CAS protocol employs nine hash functions E_h at $\mathbb{U}_i$ side. Therefore, the execution time at $\mathbb{U}_i$ side is $(9 \times 0.705) \approx 6.345$ ms. Whereas, it employs eight hash functions at $\mathbb{CTS}_k$ side and consumes $(8 \times 0.036) \approx 0.288$ ms. Similarly, four hash functions are employed at $\mathbb{SD}_j$ side consumes $(4 \times 1.718) \approx 6.872$ ms. Hence, aggregated computation cost of CAS protocol becomes $(6.345 + 0.288 + 6.872) \approx 13.505$ ms. The computation cost of all related protocols [4], [20], [21] is also computed in a similar manner and is shown in Table IV.

B. Communication Cost

Communication cost refers to the number of bits exchanged during the authentication and key establishment phase. For evaluation, we initially assume the size of each output of employed primitive operations like hash function, elliptic curve point, identity, random number, timestamp, and symmetric key cipher (AES with 256-bit block cipher and 128-bit key size) as 256, 160, 160, 160, 160, and 128 bits, respectively. These values and the numbers of messages exchanged with/from involved entities are observed to determine the communication cost. For instance, the messages $\mathbb{W}_1 \leftarrow \{PID_i, \psi_i, \varpi_i$, $\mathbb{W}_2 \leftarrow \{\psi_k, \varpi_k$ and $\mathbb{W}_3 \leftarrow \{\psi_j, \varpi_j$ are exchanged during authenticated key establishment phase. As per the assumptions for the output of primitive operations, we can determine the number of bits consumed by each message in following way: $\mathbb{W}_1 \leftarrow \{PID_i, \psi_i, \varpi_i : 160 + 160 + 256 = 576$, $\mathbb{W}_2 \leftarrow \{\psi_k, \varpi_k : 160 + 256 = 416$ and $\mathbb{W}_3 \leftarrow \{\psi_j, \varpi_j : 160 + 256 = 416$, respectively. Hence, the aggregated bits exchanged by CAS protocol are $576 + 416 + 416 = 1408$ in bits. Similarly, the communication cost of related protocols [4], [20], [21] is observed in the same fashion (See Table V).

C. Security Features Analysis

This section highlights the security features enrichment of CAS protocol as compared to related protocols [4], [20], and [21]. We have designed CAS protocol in a way to

TABLE IV: Computation Overhead Analysis (in milliseconds)

Schemes	$\mathbb{U}_i$ Side	$\mathbb{CTS}_k$ Side	$\mathbb{SD}_j$	Accumulative Computation Overhead
CAS	$9E_h \approx 6.345$	$8E_h \approx 0.288$	$4E_h \approx 6.872$	13.505
[4]	$16E_h \approx 11.28$	$10E_h \approx 0.36$	$12E_h \approx 20.616$	32.256
[20]	$7E_h + 2E_{fe} + 1E_p \approx 7.91$	$4E_h \approx 0.144$	$5E_h + 1E_{pm} \approx 11.013$	19.067
[21]	$8E_h + 2E_{pm} \approx 6.224$	$10E_h \approx 0.36$	$5E_h \approx 6.872$	14.814

TABLE V: Communication Overhead Analysis (in bits)

Schemes	$\mathbb{U}_i$ Side	$\mathbb{CTS}_k$ Side	$\mathbb{SD}_j$	Accumulative
CAS	576	416	416	1408
[4]	736	1184	1184	3104
[20]	896	1056	992	2944
[21]	992	1856	672	3520

TABLE VI: Analysis of Security Features

Protocols → **Security Features ↓**	CAS	[4]	[21]	[20]
Resists Physical Attacks	✓	✗	✗	✗
Anonymity and Untracability	✓	✓	✓	✓
Resists Mobile User Impersonation Attack	✓	✓	✗	✓
Resists Cloud of Things Server Impersonation Attack	✓	✓	✗	✓
Resists Sensing Device Impersonation Attack	✓	✓	✗	✓
Ensures Perfect Forward and Backward Secrecy	✓	✗	✗	✗

Note: ✓Provided; ✗Not Provided

prevent major security threats (see Table VI). We have already discussed in detail how CAS protocol offers protection against significant security threats (see section IV-B).

D. Discussion

Let us analyze the stats given in the Tables IV, V and VI. We can easily predict that CAS protocol outperforms the related protocols in computation and communication cost analysis. It is worth mentioning that CAS protocol achieves a minimum of 9% and maximum 58% reduction in the computation cost as compared to related protocols [4], [20], [21]. Whereas it achieves at least 52% reduction in the communication cost as compared to related protocols [4], [20], [21]. Table VI shows the enrichment of CAS protocol in terms of security features.

VI. Conclusion

This paper introduced an authenticated solution for 6G-enabled AIoMT healthcare applications. It is specifically developed for resource-constrained infrastructure to facilitate secure and seamless interactions among users. It is thoroughly substantiated and validated through formal and informal security analysis. Furthermore, we compare the performance of the introduced protocol with contemporary related competing protocols. We observed the performance of CAS protocol under the assumption of three metrics, computation, communication, and security features analysis. The comparison reveals that our CAS protocol has outperformed related competing protocol by achieving 8% and 52% more efficiency in terms of computation and communication cost, respectively. Moreover, it promises to offer security enrichment compared to contemporary related protocols.

References

[1] B. Karschnia, "Industrial internet of things (iiot) benefits," *Control Engineering*, 2015.

[2] Q. Chi, H. Yan, C. Zhang, Z. Pang, and L. Da Xu, "A reconfigurable smart sensor interface for industrial wsn in iot environment," *IEEE transactions on industrial informatics*, vol. 10, no. 2, pp. 1417–1425, 2014.

[3] W.-Y. Chung, C.-L. Yau, K.-S. Shin, and R. Myllyla, "A cell phone based health monitoring system with self analysis processor using wireless sensor network technology," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 3705–3708.

[4] J. Srinivas, A. K. Das, N. Kumar, and J. J. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 942–956, 2018.

[5] D. He, N. Kumar, H. Wang, L. Wang, K.-K. R. Choo, and A. Vinel, "A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 633–645, 2016.

[6] C.-T. Li, C.-C. Lee, C.-Y. Weng, and S.-J. Chen, "A secure dynamic identity and chaotic maps based user authentication and key agreement scheme for e-healthcare systems," *Journal of medical systems*, vol. 40, no. 11, pp. 1–10, 2016.

[7] R. Madhusudhan and C. S. Nayak, "A robust authentication scheme for telecare medical information systems," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15 255–15 273, 2019.

[8] S. Qiu, G. Xu, H. Ahmad, and L. Wang, "A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems," *IEEE access*, vol. 6, pp. 7452–7463, 2017.

[9] S. Kumari and K. Renuka, "Design of a password authentication and key agreement scheme to access e-healthcare services," *Wireless Personal Communications*, pp. 1–19, 2019.

[10] S. Barman, H. P. Shum, S. Chattopadhyay, and D. Samanta, "A secure authentication protocol for multi-server-based e-healthcare using a fuzzy commitment scheme," *IEEE Access*, vol. 7, pp. 12 557–12 574, 2019.

[11] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven iot healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.

[12] S. Shamshad, M. S. Obaidat, U. Shamshad, S. Noor, K. Mahmood *et al.*, "On the security of authenticated key agreement scheme for fog-driven iot healthcare system," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021, pp. 1760–1765.

[13] Z. Ali, S. Hussain, R. H. U. Rehman, A. Munshi, M. Liaqat, N. Kumar, and S. A. Chaudhry, "Itssaka-ms: An improved three-factor symmetric-key based secure aka scheme for multi-server environments," *IEEE Access*, vol. 8, pp. 107 993–108 003, 2020.

[14] V. Sureshkumar, R. Amin, M. S. Obaidat, and I. Karthikeyan, "An enhanced mutual authentication and key establishment protocol for tmis using chaotic map," *Journal of Information Security and Applications*, vol. 53, p. 102539, 2020.

[15] T. Limbasiya, S. K. Sahay, and B. Sridharan, "Privacy-preserving mutual authentication and key agreement scheme for multi-server healthcare system," *Information Systems Frontiers*, pp. 1–14, 2021.

[16] V. P. Gaikwad, J. V. Tembhurne, C. Meshram, and C.-C. Lee, "Provably secure lightweight client authentication scheme with anonymity for tmis using chaotic hash function," *The Journal of Supercomputing*, pp. 1–24, 2021.

[17] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[18] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2001, pp. 453–474.

[19] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[20] Z. Ali, A. Ghani, I. Khan, S. A. Chaudhry, S. H. Islam, and D. Giri, "A robust authentication and access control protocol for securing wireless healthcare sensor networks," *Journal of Information Security and Applications*, vol. 52, p. 102502, 2020.

[21] X. Liu, R. Zhang, and M. Zhao, "A robust authentication scheme with dynamic password for wireless body area networks," *Computer Networks*, vol. 161, pp. 220–234, 2019.