Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077157

ARTICLE

A Verifiably Secure and Efficient Authentication Protocol for Resource-Constrained IoT Devices in Cloud-Assisted E-Healthcare

Fahad Algarni[1,2,*] **and Saeed Ullah Jan**[3]

[1]Faculty of Computing and Information Technology, University of Bisha, Bisha, Saudi Arabia
[2]Smart Cyber-Physical System Research Centre, University of Bisha, Bisha, Saudi Arabia
[3]Higher Education Department of Khyber Pakhtunkhwa, Government Degree College Wari (Dir Upper), Wari, Pakistan

*Corresponding Author: Fahad Algarni. Email: fahad.alqarni@ub.edu.sa

Received: 03 December 2025; Accepted: 13 February 2026; Published: 08 May 2026

ABSTRACT: With the increasing connectivity and intelligence of Internet-of-Things (IoT) devices, which interface with numerous aspects of our daily lives, security remains a major concern for IoT devices deployed in e-healthcare systems. The existing solutions demonstrate that authentication of IoT devices across all domains, especially in healthcare, poses significant vulnerabilities, including side-channel, insider, and replay attacks. Alternatively, it is not feasible for resource-constrained IoT devices due to the computational, communicational, and space overheads of modular exponentiation or bilinear pairing, or because it requires four to five round-trips for authentication. The rapid growth of IoT in the e-healthcare sector is expected to cross "50 billion" or more by 2030, highlighting desynchronization, man-in-the-middle (MITM) attacks, and unavailability flaws in e-healthcare. If the aforementioned security concerns are not adequately addressed, they will, in turn, escalate and lead to severe consequences. Therefore, this article introduces a security protocol for an e-healthcare system to ensure secure communication for the voluminous data collected by IoT devices and to transfer it to the cloud safely. The proof of correctness and robustness of the proposed protocol was conducted using BAN (Burrows-Abadi-Needham) logic, the Real-Or-Random (ROR) model, the ProVerif verification toolkit, and pragmatic discussions. The performance analysis section was addressed by measuring several key metrics, including communication, computation, space, and energy consumption, along with scalability. The results obtained demonstrate that the communication cost may be reduced by up to 76%, the computation cost by up to 92%, and the energy consumption by up to 31%.

KEYWORDS: Authentication; confidentiality; logic; cryptography; vulnerability; authentication

1 Introduction

Communications systems are heterogeneous distributed systems explicitly designed for e-healthcare, logistics, complex tactical tasks, or industrial work. A communication system can respond to changes, assess their impact and operation, and react intelligently to complex tactical tasks [1]. It is typically formulated with multiple sensors, IoT devices, or actuators connected to a centralized server, control station, or cloud server [2]. The specified task is achieved through an interconnected system of components, including sensors, wearables, IoT devices, computing nodes, actuators, and the channels that link the communication system [3]. For instance, in the e-healthcare communication system, a feedback loop that attempts to reach patient diagnoses is created by the interaction of wearables, sensors, and IoT devices by recognizing the physiological vitals of the patient's body and transmitting it to the server for examination by the physicians through their mobile device [4]. These interconnected smart devices, whether interacting with the patient's

body or with the physician, including IoT devices, sensors, wearables, or mobile devices, along with servers and communication channels, all belong to the communication system domain [5].

The essential factors that a healthcare communication system should consider include how to successfully implement a security protocol for the secure transmission of data to and from these sensors, wearables, or IoT devices, as well as how to manage data efficiently [6]. The successful implementation of a security system is the backbone of a healthcare communication system, providing support to enable the entire system to function effectively [7]. Besides these, data sensitivity, scalability, intelligence capabilities, and interoperability of these components are also essential [8]. Still, implementing a robust security mechanism poses considerable challenges. It can only be achieved by efficiently authenticating all components of a communication system, including its message with the cloud server, and then with the mobile device of a physician. If these components become authenticated, the remaining issues and challenges can be resolved automatically. Therefore, there is a dire need for an authentication mechanism to ensure that all the connected devices in the healthcare communication system securely communicate with each other and with the cloud server. Therefore, many researchers proposed several authentication mechanisms, which are discussed in detail in the literature section of the article. Despite the many benefits of these mechanisms in the form of offering flexibility, mobility, and cost-effectiveness in our daily lives, they still present a series of issues and challenges described as follows:

- The existing solutions for healthcare communication systems either suffer from side-channel [9], MITM [10], and replay attacks [11], are not feasible for resource-constrained IoT due to modular exponentiation or bilinear pairing, or are completed in four to five round trips [12].
- Within the next ten years, approximately half a trillion sensors, wearables, or IoT devices are anticipated. This unprecedented growth in IoT devices is creating a significant problem for cloud servers, particularly in terms of managing their access to the healthcare communication system.
- Additionally, due to the limitations of low-power communication technologies, the lack of synergy, and the integration of hundreds of thousands of sensors, wearables, or IoT devices, latency is a significant issue for security protocols in the existing literature [13].
- The sensors, wearables, or IoT devices in the healthcare communication system are resource-constrained in nature [14]. Energy consumption is a major issue for them, as frequent battery changes become unsustainable, making the security protocol vulnerable to numerous vulnerabilities like ephemeral secret leakage [15], session key disclosure [16], and traceability attacks [17].

Moreover, as stated in detail, security is a major concern for the healthcare communication system, especially those assisted by cloud computing and driven by IoT devices, due to the lack of sufficient security measures and the presence of strong adversaries. Vulnerability identification in existing solutions, misuse by hardware vendors, and the lack of significant network features to integrate embedded IoT devices with cloud servers necessitate the design of a robust security mechanism. It is worth noting that the said design is a complex task that requires careful attention and planning, as a minor lapse could create a significant hurdle for the system. Given the urgency of the situation and the importance of IoT in the healthcare communication system, the said efforts would answer these questions, including (a) How will the IoT devices securely authenticate with each other, with the physician's mobile device, and with the cloud server, and (b) how patient trust be managed on the IoT devices, sensors, and wearables on the physician and with the cloud server. Hence, keeping these questions in mind, the key contributions of this research work are as follows:

- To propose an authentication mechanism based on a simple hash cryptographic function that offers robust authentication to all the components of the e-healthcare communication system and works efficiently by securely transmitting data to and from these components to the cloud server. To integrate the SHA256 hashing algorithm for reliability and high data availability.

- The proof of correctness and robustness analysis of the proposed authentication scheme is scrutinized through a well-known and widely used BAN authentication logic, RoR model, ProVerif 2.03 validation, and a pragmatic discussion.
- To test the efficacy and feasibility by evaluating the performance metrics through a testbed research method that utilized the Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL), an outstanding open-source Software Development Kit (SDK) written in the C/C++ programming language. The testing was performed on a Windows 10 Pro system equipped with an Intel Core™ i7-6500U CPU (2.50 GHz, with a boost up to 2.59 GHz), 16 GB of RAM, and a 128 GB SSD.
- To comparatively analyze the other performance trade-offs, including energy consumption vs. task offloading, latency vs. runtime, runtime vs. energy consumption, and throughput vs. execution time, for checking its real-world implementation.

The remainder of this paper is organized as follows: Section 2 presents a detailed literature review of various schemes/existing solutions, in which prior works are critically analyzed. Section 3 outlines the proposed system model, threat model, and design goals. In Section 4, the proposed protocol is designed, while Section 5 analyzes its security using BAN logic and ProVerif validation. Section 6 presents the performance evaluation and comparative analysis. Finally, the last section concludes the work presented in this article.

2 Related Works & Problem Definition

Masud et al. [9] proposed a key derivation function-based protocol for data transmission to the cloud server in the e-healthcare system. They [9] argued that e-healthcare records through cloud computing technology would be secure if their security mechanism were implemented. However, they [9] don't analyze the security of their scheme, so no one can say whether their [9] scheme is robust and feasible. Padmaja and Seshadri [10] used MD5 and the chaotic method for crypto-hashing of the e-healthcare record in the cloud server but suffered from a hash collision attack. Chandrakar et al. [11] used an asymmetric method, bilinear mapping, and the SHA1 technique to design a conditional privacy protection scheme for remote patient monitoring. However, their scheme [11] is vulnerable to side-channel and traceability attacks and performs poorly. Deebak and Al-Turjman [12] used a bio-hashing method along with pairing-based cryptography by designing a protocol for a cloud-driven medical system, but due to the exponentiation function, their scheme [12] consumed more energy and had high computation cost, which is not feasible for resource-constrained IoT applications. Chiou et al. [13] also utilized a bilinear mapping method to design the authentication scheme and claimed that their scheme preserves patient privacy, achieving unlinkability and successfully reducing computation and communication costs. However, it doesn't resist traceability, spoofing, and session key hijacking attacks. Qadir and Hussan [14] proposed the security secret key provider (SSKP) model to improve data security and privacy in the cloud server, but their [14] proposed model, while a significant step to the cloud-based e-healthcare system, still does not allow patients to access their medical records.

Okikiola et al. [15] used symmetric encryption, decryption, and watermark extraction techniques for a cloud-based healthcare system to detect unauthorized activity and illegitimate changes in medical records. They [15] highlighted the significance of resisting an insider threat in the e-health record by proposing a strategy that includes a logging system and watermark extraction; OpenNebula, Microsoft Azure, PHP, and MySQL were utilized to implement their proposed security model. Benil and Jasper [16] proposed a model for protecting patient-sensitive data on a cloud server, successfully improved the security of a medical cloud server, and extensively tested for a case study; however, their model has the limitation of low performance. Jan et al. [17] recommended a hybrid cryptosystem-based security protocol for the Internet of Medical Things

(IoMT) of the healthcare system, while [18] proposed a simple hash-based security scheme for IoT-driven e-healthcare. However, these schemes [17,18] are vulnerable to insider threats and identity theft attacks and have traceability issues. The remaining summary of the literature review is shown in Table 1.

Table 1: Summary of the remaining literature review.

Ref.	Technology/Domain	Limitations
Ahmim et al. [19]	Internet of Healthcare Things (IoHT)	• Side channel attack
Anandhi and Sangari [20]	ECC for Cloud-based e-healthcare system	• Formally not proved through a widely adopted technique.
Tanveer et al. [21]	IoT and mobile device	• Password guessing and ESL attacks
Mir and Nikooghadam [22]	Telemedicine information systems for e-healthcare	• Replay attack
Ni et al. [23]	Telemedicine information systems for e-healthcare	• Replay and tracking attacks
Yu and Park [24]	IoT and Telemedicine information systems for e-healthcare	• MITM attack • Lack of mutual authentication
Zheng et al. [25]	Telemedicine and e-healthcare system	• Insider and DoS Attacks
Li et al. [26]	Telemedicine information systems for e-healthcare	• Poses anonymity and confidentiality
Mohit et al.'s [27]	Cloud-based e-healthcare system	• Side channel and forgery attacks
Amin et al. [28]	Telemedicine system	• Eavesdropping, and impersonation attacks
Setianto et al. [29]	JavaScript Object Notation Remote Procedure Call (JSON-RPC) for remote patient monitoring	• Traceability, MITM attacks
Son et al. [30]	WSN-based TMIS through wearables	• Side-channel and traceability attack
Lei and Chuang [31]	Biometric fuzzy extractor and Telemedicine system	• Side channel and impersonation attack

Furthermore, Hamed and Yassin [32] employed OTP (one-time password) and SHA-2 to secure the e-healthcare record on a cloud platform, but it didn't preserve the privacy of the patients and is susceptible to an insider attack. Yao et al. [33] combined biometrics, PUF, and passwords with the ECC technique to securely validate the security mechanism for patient records. Lee et al. [34] introduced a PUF-based authentication mechanism for patient-sensitive information protection; however, it also has a privacy issue and is vulnerable to a side-channel attack. Zhang et al. [35] proposed an IoT-driven authentication mechanism that combines SHA1, asymmetric cryptography, and biometrics but doesn't resist a hash collision attack. Sun et al. [36] employed a blockchain-based data-driven trust mechanism that leveraged the RSA algorithm to exchange medical information securely; however, it is vulnerable to a masquerade attack in

which an attacker can easily impersonate the cloud server by showing themselves as a legitimate user. Sunitha et al. [37] presented an asymmetric encryption-based system and a three-factor authentication scheme to guarantee the secure exchange of medical records to the cloud server. However, their scheme [37] lacks privacy issues and is susceptible to a stolen verifier attack.

Therefore, considering the extensive analysis of the existing solutions available in the literature [13–17], it has been concluded that these schemes either suffer from side-channel, MITM, and replay attacks, are not feasible for resource-constrained IoT due to modular exponentiation or bilinear pairing, or are completed in four to five round trips.

Problem Formulation

Very recently, a cloud-centric authentication scheme [38] has been presented to protect the same environment utilizing Elliptic Curve Cryptography (ECC). The previous scheme [38] consisted of five phases: setup, registration, authentication, sensor addition, and revocation. During the authentication phase, the patient and cloud server, the physician and cloud server, and the patient and physician are authenticated and exchange messages that are vulnerable to various attacks, including side-channel, DoS, and MITM attacks. These are explained one by one as follows:

Side Channel Attack: If an attacker captures these messages $\{HID_P, PK_{PH1}, X_{PA}, PK_{U1}, T_3\}$, $\{HID_{CS}, PK_{PH2}, X_{CS1}, PK_{U2}, T_7\}$, and $\{HID_{PA}, PK_{PA2}, X_{PA2}, T_{11}\}$, they can easily compute $PK_P = R_P \oplus h(HID_P||R_P||PK_P)$, $PK_{V1} = S_{PH}||(X_{PH} \oplus n_4)||PK_P$, $PK_{W1} = n_4.X_{PH,}$ $K_{CS} = R_{PH} \oplus h(HID_{CS}||R_{CS}||PK_{CS})$, $PK_{V2} = S_{CS}||(X_{CS} \oplus n_5)||PK_{PH}$, $PK_{W2} = n_5.$ $X_{CS},$:$PK_{CS2} = R_{PA} \oplus h(HID_{PA}||R_{CS}||PK_{CS})$, $PK_{V3} = S_{CS2}||(X_{CS2} \oplus r_6)||PK_{CS2}$, $PK_{W3} = r_6.$ X_{CS2} and reach for calculating the session secret key $SK = h(PK_{V1}||PK_{W1})$, $SK = h(PK_{V3}||PK_{W3})$, and $SK = h(PK_{V2}||PK_{W2})$. Because the keys are not random, they remain static for every new session. If an attacker copies the publicly transmitted messages, they can easily launch a side-channel attack on the system. Therefore, Ref. [38] is vulnerable to side-channel attacks.

Man-in-the-Middle (MITM) Attack: Due to the static parameters and fixed keys in [38], the attacker can execute the MITM attack on the previously proposed scheme for the same environment. The attacker can easily intercept and computes $PK_{CS2} = R_{CS} \oplus h(HID_{PC}||R_{PA}||PK_{PA})$, $PK_{S3} = (S_{PA}||X_{PA2} \oplus n_6)||PK_{PA2})$, $PK_{HP2} = R_{PH} \oplus h(HID_{PH}||R_{PH}||PK_{PH1})$, $PK_{S2} = (S_{PH}||(X_{PH} \oplus r_5)||PK_{HP2})$, $PK_{PH1} = R_{PH} \oplus h(HID_{PC}||R_{PH}||PK_{PH})$, and $PK_{S1} = (S_P||(X_{PA} \oplus r_4)||PK_{PH1})$ messages between participants, resend old messages, and modify the stored parameters with ease.

DoS Attack: As the scheme presented in [38] relies on static parameters, such as fixed keys and hardcoded credentials, and lacks dynamic parameters, it is highly susceptible to a denial-of-service (DoS) attack. By flooding the patient with malicious requests, the attacker can interfere with normal operations and cause system failure. They can easily compute $PK_{CS2} = R_{CS} \oplus h(HID_{PC}||R_{PA}||PK_{PA})$, $PK_{S3} = (S_{PA}||X_{PA2} \oplus n_6)||PK_{PA2})$, $PK_{HP2} = R_{PH} \oplus h(HID_{PH}||R_{PH}||PK_{PH1})$, $PK_{S2} = (S_{PH}||(X_{PH} \oplus r_5)||PK_{HP2})$, $PK_{PH1} = R_{PH} \oplus h(HID_{PC}||R_{PH}||PK_{PH})$, $PK_{S1} = (S_P||(X_{PA} \oplus r_4)||PK_{PH1})$, $PK_P = R_P \oplus h(HID_P||R_P||PK_P)$, $PK_{V1} = S_{PH}||(X_{PH} \oplus n_4)||PK_P$, $PK_{CS2} = R_{PA} \oplus h(HID_{PA}||R_{CS}||PK_{CS})$, $PK_{V3} = S_{CS2}||(X_{CS2} \oplus r_6)||PK_{CS2}$, $PK_{CS} = R_{PH} \oplus h(HID_{CS}||R_{CS}||PK_{CS})$, and $PK_{V2} = S_{CS}||(X_{CS} \oplus n_5)||PK_{PH}$, and reach the internal secrets or disturbed the normal operation of the system. Hence [38] is vulnerable to a DoS attack.

3 System Architecture

In this section of the article, the system architecture can be modeled by explaining the key concepts related to the system, including all possible threats in the threat model, and design goals for presenting

efficient and effective solutions for such a vulnerable environment. So far, these basic ideas and system model or network model have been explained as follows:

3.1 Network Model

The system or network model presented here in the research work consisted mainly of three participating entities, namely sensors/wearables or IoT devices inside the patient's body or worn by the patient for the collection of physiological vitals from the body of the patient, mobile device of paramedical staff or used by a physician for real-time monitoring of the patient, and a cloud server. The participating entities are explained below, while the diagrammatic representation of the system/network model is depicted in Fig. 1.

(1) **Sensor/Wearable or IoT Device:** The sensor/wearable or IoT device, a key element, might be either implanted inside the patient's body or embedded in the affected part of the body, and plays a crucial role in sensing the physiological condition of a patient. The collected sensitive information, like blood pressure, heart rate, ECG, EEG, blood oxygen, and blood circulation, etc. are transmitted into the cloud server through a wireless channel. These sensors/wearables or IoT devices are resource constrained means consume very low energy, have small latency, and limited bandwidth. Only a lightweight and efficient security mechanism is feasible for it to perform appropriately because it has a small processor and Electrically Erasable Programmable Read Only Memory (EEPROM).

(2) **Physician Mobile Device:** A smartphone or mobile device plays a crucial role in the proposed healthcare communication system and is considered an essential part of the proposed network model. Smartphones and tablets, the backbone of remote patient monitoring in the cloud-assisted e-healthcare system, enable real-time viewing of patient data from multiple linked sensors, wearables, or IoT devices through the cloud server. The smartphone is connected to a specified cloud-based e-healthcare system, displays patient data from associated monitoring IoT devices, equipment, sensors, and wearables, and the physician can easily diagnose the patient. The cloud system ensures that vital signs and other appropriate health indicators/sensors or wearables are updated in real-time for the physicians to examine, enhancing the efficiency of healthcare delivery. If a patient's readings deviate from a signal range, the mobile devices provide immediate patient interaction through an alarm or a message.

(3) **Cloud Server:** A cloud server, while providing scalability and affordability compared to conventional physical infrastructure, serves as a central, easily accessible center for handling and storing data in an e-healthcare system. A cloud server enables remote care examinations, where healthcare professionals can access patient data and conduct examinations from any location with internet access. It also allows for monitoring patients remotely, allowing for continuous tracking of patient health metrics. The cloud server also optimizes treatment processes across several healthcare professionals, ensures the sound and efficient handling of patient health-related records, scans, and other private information, and offers flexibility and mobility. It generates security parameters, cryptographic keys, and other credentials, hosts the e-healthcare system, stores sensitive patient data, maintains physician records, provides real-time patient monitoring facilities, and ensures patient privacy for the entire healthcare communication system.

3.2 Threat Model

We have adopted an extended Dolev-Yao (DY) adversarial model [39] enhanced to capture fully active and insider threats. The adversary $\mathbb{A}$ has the following capabilities:

1. **Network Adversary:** An adversary $\mathbb{A}$ can modify a message in transit, inject new things into the message, delete or delay a message, or compromise the whole session. Also, the adversary $\mathbb{A}$ can

act as a malicious operator for the server, compromise the sensor, or rogue either the patient or the physician device.

2. **Insider Adversary:** The adversary $\mathbb{A}$ might compromise a legitimate entity like a sensor, a physician device, or a cloud server by extracting stored credentials, but not the long-term secrets, which are protected by hardware or might act as a malicious insider by compromising the cloud server.
3. **Cryptographic Assumptions:** The one-way hash cryptographic function h(.) is collision-resistant and behaves as a random oracle; the different random numbers generated at different round-trip are secure and unpredictable, while the long-term secrets d_S are stored in tamper-resistant memory where an attack is not possible.
4. **Adversarial Goals:** The adversary $\mathbb{A}$ tries to get SK, impersonate any legal entity, break the confidentiality, cause a desynchronization anomaly or launch a DoS attack on the system.

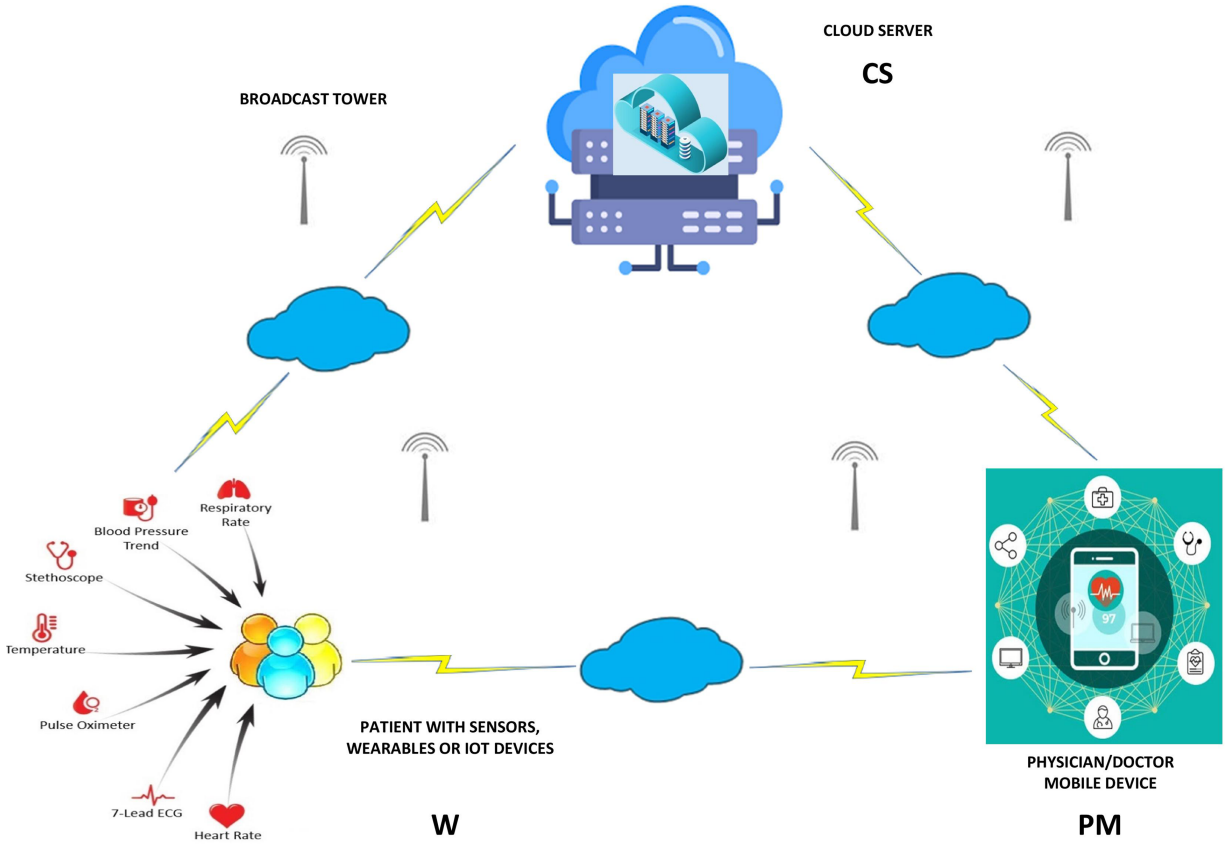

Figure 1: Proposed network model.

3.3 Design Goals

After designing the proposed protocol for the e-healthcare communication system, which works through the cloud and is assisted by IoT, the following design goals should be achieved.

(i) **Confidentiality:** The proposed protocol is designed to maintain the privacy of patient-sensitive information proactively. Patient sensors, wearables, or IoT devices registered and utilized by a patient for monitoring will be allowed to access the cloud server for which these devices are functionalized. Only a legitimate mobile device will be used by the physician to access the designated patient data

from the cloud server. The system is designed to promptly detect and discard most illegal attempts to access the system.

(ii) **Integrity:** The proposed protocol is robust in maintaining the legitimacy of patient-sensitive information. When receiving patient data, the cloud server is designed to promptly discard any illegal modifications and ensure the integrity of the data. Only legitimate doctors/physicians should be allowed to access the cloud server, and any unauthorized attempts will be promptly detected and discarded. This robust protection ensures the integrity of patient data, physician sessions with the patient, and cloud server availability.

(iii) **Availability:** The proposed protocol is reliable in maintaining the availability of patient data. Using this security mechanism, the sensors can securely exchange patient-sensitive data, either bandwidth-limited, low-latency, and low-powered wireless networking channels or powerful ones, to the cloud server. The cloud server incorporates any number of legitimate sensors, wearables, or IoT devices and grants access permissions from authentic doctors/physicians to access patient data, which are properly registered, and their records are available in the cloud server. Unauthorized attempts to access patient-sensitive information are expected to be discarded, and permission should not be granted under normal conditions. The proposed protocol also ensures the availability of all the resources associated with the proposed e-healthcare communication system to access data securely. It enables effective network reconfiguration when a patient goes out of the system.

4 Proposed Protocol

The protocol involves three main entities, including Wearables (W), Physician Mobile Devices (PM), and the Cloud Server (CS). The proposed protocol is based purely on simple hash functions, random nonces, timestamps, and simple bitwise operations. For clarity, Table 2 summarizes the key symbols that h(.) denotes a one-way hash function, $\oplus$ represents XOR, and $||$ denotes concatenation. Timestamps like T_W, T_{PM}, and T_{CS} and random numbers like r_1, r_2, and r_3 that can ensure freshness and resist replay attacks.

Table 2: Symbols/notations and their descriptions.

Notation	Description	Notation	Description
ID_W	Sensor Identity	Δ	Matching Algorithm
ID_{PM}	Physician Identity	d_S	Cloud Server secret key
PID_{PM}	Physician Pseudo-density	r_1, r_2, r_3	Random numbers
$\|\|$	Concatenation function	T_W	Patient side Timestamp
SK	Session Key	T_{PM}	Physician side Timestamp
T_C	Current Timestamp	T_{CS}	Cloud Server side Timestamp

4.1 Setup Phase

The cloud server chooses a one-way hash function h(.): $\{0, 1\}^* \rightarrow \{0, 1\}^l$, secret key d_S, publishes $\{h(.), d_S\}$, and keeps d_S as secret key.

4.2 Registration Phase

The registration phase is completed in the following two sub-phases:

(1) **Sensor Registration:** Upon registering the sensor/wearable or IoT device, the following steps of computation are performed:

Step 01: The cloud (CS) server selects a unique identity ID_W, a random number r_1, and in a secure manner transmits $\{ID_W, r_1\}$ to the sensor over a secure channel, and keeps $\{ID_W, r_1\}$ in its memory.

Step 02: When receiving $\{ID_W, r_1\}$, the sensor stores for the record, as shown in module 1.

Module 1: Sensor Registration.

Cloud Server (CS)	Sensor (W)
Select: ID_W, r_1	
$\{ID_W, r_1\} \longrightarrow$	
Store: $\{ID_W, r_1\}$	Store:$\{ID_W, r_1\}$

(2) **Device Registration: This sub-phase is completed in the following two steps:**

Step 01: Now registering the mobile device, the cloud server selects a unique identity ID_{PM}, pseudo-identity PID_{PM}, computes $Q_1 = h(PD_{PM}||d_S)$, $Q_2 = h(ID_{PM}||d_S)$, and transmits $\{ID_{PM}, PID_{PM}, Q_1, Q_2\}$ to the mobile device and keeps its record in its memory.

Step 02: When receiving $\{ID_{PM}, PID_{PM}, Q_1, Q_2\}$ message, the mobile device securely stores these parameters in its memory for later authentication purposes, as shown in module 2.

Module 2: Registration Phase.

Cloud Server (CS)	Mobile Device (PM)
Select: ID_{PM}, PID_{PM}	
Compute: $Q_1 = h(PID_{PM}||d_S)$	
$Q_2 = h(ID_{PM}||d_S)$	
$\{PID_{PM}, ID_{PM}, Q_1, Q_1\} \longrightarrow$	
	Store: $\{PID_{PM}, ID_{PM}, Q_1, Q_2\}$

4.3 Authentication Phase

This is the most important phase of the protocol, through which all three entities agree securely to a single session key. This phase is completed in the following computation steps.

Step 01: The mobile device user selects a random number, r_2, records the current timestamp T_{PM}, and performs a series of computations to generate S_1, S_2, and S_3. These values are then sent as a message $\{PID_{PM}, T_{PM}, S_1, S_2, S_3\}$ towards the sensor over a public channel.

Step 02: Upon receiving the $\{PID_{PM}, T_{PM}, S_1, S_2, S_3\}$ message, the sensor's role becomes crucial. It undergoes a meticulous verification process, checking the time $T_c - T_{PM} \leq \Delta T$ to ensure the absence of any potential replay/DoS attack. This verification process is a key step in maintaining the protocol's security. If the message is validated, the sensor proceeds to the next step, choosing a random number r_3 and recording the present timestamp T_W, computing $U_1 = h(r_1||T_W) \oplus r_3$, $U_2 = h(PID_{PM}||ID_W||S_1||S_2||T_{PM}||T_W||r_3)$ and sends $\{PID_{PM}, ID_W, S_1, S_2, S_3, U_1, U_2, T_{PM}, T_W\}$ message to the cloud server over a public channel.

Step 03: The cloud server, when receiving $\{PID_{PM}, ID_W, S_1, S_2, S_3, U_1, U_2, T_{PM}, T_W\}$ message, validates the received times (both T_{PM} and T_W) with the system current time $T_c - T_{PM} \leq \Delta T$ and $T_C - T_W \leq \Delta T$, if it doesn't found with the pre-defined time threshold, the process is terminated, otherwise, the cloud server computes $r_3 = U_1 \oplus h(r_1||T_W)$, $U_2{}^* = h(PID_{PM}||ID_W||S_1||S_2||T_{PM}||T_W||r_3)$, and confirms $U_2{}^*? = U_2$, if become not validated, the process will be terminated, otherwise, computes $ID_{PM} = S_1 \oplus h(Q_1||r_2)$, $S_3{}^* =$

$h(PID_{PM}||ID_{PM}||r_2||T_{PM})$, and again confirms $S_3^{*?} = S_3$ and again if not confirmed, the process will stop and terminate, else, the cloud server selects a pseudo-identity PID_{PM2}, records time T_{CS} and computes the session key $SK = h(ID_{PM}||ID_W||r_2||r_3||T_{PM}||T_W)$. Further the cloud server computes $W_1 = h(r_3||r_1||T_{CS}) \oplus SK$, $W_2 = h(ID_W||r_3||T_{CS}||SK)$, $W_3 = h(ID_{PM}||r_2||T_{CS}) \oplus SK$, $W_4 = h(ID_{PM}||r_2||T_{CS}) \oplus PID_{PM2}$, $W_5 = h(ID_{PM}||r_2||T_{CS}) \oplus h(PID_{PM2}||d_S)$, $W_6 = h(ID_{PM}||PID_{PM2}||r_2||SK||h(d_S||PID_{PM2}||T_{CS})$ and sends $\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}\}$ message back to the sensor over a public channel.

Step 04: The sensor when receiving $\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}\}$ message, it first checks $T_W - T_{CS} \leq \Delta T$, if validated, computes $SK = W_1 \oplus h(r_3||r_1||T_{CS})$, $W_2^{*} = h(ID_W||r_3||T_{CS}||SK)$, and confirm $W_2^{*?} = W_2$ if matched, sends $\{W_3, W_4, W_5, W_6, T_{CS}\}$ message towards the user over a public channel.

Step 05: The user, upon receiving $\{W_3, W_4, W_5, W_6, T_{CS}\}$ message, it first check $T_W - T_{CS} \leq \Delta T$, if validated, computes $SK = W_3 \oplus h(ID_{PM}||r_2||T_{CS})$, $PID_{PM2} = W_4 \oplus h(ID_{PM}||r_2||T_{CS})$, $Q_1^{new} = W_5 \oplus h(ID_{PM}||r_2||T_{CS})$, $W_6^{*} = h(ID_{PM}||PID_{PM2}||r_2||SK||h(d_S||PID_{PM2}||T_{CS})$ confirms $W_6^{*?} = W_6$ and replaces Q_1 with Q_1^{new} and PID_{PM} with PID_{PM2}, keeps SK as session secret key, as shown in module 3.

Module 3: Authentication Phase.

Mobile Device (PM)	**Sensor (W)**	**Cloud Server (CS)**
Select: r_2 and Record T_{PM} Compute: $S_1 = h(Q_1||r_2) \oplus ID_{PM}$ $S_2 = Q_2 \oplus r_2$, $S_3 = h(PID_{PM}||ID_{PM}||r_2||T_{PM})$ $\{PID_{PM}, T_{PM}, S_1, S_2, S_3\} \rightarrow$		
	Verify: T_{PM}, Choose: r_3, and Record: T_W Compute: $U_1 = h(r_1||T_W) \oplus r_3$ $U_2 = h(PID_{PM}||ID_W||S_1||S_2||T_{PM}||T_W||r_3)$ $\{PID_{PM}, ID_W, S_1, S_2, S_3, U_1, U_2, T_{PM}, T_w\} \rightarrow$	
		Verify: T_{PM} and T_W Compute: $r_3 = U_1 \oplus h(r_1||T_W)$ $U_2^{*} = h(PID_{PM}||ID_W||S_1||S_2||T_{PM}||T_W||r_3)$ Confirm: $U_2^{*?} = U_2$ Compute: $ID_{PM} = S_1 \oplus h(Q_1||r_2)$ $r_3 = U_1 \oplus h(r_1||T_W)$ $S_3^{*} = h(PID_{PM}||ID_{PM}||r_2||T_{PM})$ Check: $S_3^{*?} = S_3$ Select: PID_{PM2} and Record T_{CS} Compute: $SK = h(ID_{PM}||ID_W||r_2||r_3||T_{PM}||T_W)$ $W_1 = h(r_3||r_1||T_{CS}) \oplus SK$ $W_2 = h(ID_W||r_3||T_{CS}||SK)$ $W_3 = h(ID_{PM}||r_2||T_{CS}) \oplus SK$ $W_4 = h(ID_{PM}||r_2||T_{CS}) \oplus PID_{PM2}$ $W_5 = h(ID_{PM}||r_2||T_{CS}) \oplus h(PID_{PM2}||d_S)$ $W_6 = h(ID_{PM}||PID_{PM2}||r_2||SK||h(d_S||PID_{PM2}||T_{CS})$
	$\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}\} \leftarrow$	
	Verify: T_{CS} Compute: $SK = W_1 \oplus h(r_3||r_1||T_{CS})$ $W_2^{*} = h(ID_W||r_3||T_{CS}||SK)$ Confirm: $W_2^{*?} = W_2$	
$\{W_3, W_4, W_5, W_6, T_{CS}\} \leftarrow$		
Verify: T_{CS} Compute: $SK = W_3 \oplus h(ID_{PM}||r_2||T_{CS})$ $PID_{PM2} = W_4 \oplus h(ID_{PM}||r_2||T_{CS})$ $Q_1^{new} = W_5 \oplus h(ID_{PM}||r_2||T_{CS})$ $W_6^{*} = h(ID_{PM}||PID_{PM2}||r_2||SK||h(d_S||PID_{PM2}||T_{CS})$ Confirm: $W_6^{*?} = W_6$ Replace: Q_1 with Q_1^{new} and PID_{PM} with PID_{PM2} and Keep: SK as session secret key		

4.4 Algorithmic Representation

The algorithmic overview of this crucial phase of the protocol is demonstrated in Algorithm 1, which depicts the flow of information and random checks in the scheme. These conditions are a clear indication of its robustness against attacks; the implementation code is given in Appendix A of the article.

Algorithm 1: Authentication of PM, W, and CS

Input: ID_{PM}, r_2, T_{PM}, ID_W, r_3, T_W, PID_{PM}, T_{CS}

Output: $SK=W_3 \oplus h(ID_{PM}||r_2||T_{CS})$, $PID_{PM2}=W_4 \oplus h(ID_{PM}||r_2||T_{CS})$, and $Q_1^{new}=W_5 \oplus h(ID_{PM}||r_2||T_{CS})$

1: $S_1=h(Q_1||r_2) \oplus ID_{PM}$, $S_2=Q_2 \oplus r2$, $S_3=h(PID_{PM}||ID_{PM}||r_2||T_{PM})$

2: if $T_c - T_{PM} \leq \Delta T$ then

3: $U_1=h(r_1||T_W) \oplus r_3$, $U_2=h(PID_{PM}||ID_W||S_1||S_2||T_{PM}||T_W||r_3)$

4: if $T_C - T_W \leq \Delta T$

5: $r_3=U_1 \oplus h(r_1||T_W)$, $U_2^*= h(PID_{PM}||ID_W||S_1||S_2||T_{PM}||T_W||r_3)$

6: if $(U_2^* ? = U_2)$

7: $ID_{PM}=S_1 \oplus h(Q_1||r_2)$, $r_3=U_1 \oplus h(r_1||T_W)$, $S_3^*= h(PID_{PM}||ID_{PM}||r_2||T_{PM})$

8: if $S_3^* ? = S_3$

9: $SK=h(ID_{PM}||ID_W||r_2||r_3||T_{PM}||T_W)$

10: if $T_W - T_{CS} \leq \Delta T$

11: Repeat step 10

12: if $T_{PM} - T_{CS} \leq \Delta T$

13: Repeat step 10

14: Return (SK) Pass

15: else

16: Return (0) Failed

17: Return (SK) Pass

18: else

19: Return (0) Failed

20: Return (SK) Pass

21: else

22: Return (0) Failed

23: Return (SK) Pass

24: else

25: Return (0) Failed

26: Return (SK) Pass

27: else

28: Return (0) Failed

29: Return (SK) Pass

30: else

31: Return (0) Failed

32: Exit

5 Proof of Correctness & Robustness

This section scrutinizes the proof of correctness and robustness of the proposed IoT-driven and cloud-assisted authentication protocol, which is accomplished through a well-known and widely used BAN logic [40], Random Oracle Model [41], ProVerif toolkit [42], and informal analysis, as follows:

5.1 BAN Logic Analysis

The BAN logic, introduced by Burrows et al. [40], is a systematic approach to protocol analysis. Its proof of correctness and security analysis is particularly adept at identifying potential flaws in a protocol, keeping us vigilant, or demonstrating its validity in terms of authentication. The logic consists of different notations and symbols (described in Table 3), different statements, postulates, and goals for checking the exchange of random numbers, keys, and hash codes. These are explained one by one for the proposed protocol in a systematic order as follows:

Table 3: BAN logic notations, their pronunciations and descriptions.

Notation/Statement Rules/Formulas	Pronounced/Named As	Description/Explanation
$\mid\sim$	Once Said	Participant A Once Said Participant B like A$\mid\sim$ B
#	Fresh	Participant A believes the freshness of message M like (A$\mid\equiv$#(M))
$\mid\equiv$	Believes	If Participant A believes Participant B, and both A and B believes on the freshness of message M like A$\mid\equiv$ B, AB$\mid\equiv$ #(M)
$\Rightarrow$	Jurisdiction	Participant A has complete jurisdiction over key K like A $\Rightarrow$
$\triangleleft$	Sees	If A sees message M, and B sees message M, and both A and B sees message M like A$\triangleleft$M, B$\triangleleft$M, AB$\triangleleft$M
<.>	Combined	A combine the message of A and B like <M_A, M_B>
$(.)_K$	Encryption	A encrypt message M via key K like M_K
$1/(.)_K$	Decryption	B decrypt message M via key K like $1/M_K$
$\frac{W\mid\equiv\left(W\xrightarrow{r_2}PM\right)\triangleleft<M>}{W\mid\equiv PM\Rightarrow M}$	Message Meaning Rule	Suppose the patient-side sensor believes that message communication between a sensor and mobile device is performed via r_2 and sees message M, and *vice versa*.
$\frac{W\mid\equiv\left(W\xrightarrow{r_2}PM\right)\triangleleft<M>_K}{W\mid\equiv PM\mid\sim M}$	Verification Rules	Suppose the patient-side sensor believes that message communication between a sensor and mobile device is performed via r_2 and sees message M. In that case, the sensor thinks the mobile device once sees message M.
$\frac{W\mid\equiv\#(M), W\mid\equiv PM\mid\sim M}{W\mid\equiv PM\Rightarrow M}$	Jurisdiction Rules	Suppose the patient-side sensor believes that message M is fresh and believes on mobile device once sees message M. In that case, the sensor thinks the mobile device has jurisdiction over message M.
$\frac{W\mid\equiv\#(M)}{PM\#(M)}$	Freshness Rules	Suppose the patient-side sensor believes that message M is fresh, then mobile device also thinks the freshness of message M.

5.1.1 Message Content

The transmission of different messages among different participating entity during the authentication phase of the protocol is presented in message content form as follows:

$$W\xrightarrow{\{PID_{PM,T_{PM},S_1},S_{2,S_3}\}}PM \tag{1}$$

$$PM\xrightarrow{\{PID_{PM,ID_W,T_{PM},S_1},S_{2,S_3,T_W}\}}CS \tag{2}$$

$$\mathrm{CS}\underrightarrow{\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}, \}}PM \tag{3}$$

$$\mathrm{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W \tag{4}$$

5.1.2 Idealization Form

The representation of numerous parameters into the rules defined by BAN logic or simply converting them into different messages, statements, and credentials in a form that expresses the belief and intentions of the participant. It is an essential step because it allows us to build trust, freshness, and authentication when checking its implementation and ensuring correctness, as well as running syntax. The idealized form of the proposed protocol is represented as follows:

$$\mathrm{PM}| \equiv \mathrm{r}_2,\ \mathrm{PM}| \equiv \mathrm{T}_{\mathrm{PM}}, \mathrm{PM}| \equiv \mathrm{d}_{\mathrm{S}} \tag{5}$$

$$\mathrm{W}| \equiv \mathrm{r}_3, \mathrm{W}| \equiv \mathrm{T}_{\mathrm{W}}, \mathrm{W}| \equiv \mathrm{d}_{\mathrm{S}} \tag{6}$$

$$\mathrm{CS}| \equiv \mathrm{r}_1, \mathrm{r}_2, \mathrm{r}_3 \tag{7}$$

$$\mathrm{CS}| \equiv \mathrm{T}_{\mathrm{PM}}, \mathrm{T}_{\mathrm{W}}, \mathrm{T}_{\mathrm{CS}} \tag{8}$$

$$\mathrm{CS}| \equiv \mathrm{ID}_{\mathrm{PM}}, \mathrm{ID}_{\mathrm{W}}, \mathrm{PID}_{\mathrm{PM}} \tag{9}$$

$$\mathrm{CS}| \equiv \mathrm{d}_{\mathrm{S}} \tag{10}$$

5.1.3 Goals

It means to determine whether the protocol achieves the security features like mutual authentication, freshness, key secrecy, and confidentiality through the use of assumptions, rules, principles, identification, etc., and derive conclusions to check the proposed protocol for replay attack and trust violation or not, and obtain the core objectives, etc. So, the goals set for achieving are shown as follows:

$$W| \equiv (\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM) \tag{11}$$

$$PM| \equiv (\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM) \tag{12}$$

$$PM| \equiv (\mathrm{PM}\underrightarrow{\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}}CS) \tag{13}$$

$$CS| \equiv (\mathrm{PM}\underrightarrow{\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}}CS) \tag{14}$$

$$CS| \equiv (\mathrm{CS}\underrightarrow{\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}, \}}PM) \tag{15}$$

$$PM| \equiv (\mathrm{CS}\underrightarrow{\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}, \}}PM) \tag{16}$$

$$PM| \equiv (\mathrm{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W) \tag{17}$$

$$W| \equiv (\mathrm{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W) \tag{18}$$

5.1.4 Assumptions

The different keys, random numbers, nonces, etc., that form the foundation and derive the conclusion of scheme correctness, while mutually authenticating each other. It also explains how beliefs evolve during the shared session key computation and ensures that the proof is based on these assumptions. The assumptions for the proposed protocol are expressed as follows:

$$W| \equiv \mathrm{PM}| \equiv (\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM) \tag{19}$$

$$PM| \equiv \mathrm{W}| \equiv (\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM) \tag{20}$$

$$PM| \equiv \mathrm{CS}| \equiv (\mathrm{PM}\underrightarrow{\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}}CS) \tag{21}$$

$$CS| \equiv \mathrm{PM}| \equiv (\mathrm{PM}\underrightarrow{\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}}CS) \tag{22}$$

$$CS| \equiv \mathrm{PM}| \equiv (\mathrm{CS}\underrightarrow{\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}, \}}PM) \tag{23}$$

$$PM| \equiv \mathrm{CS}| \equiv (\mathrm{CS}\underrightarrow{\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}, \}}PM) \tag{24}$$

$$PM| \equiv \mathrm{W}| \equiv (\mathrm{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W) \tag{25}$$

$$W| \equiv \mathrm{PM}| \equiv (\mathrm{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W) \tag{26}$$

$$\mathrm{CS}| \equiv (\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM) \tag{27}$$

$$\mathrm{CS}| \equiv \mathrm{PM}| \equiv (\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM) \tag{28}$$

$$\mathrm{CS}| \equiv \mathrm{PM}| \equiv \mathrm{W}| \equiv (\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM) \tag{29}$$

$$W| \equiv PM| \equiv \mathrm{CS}| \equiv (\mathrm{PM}\underrightarrow{\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}}CS) \tag{30}$$

$$PM| \equiv \mathrm{W}| \equiv (\mathrm{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W) \tag{31}$$

$$PM| \equiv \mathrm{W}| \equiv \mathrm{CS}| \equiv (\mathrm{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}\}}W) \tag{32}$$

5.1.5 Proof

For the proof of correctness of the protocol, the message contents, idealization, and assumptions will be used for achieving the goals. So, according to according to messages (1) and (23), the following result will be achieved.

$$\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM: W| \equiv \mathrm{PM}| \equiv (\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM) \tag{33}$$

Eq. (33) can be written as:

$$W| \equiv \mathrm{PM}| \equiv \left(W\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM\right) : \left(\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM\right) \tag{34}$$

Eq. (34), can be expressed as:

$$\mathrm{W}| \equiv \left(W\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM\right) : \left(\mathrm{W}\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM\right) \tag{35}$$

Eq. (35), means

$\mathrm{W}| \equiv \left(W\underrightarrow{\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}}PM\right)$ Goal1 Eq. (11) Achieved

Now, Eq. (35) becomes

$$\mathrm{PM}| \equiv \left(W\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}PM\right) : \left(\mathrm{W}\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}PM\right) \tag{36}$$

Eq. (36) can be written as

$\mathrm{PM}| \equiv \left(W\{PID_{PM,T_{PM},S_1}, S_{2,S_3}\}PM\right)$ Goal2 Eq. (12) Achieved

Taking Eqs. (3) and (25), we get

$$\mathrm{PM}\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS\, PM| \equiv \mathrm{CS}| \equiv (\mathrm{PM}\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS) \tag{37}$$

Eq. (37), can be written as

$$PM| \equiv \mathrm{CS}| \equiv \left(PM\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS\right) : \left(\mathrm{PM}\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS\right) \tag{38}$$

Eq. (38), can also be expressed as

$$PM \equiv \left(PM\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS\right) : \left(\mathrm{PM}\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS\right) \tag{39}$$

Eq. (39) means

$PM \equiv \left(PM\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}CS\right)$ Goal3 Eq. (13) Achieved

Taking Eq. (38), we get

$$\mathrm{CS}| \equiv \left(PM\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS\right) : \left(\mathrm{PM}\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS\right) \tag{40}$$

Eq. (40) can also be expressed as

$\mathrm{CS}| \equiv \left(PM\{PID_{PM,ID_W,T_{PM},S_1}, S_{2,S_3,T_W}\}CS\right)$ Goal4 Eq. (14) Achieved

Taking Eqs. (5) and (27), we get

$$\mathrm{CS}\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS},\}PM{:}\, CS| \equiv \mathrm{PM}| \equiv (\mathrm{CS}\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS},\}PM) \tag{41}$$

Eq. (41), can also be expressed as

$$CS| \equiv \mathrm{PM}| \equiv (\mathrm{CS}\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS},\}PM) : (\mathrm{CS}\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS},\}PM) \tag{42}$$

Eq. (42), means

$$CS| \equiv \left(CS\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}\}PM\right) : \left(CS\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}\}PM\right) \tag{43}$$

Eq. (43) can also be represented as

$$CS| \equiv \left(CS\{\underrightarrow{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}, \}PM}\right) \text{Goal5 Eq. (15) Achieved}$$

Taking Eq. (43), we get

$$\text{PM}| \equiv \left(CS\{\underrightarrow{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}}\}PM\right) : \left(CS\{\underrightarrow{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}}\}PM\right) \tag{44}$$

Eq. (44) means

$$\text{PM}| \equiv \left(CS\{\underrightarrow{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}, }\}PM\right) \text{Goal6 Eq. (16) Achieved}$$

Taking Eqs. (7) and (29), we get

$$\text{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}\}}W : PM| \equiv \text{W}| \equiv (\text{PM}\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}\}}W) \tag{45}$$

Eq. (45) can be written as

$$PM| \equiv \text{W}| \equiv \left(PM\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}\}}W\right) : \left(PM\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}\}}W\right) \tag{46}$$

Eq. (46), means

$$PM| \equiv \left(PM\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W\right) : \left(PM\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W\right) \tag{47}$$

Eq. (47), can be written as

$$PM| \equiv \left(PM\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W\right) \text{Goal7 Eq. (17) Achieved}$$

Taking Eq. (46), we get

$$\text{W}| \equiv \left(PM\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W\right) : \left(PM\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W\right) \tag{48}$$

Eq. (48), means

$$\text{W}| \equiv \left(PM\underrightarrow{\{, W_3, W_4, W_5, W_6, T_{CS}, \}}W\right) \text{Goal8 Eq. (18) Achieved}$$

Therefore, the BAN has verified the mutual authentication and freshness of the nonce, SK, and identities, not to prove resistance to all attacks. Hence, the proposed IoT-driven, and cloud centric security protocol is correct, and feasible for real-world e-healthcare system.

5.2 Limitations of BAN Logic and Complementary Analysis

In our design, the Dolev-Yao threat model [39] (Section 3.2) automatically takes into consideration the presence of an active attacker with total control over the network. Although BAN logic [40] (Section 5.1) is a formal methodological tool that verifies the authenticity of the designed protocols in an ideal state, this tool does not automatically assume the presence of multiple parallel protocol instances (concurrent sessions) or an adaptive attacker who can react during the protocol execution with the goal of breaching the protocol security. This weakness in BAN logic can be further overpowered by the random-oracle model (ROM) [41] analysis (Section 5.3) and an automated security protocol verification tool called "ProVerif" [42] (Section 5.4) that can formally verify the security of the designed protocols in the presence of multiple

parallel protocol instances and an adaptive attacker. This tool shows the correctness of our designed protocol in providing a secure transmission of the data through the "session key" (SK) with the goal of securely covering the data transmission between the two parties. Meanwhile, the designed protocol provides the secrecy of the SK used in the secure transmission of the data. Also, the proposed protocol ensures mutual authentication (authentication of both parties at Sections 5.3 and 5.4). Moreover, the proposed protocol successfully resists the threats of "man-in-the-middle" attacks (the introduction of the attacker in the data transmission). Additionally, the proposed protocol uses "timestamps" (time indicators of the protocol messages) to guarantee the presence of each unique "session" with the goal of addressing the weakness of the adapted BAN logic methodology. It also generates multiple sessions that will have different session keys, due to using "dynamic pseudo-identities." It means that the BAN logic proof in this work is used to establish belief consistency and authentication correctness, while resistance to concrete attacks such as MITM, replay, forgery, and insider attacks is separately analyzed through Random Oracle Model (ROM), ProVerif verification, and informal security analysis in the subsequent sections.

5.3 Analysis through RoR Model

The proposed protocol $\mathbb{P}$ is a symmetric based scheme; it has no public key, so we will now adopt the widely accepted Real-Or-Random (ROR) model [41] and connect analysis for the proposed protocol $\mathbb{P}$ with the DY-threat model [39]. The probabilistic polynomial-time adversary $\mathbb{A}$ and the birthday paradox [43] will be used for the security assumption in the following manner.

Participants: Let Π_{PM}^{i} be the ith instance of physician device (PM), Π_{W}^{j} be the jth instance of sensor/wearable (W), and Π_{CS}^{k} be the kth instance of cloud server (CS), respectively, then there exist three states i.e., accepted state in which the three instances successfully computes the SK, rejected state means not computed, and null ($\perp$)-state means nothing is output while running the protocol $\mathbb{P}$.

Partnering: Suppose two instances, Π_l and Π_m, are be the partners of $\mathbb{P}$, which either generated the same session key SK and share the identities to become partners of each other.

Freshness: If the SK computed by instance Π_l and not revealed to $\mathbb{A}$, then it means that it is fresh, because $\mathbb{A}$ has complete control over the public channel and he/she is executing the following queries:

- ***Execute*** $(\boldsymbol{\Pi_l, \Pi_m})$ **:** An adversary $\mathbb{A}$ eavesdrops on the device and sensor to play a passive attacker.
- ***Send*** $(\boldsymbol{\Pi_l, m})$**:** The adversary $\mathbb{A}$ sends a message m to Π_l and the output tuple they get shall be compared with the SK.
- ***Reveal*** $(\boldsymbol{\Pi_l})$ **:** Adversary $\mathbb{A}$ gets SK of Π_l.
- ***Corrupt*** $(\boldsymbol{\Pi_l})$ **:** $\mathbb{A}$ gets long-term secret of $\mathbb{P}$.
- ***Test*** $(\boldsymbol{\Pi_l})$ **:** Adversary $\mathbb{A}$ in this query test the SK they get while running different queries by flipping a coin; if they get 1—succeeded, 0—failed, $\perp$(Null)—blank.

Theorem 1: *Under the oracle, the SK is indistinguishable from a random string to the adversary* $\mathbb{A}$.

Proof: The session key SK = $h(ID_{PM}||ID_W||r_2||r_3||T_{PM}||T_W)$, which is established from random nonce r_2, r_3 and is unknown to the adversary $\mathbb{A}$. It means the input to the hash (h(.)) is unique per every session with overwhelming probability. So, in the oracle, the output of h(.) is a random string, so different from the $\mathbb{A}$'s random string. Hence, the proposed protocol's SK is indistinguishable. □

Semantic Security: Let $\mathbb{A}$ act as a polynomial-time attacker against the proposed authentication protocol for the CS, has λ parameter. The ADV (advantage) with $\mathbb{A}$ in breaching $\mathbb{P}$ is shown by:

$$ADV_{\mathbb{P}}(\mathbb{A}) \leq \frac{(Q_H)^2}{2^{l_h}} + \frac{(Q_S)}{2^{l_r}} + \frac{(Q_S)}{|D|} + ADV_{ECDLP}(t) \tag{49}$$

Theorem 2: *Under the oracle, P provides mutual authentication.*

Proof: The authentication depends on U_2, S_3, W_2, W_6, and each one is masked by a hash for every individual session, along with a random nonce, and a timestamp. Hence, by running any query for finding collision of the preimages of h(.) is negligible under ROM.

In Eq. (49), the Q_H stands for "hash queries", Q_S for "send queries", l_h is "length of hash output", l_r is "length of random number", |D| is "size of password dictionary", and $ADV_{SHA256}(t)$ is $\mathbb{A}$'s advantage in breaking Secure Hash Algorithm 256 in time t. $\mathbb{A}$ goes through different games which are mentioned as follows:

Game G_0: $\mathbb{A}$ plays this game for physically interacting $\mathbb{A}$ as shown in Eq. (50). □

$$ADV_{\mathbb{P}}(\mathbb{A}) = |2Prob[S_0] - 1| \tag{50}$$

Theorem 3: *The server secret key d_S does not reveal the past SK.*

Proof: SK depends on r_2 and r_3, which are discarded after the session is accomplished, so the knowledge of d_S alone is not a guarantee for SK.

Game G_1: $\mathbb{A}$ plays this game by acting a eavesdropper and is shown in Eq. (51).

$$|Prob[S_1] - Prob[S_0]| \leq ADV_{ECDLP}(t) + \frac{(Q_H)^2}{2^{l_h}} \tag{51}$$

Game G_2: $\mathbb{A}$ plays this game for forging SK which is shown in Eq. (52). □

$$|Prob[S_2] - Prob[S_1]| \leq \frac{(Q_S)}{2^{l_r}} \tag{52}$$

Theorem 4: *The replay of an old message or malicious attack is not valid for P.*

Proof: The time stamp TPM, TCS, and TW, nonces r_3, r_4, and different checks at each round trip not only ensure freshness but also prohibit the adversary $\mathbb{A}$ from replaying some old message. Such a reply to an old message failed the time checks, while a malicious attack failed by the hash chain in P.

Game G_3: $\mathbb{A}$ plays this game to reach the ephemeral random values is depicted in Eq. (53).

$$|Prob[S_3] - Prob[S_2]| \leq \frac{(Q_S)}{2^{l_r}} \tag{53}$$

Game G4: $\mathbb{A}$ plays this game for guessing the password or random number is shown by Eq. (54).

$$|Prob[S_4] - Prob[S_3]| \leq \frac{(Q_S)}{|D|} \tag{54}$$

Game G5: $\mathbb{A}$ play this game for reaching the secret key d_s is shown in Eq. (55).

$$|Prob[S_5] - Prob[S_4]| \leq ADV_{ds}(t) \tag{55}$$

Eq. (55) can be written as:

$$Prob[S_5] = \frac{1}{2} \tag{56}$$

Combining Eqs. (50) to (56), we get

$$ADV_{\mathbb{P}}(\mathbb{A}) \leq \frac{(Q_H)^2}{2^{l_h}} + \frac{(Q_S)}{2^{l_r}} + \frac{(Q_S)}{|D|} + ADV_{SHA256}(t) + ADV_{ds}(t) \tag{57}$$

Eq. (57) demonstrated that $\mathbb{A}$ cannot break the semantic security of $\mathbb{P}$. □

5.4 ProVerif Simulation

A software verification toolkit called ProVerif [42] is used to check the robustness of the proposed protocol. The methodology of using this toolkit is shown in Fig. 2, which demonstrates that an attacker couldn't crack the session secret key, a man-in-the-middle attack is not possible on the proposed protocol, and the confidentiality and reachability of the session secret are preserved. The code is shown in Appendix B of the article.

```
-- Process 1-- Query not attacker(SK[]) in process 1
Translating the process into Horn clauses...
Completing...
200 rules inserted. Base: 190 rules (24 with conclusion selected). Queue: 14 rules.
Starting query not attacker(SK[])
RESULT not attacker(SK[]) is true.
-- Query event(end_CS(x)) ==> event(begin_PM(x)) in process 1
Translating the process into Horn clauses...
Completing...
200 rules inserted. Base: 192 rules (24 with conclusion selected). Queue: 8 rules.
Starting query event(end_CS(x)) ==> event(begin_PM(x))
RESULT event(end_CS(x)) ==> event(begin_PM(x)) is true.
-- Query event(end_PM(x)) ==> event(begin_CS(x)) in process 1
Translating the process into Horn clauses...
Completing...
200 rules inserted. Base: 187 rules (24 with conclusion selected). Queue: 7 rules.
Starting query event(end_PM(x)) ==> event(begin_CS(x))
RESULT event(end_PM(x)) ==> event(begin_CS(x)) is true.

------------------------------------------------------------
Verification summary:

Query not attacker(SK[]) is true.

Query event(end_CS(x)) ==> event(begin_PM(x)) is true.

Query event(end_PM(x)) ==> event(begin_CS(x)) is true.

------------------------------------------------------------
```

Figure 2: ProVerif simulation result summary.

The ProVerif model [42] implements an active adversary $\mathbb{A}$ capable of intercepting, modifying, and forging messages on public channels, so the verification summary confirms that even under this strong adversary model, the SK remains secret, mutual authentication holds, and MITM are prevented. Also, insider threats are modeled by allowing the adversary $\mathbb{A}$ to possess certain long-term credentials; the results show that the protocol remains secure under these conditions.

5.5 Informal Analysis

The proposed protocol $\mathbb{P}$ can pragmatically be discussed for different attacks and security features in showing its robustness and feasibility. These are explained as follows:

(1) **Offers Mutual Authentication:** The $U_2 = h(PID_{PM}||ID_W||S_1||S_2||T_{PM}||T_W||r_3)$ at the physician side confirms at the cloud side $U_2^* = h(PID_{PM}||ID_W||S_1||S_2||T_{PM}||T_W||r_3)$, the $S_3^* = h(PID_{PM}||ID_{PM}||r_2||T_{PM})$ at the patient side confirms at the cloud side $S_3 = h(PID_{PM}||ID_{PM}||r_2||T_{PM})$, the $W_2 = h(ID_W||r_3||T_{CS}||SK)$ at the cloud side confirms at the patient side $W_2^* = h(ID_W||r_3||T_{CS}||SK)$ and $W_6 = h(ID_{PM}||PID_{PM2}||r_2||SK||h(d_S||PID_{PM2}||T_{CS})$ at the cloud side confirms at the patient side $W_6^* = h(ID_{PM}||PID_{PM2}||r_2||SK||h(d_S||PID_{PM2}||T_{CS})$ demonstrates that all the participating parties mutually authenticate each other.

(2) **Offers Unlinkability, Anonymity, and Untraceability:** The mentioned three security features can be attained for the proposed protocol because the cloud server communicates with the patient and doctor using a pseudonym PID_{PM}. The pseudo-identity PID_i is updated for the current session as well as each upcoming session. There is no interaction between the two sessions if the attacker receives the most recent one from the public network channel. Therefore, the proposed protocol aims to provide unlinkability, anonymity, and untraceability under the assumed threat model.

(3) **Resists Replay Attack:** Let $\mathbb{A}$ captures the first message $\{SID, PK_2, E_2, T_1\}$ which is transmitted between sensor and device and replay some other time, it has to pass $T_2 - T_1 \leq \Delta T$ which is not possible, also the existence of 160-bits long key PK_2 pseudo-identity SID, and $E_2 = r_6 \oplus A_3 \oplus h(PK_2||SID||T_1)$ complex set of calculation doesn't allow the attacker to launch replay attack on the system. And suppose, the attacker captures the second $\{F_2, F_3, F_6, F_7, ID_{de}, T_3\}$ message which is transmitted between device and sensor, the attacker has to pass these random checks $T_4 - T_3 \leq \Delta T$, $F_1^* = F_4 \oplus B_1 \oplus PK \oplus h(k.PK_1)$, $F_1^{*?} = F_1$ which is hard to launch the replay attack. Hence, the proposed protocol is designed to resist replay attacks under typical deployment scenarios.

(4) **Withstands MITM Attack:** Suppose $\mathbb{A}$ sends something towards the device by acting as a malicious user, the device first check the validity of the message with its current time, if lying within the pre-defined time threshold, it allow to verify the identity of sensor ID_{sr}, PK_2 and $E_2 = r_6 \oplus A_3 \oplus h(PK_2||SID||T_1)$, otherwise there any illegal attempt will never be entertained. Similarly, if the attacker sends a fake message toward the sensor, the sensor first check the message with their own time threshold, and then compute $F_1^* = F_4 \oplus B_1 \oplus PK \oplus h(k{\cdot}PK_1)$ and confirms $F_1^{*?} = F_1$ which is looking very hard for the attacker to play the role of man-in-the-middle. Thus, the proposed scheme is designed to withstand MITM attacks under the Dolev-Yao adversary model.

(5) **Provide Perfect Forward Secrecy:** The parameters are concealed with a SHA2 algorithm, through which all the parameters are securely stored in the memory of the sensor, mobile device, and cloud server. They are used by each participating entity for mutual authentication and establishing the shared secret key. The server secret key d_S is entirely inaccessible to any potential attacker. Even if they manage to obtain the random number and secret key from the mobile device, they still be unable to use it. Furthermore, any changes made at the patient end are immediately reflected at the cloud server end.

(6) **Safe against an Insider Attack:** The certificate authority stores $\{PK, G, p, P, q, h(.)\}$, where $k \in Zp^*$, and the public key is 160 bits long. The sensor memory consists of $\{A_2, A_3\}$ calculated from $A_1 = r_1. P$, $B_1 = r_2. P$, $B_2 = r_2 \oplus (k{\cdot}PK)$, $r_2 = B_2 \oplus (k{\cdot}PK)$, computes $A_2 = B_1 \oplus A_1$, $A_3 = k. PK$, while the device memory stores $\{C_2, C_3, r_5, PK_1\}$ parameters, constructed from a complex set of calculations, $C_1 = r_3. P$ $B_3 = r_4. P$, $B_4 = r_4 \oplus (k{\cdot}PK)$, $r_4 = B_4 \oplus (k{\cdot}PK)$, $C_2 = B_4 \oplus r_3$, $C_3 = k{\cdot}PK$. $PK_1 = r_5. P$. Therefore, if an attacker gains control and enters the system to identify useful credentials from these stored parameters, due to the use of a 160-bit key, random numbers, the SHA256 algorithm, and curve points, they cannot succeed

in launching an insider attack. Therefore, the proposed protocol is designed to resist insider threats under the specified security assumptions.

Moreover, if an adversary $\mathbb{A}$ wants to act as an insider, he/she have to find the secret d_S, r_2, r_3, and pseudo-identities PID_{PM} and PID_{PM2}. Let the adversary $\mathbb{A}$ gets the Q_1 and Q_2. Still, he/she cannot compute the SK due to no knowledge of r_2 and r_3 because the server secret d_S is not shared with anyone, and it is safely stored from anyone, limiting the insider. At the same time, the pseudo-identities PID_{PM} and PID_{PM2} provide a shield to the long-term secret.

(7) **Free from Impersonation Attack:** Suppose $\mathbb{A}$ enters the open network channel and tries to get $\{PSID, E_3, F_4, E_5, E_6, T_5\}$ message, they have to computes $PSID = ID_{sr} \oplus h(PK_2||r_9)$, $E_4 = F_4 \oplus B_1 \oplus PK \oplus h(k{\cdot}PK_1)$, $E_5 = PSID \oplus h(r_9||E_4)$, and $E_6 = r_9 \oplus PK \oplus h(k{\cdot}PK_1)||T_5$ which is not possible because each parameters is tightly bounded with other, so attacker cannot impersonate a legitimate user. Therefore, the proposed scheme is designed to resist impersonation attacks under normal operational conditions.

(8) **Stolen Verifier Attack Is Not Valid:** Suppose $\mathbb{A}$ steals the mobile device and retrieves the stored credentials from its memory. The memory of the mobile device contains the $\{C_2, C_3, r_5, PK_1\}$ parameters, which were calculated from a complex set of calculations, including $C_1 = r_3$. P $B_3 = r_4$. P, $B_4 = r_4 \oplus (k{\cdot}PK)$, $r_4 = B_4 \oplus (k{\cdot}PK)$, $C_2 = B_4 \oplus r_3$, $C_3 = k{\cdot}PK$. $PK_1 = r_5$. The attacker has to pass through a complex set of computations to find the 160-bit key, random numbers that are different for each session, a 192-bit secret key, and a curve point that is not fixed. Thus, the proposed protocol is designed to mitigate stolen verifier attacks under the assumed adversary capabilities.

(9) **Resists a Desynchronization Attack:** The cloud server modifies the selection of a pseudo-identity PID_{PM} in the first round and then updates the pseudo-identity in the second round PID_{PM2} and secret key d_S, and then replacing the old pseudo-identity and secret key with the new ones PID_{PM}^{new} and d_S^{new}. This process ensures that even though an attacker attempts to cease communication by flooding data, the communication will continue to function, and the protocol can secure the available data, by maintaining the system form a desynchronization attack.

(10) **Compromise Entity Analysis:** Suppose adversary $\mathbb{A}$ learns ID_W, r_1, and desires to compute SK, he/she cannot, due to no knowledge of r_2 and r_3. If he/she gets PID_{PM}, Q_1, and Q_2, they still cannot implement the CS due to a lack of knowledge of d_S, because the d_S of the server is protected, and the scheme perfectly provides key secrecy.

6 Performance Evaluations and Comparative Analysis

This section of the article will not only measures the performance metrics like computation, communication costs and energy consumption, but also other performance trade-offs and scalability as discussed one by one as under:

6.1 Implementation

The cost for each individual cryptographic operation was measured through a careful assessment process for the proposed IoT-driven and cloud-assisted protocol. The following resources were used to determine computation costs, communication costs, storage overhead, and energy consumption:

1. The Raspberry Pi 5 [44] was chosen as the IoT protocol runner due to its Broadcom BCM2712 2.4 GHz quad-core 64-bit Arm Cortex-A76 CPU, 2 GB of RAM, and dual-band 802.11ac Wi-Fi. These specifications were selected for their compatibility and optimal performance with the MIRACL Crypto SDK using C/C++ programming library [45].

2. The Samsung Galaxy A21s smartphone, with a CPU with 4 cores at 2.0 GHz Cortex-A55 and 4 cores at 2.0 GHz Cortex-A55, is equipped with 6 GB of RAM and an Android operating system.
3. The Core i7 laptop, running Windows 10 Pro, was selected for its specifications, including an Intel® Core™ i7-6500U CPU operating at 2.50 GHz (2.59 GHz burst speed).

The results of running the proposed protocol with different cryptographic operations. Considering the results obtained for the hash cryptography operation, random number extraction, and encryption/decryption operations mentioned in Table 4, the evaluation of various performance metrics like computation, communication, and storage costs is explained below, one by one. It is important to note that because xor and concatenation operations are simple and require little computing power, their execution times are too short—nearly zero—and they have no effect on the overall performance of the system.

Table 4: Computation cost for cryptographic operations.

Operation Name	Symbol	Execution Time in Milliseconds (ms)		
		Laptop	Cellphone	Raspberry Pi
One way hash	T_H	0.149	0.674	0.891
Encryption Function	T_E	0.461	0.851	1.014
Decryption Function	T_D	0.461	0.851	1.014
Extraction of random number	T_{RN}	2.011	2.448	2.946

6.2 Computation Cost Analysis

The total number of cryptographic operations performed by the physician, patient, and cloud sides is detailed as: for the physician's side, the total number of hash functions is 6 (denoted as $6T_H$), the number of XOR operations is 5 (denoted as $5T_\oplus$), and the random number extraction is $1T_{RN}$. Consequently, the cumulative cost at the physician's end can be calculated as: $6T_H + 5T_\oplus + 1T_{RN} = 6(0.674) + 5(0) + 1(2.448) = 4.044 + 0 + 2.448 = 6.492$ ms. For the patient side, the total number of hash functions is $4T_H$, the number of XOR operations is $2T_\oplus$, and the random number extraction remains $1T_{RN}$. Thus, the cumulative cost at the patient end is: $4T_H + 2T_\oplus + 1T_{RN} = 4(0.891) + 2(0) + 1(2.946) = 3.546 + 0 + 2.946 = 6.492$ ms. At the cloud server side, the total number of hash functions is $12T_H$, the number of XOR operations is $7T_\oplus$, and the random number extraction is again noted as $1T_{RN}$. Therefore, the cumulative cost at the server end is: $12T_H + 7T_\oplus + 1T_{RN} = 12(0.149) + 7(0) + 1(2.011) = 1.788 + 0 + 2.011 = 3.799$ ms. Adding the costs calculated for all three participating peers, i.e., physician, patient, and server ends, are given as: $6.492 + 6.492 + 3.799 = 16.783$ ms. This is shown in Table 5, and diagrammatically represented in Fig. 3.

Table 5: Computation cost analysis.

Participant	No. of Operations	Values	Cost in ms
Physician (PM)	$6T_H + 5T_\oplus + 1T_{RN}$	4.044 + 0 + 2.448	6.492
Patient (W)	$4T_H + 2T_\oplus + 1T_{RN}$	3.546 + 0 + 2.946	6.492
Server (CS)	$12T_H + 7T_\oplus + 1T_{RN}$	1.788 + 0 + 2.011	3.799
Total Computation Costs in Milliseconds (ms)			**16.79**

6.3 Storage Cost Analysis

The parameters stored in the memory of different participants includes $\{PID_{PM}, ID_{PM}, Q_1, Q_2\}$ stored by the physician end of cost, according to [46] is 60 + 60 + 256 + 256 = 632 bits, $\{ID_W, r_1\}$ of the Patient end of cost 60 + 32 = 92 bits, and h(.): $\{0, 1\}^* \rightarrow \{0, 1\}^1$, d_S, $\{PID_{PM}, ID_{PM}, Q_1, Q_2\}$m and $\{ID_W, r_1\}$ at the server end of cost 256 + 64 + 60 + 60 + 256 + 256 + 60 + 64 = 1044 bits. The cumulative storage cost for all three participating entities is 1768 bits, as shown in Table 6 and plotted in Fig. 3. This cost is the result of the individual contributions from the physician, patient, and server ends, which are 632, 92, and 1044 bits, respectively.

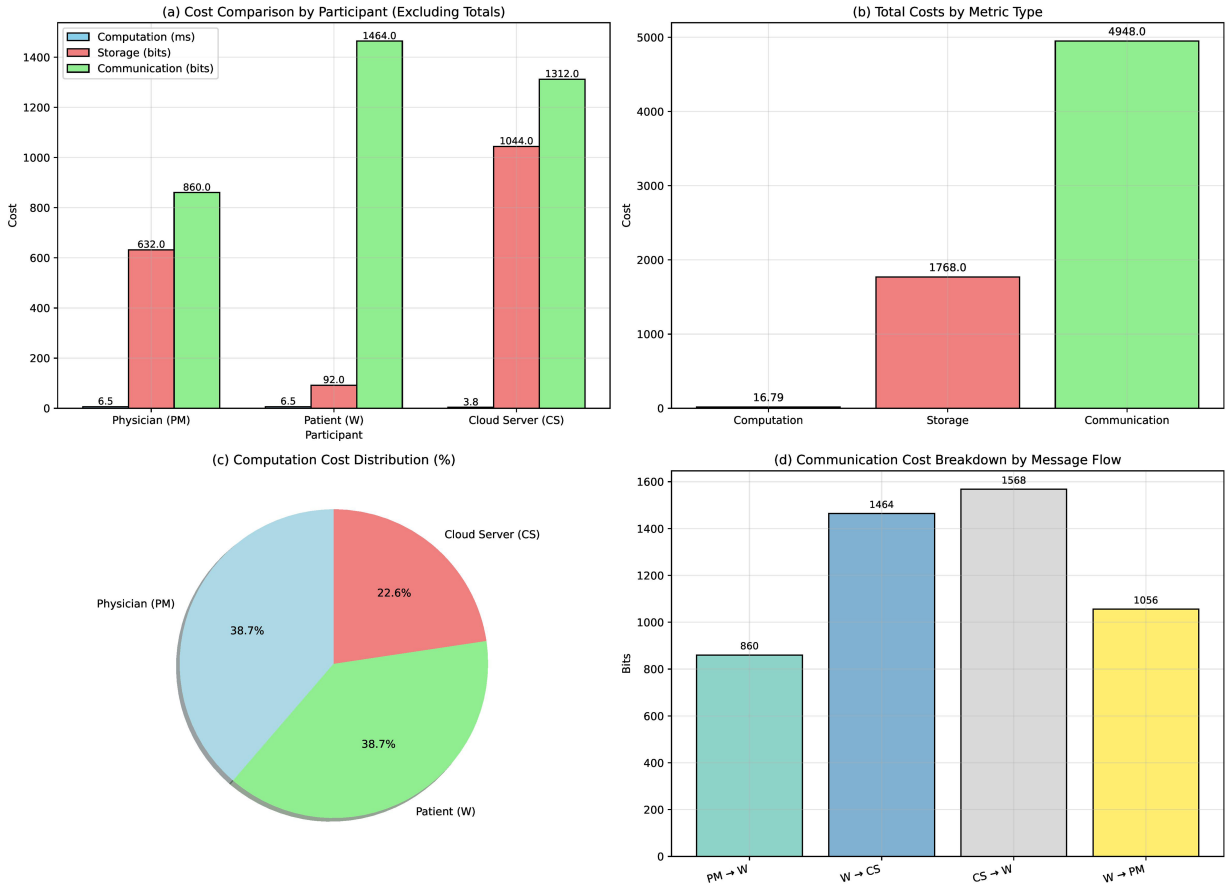

Figure 3: Performance evaluation of the proposed IoT-driven cloud-assisted authentication protocol.

Table 6: Storage cost analysis.

Peer	Credentials	Values	Cost in Bits
Physician (PM)	$\{PID_{PM}, ID_{PM}, Q_1, Q_2\}$	60 + 60 + 256 + 256	632
Patient (W)	$\{ID_W, r_1\}$	60 + 32	92
Cloud Server (CS)	h(.), d_S, $\{PID_{PM}, ID_{PM}, Q_1, Q_2\}$, and $\{ID_W, r_1\}$	256 + 64 + 60 + 60 + 256 + 256 + 60 + 64	1044
Total Storage Cost in Bits			**1768**

6.4 Communication Cost Analysis

The first message transmitted between PM (physician device) and W (patient wearable) consists of the following elements: $\{PID_{PM}, T_{PM}, S_1, S_2, S_3\}$. The total cost of this message is calculated; according to [46] is 60 + 32 + 256 + 256 + 256, which equals 860 bits. The second message transmitted between W (patient wearable) to CS (cloud server) includes the elements: $\{PID_{PM}, ID_W, S_1, S_2, S_3, U_1, U_2, T_{PM}, T_W\}$. The total cost for this message is 60 + 60 + 256 + 256 + 256 + 256 + 256 + 32 + 32, summing up to 1464 bits. The third message sent from CS (cloud server) to W (patient wearable) contains $\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}\}$. Its cost is calculated as 256 + 256 + 256 + 256 + 256 + 256 + 32, resulting in a total of 1568 bits. Finally, the last message transmitted between W (patient wearable) and PM (physician device) includes $\{W_3, W_4, W_5, W_6, T_{CS}\}$, with a total cost of 256 + 256 + 256 + 256 + 32, which equals 1056 bits. When we add up the communication costs for all four messages, we get a cumulative total of 860 + 1464 + 1568 + 1056, amounting to 4948 bits. This detailed breakdown, which is also presented in Table 7 and illustrated in Fig. 3, provides a comprehensive view of the communication costs.

Table 7: Communication cost analysis.

Participants	Message	Values	Cost in Bits
PM → W	$\{PID_{PM}, T_{PM}, S_1, S_2, S_3\}$	60 + 32 + (256 × 3)	860
W → CS	$\{PID_{PM}, ID_W, S_1, S_2, S_3, U_1, U_2, T_{PM}, T_W\}$	60 + 60 + (256 × 5) + 32 + 32	1464
CS → W	$\{W_1, W_2, W_3, W_4, W_5, W_6, T_{CS}\}$	(256 × 6) + 32	1568
W → PM	$\{W_3, W_4, W_5, W_6, T_{CS}\}$	256 + (256 × 3) + 32	1056
Total Communication Cost in Bits			**4948**

6.5 Energy Consumption Analysis

During the implementation phase, the resources consume a specific amount of battery power upon executing the proposed IoT-driven and cloud-assistant authentication protocol. The Raspberry Pi [44], Cellphone, and Laptop utilize some power to perform computation and message communication. $E_C = C_T \times C_P$ represents a wireless communication channel. In contrast, C_T is the computation cost of the proposed IoT-driven and cloud-assistant authentication protocol, which is calculated as 16.783 ms; C_P means the maximum power utilized by the CPU, which is set to be fixed at 10.88 W for the transmission of wireless data [47]. It is worth noting that computation time complexity is directly related to battery power, like a protocol that has fewer operations, which will consume less energy [48]. By inputting these values into the formula, $E_C = 16.783 \times 10.88 = 182.56$ mJ, considering the proposed protocol's energy consumption. This calculation highlights the efficiency of the proposed protocol, reassuring the effectiveness in managing energy consumption.

6.6 Scalability Analysis

Upon running the protocol for one complete round trip and calculating the session secret key (SK), this complex calculation is essentially the round trip time (RTT). The precision of these calculations is not just important; it's paramount. In the following set of calculations, where T_{SK} represents the RTT for the generation of SK, T_{SP} is the beginning time when it set computes, T_{RP} is the complete RTT of the proposed protocol, and T_{RS} is the time of reachability of SK to each participating entity, we ensure our utmost attention to detail shown in Eqs. (58) to (63):

$$T_{SK} = T_{SP} + T_{RP} + T_{RS} \tag{58}$$

Now, the maximum time needed to keep SK as a session secret key is represented as:

$$T_{RS} = \frac{DS}{Wlog_2\left(1+\frac{PD}{N}\right)} \tag{59}$$

DS means data size in bits, W is the bandwidth required for exchanging different credentials, PD means the power of an IoT device, sensor, or wearable, and N means the Gauss noise [49] for energy in the communication channel [47].

$$T_{RP} = \frac{C_P}{C_E} \tag{60}$$

Whereas C means the communication cost of the proposed cloud-assisted protocol, C_E means the time the cloud server takes to run the setup phase for generating the secret credentials.

$$T_{RS} \approx 0 \tag{61}$$

Put (51)–(53) in (50), we get

$$T_{SK} = \frac{DS}{Wlog_2\left(1+\frac{PD}{N}\right)} + \frac{C_P}{C_E} + 0 \tag{62}$$

$$T_{SK} = \frac{DS}{Wlog_2\left(1+\frac{PD}{N}\right)} + \frac{C_P}{C_E} \tag{63}$$

6.7 Performance Trade-Offs Analysis

In the case of a cloud-assisted e-healthcare system, the proposed method has significantly decreased computational costs, making it highly effective for n-number of sensors/IoT devices or wearables and energy consumption, run-time, latency in bps, and task-offloading contrary to ECC-based, PUF-based, Hash-based, Blockchain-based, Three-factor-based, and Hybrid Cryptosystem-based such as [11,12,26,27,50,51] This analysis, which is shown in Figs. 4–6, verifies the scalability of the proposed authentication scheme, which is designed for IoT, cloud servers, and mobile devices, and can process continuous processing at a lower cost. The proposed IoT-driven and cloud-assisted authentication protocol is not only cost-effective but also highly scalable, making it suitable for a wide range of real-world scenarios.

(1) **Energy Consumption vs. Number of Devices:** Fig. 4 shows the energy usage contrary to the number of devices for the proposed technique, as well as several other methods, like ECC-based, PUF-based, Hash-based, Blockchain-based, Three-factor-based, and Hybrid Cryptosystem-based are ineffective for small-scale deployments because they use more energy at lower device counts. The proposed scheme outperforms others with more than 1000 devices, retaining the lowest energy consumption and maximum number of devices as the number of devices rises. At increasing device densities, ECC-based and Three-factor based scale less successfully than the proposed method, and the inadequacy of the scalability of Hash-based and Blockchain-based is a major issue, as their energy usage is excessively high even at intermediate device counts.

(2) **Energy Consumption vs. Number of Task-Offloading:** The energy consumption comparison of the proposed scheme and various existing techniques like ECC-based, PUF-based, Hash-based, Blockchain-based, Three-factor-based, and Hybrid Cryptosystem-based; the result is depicted in Fig. 5,

which demonstrates not only the superior energy efficiency of the proposed method but also its practical implications. Our scheme consistently outperforms others at all task-offloading scales, with more significant benefits at higher workloads. The efficiency for lesser workloads, ECC-based and Hash-based exhibit poor scalability, with their energy usage increasing disproportionately at scale. Also, Hybrid-Cryptosystem based is particularly shown that as the load is distributed it doesn't perform well. The steady performance curve of the proposed approach validates strong optimization; however, the sharper drops for Blockchain-based, Three-factor-based, and Hybrid Cryptosystem-based at larger work volumes make them impractical.

(3) **Execution Time vs. Throughput:** In Fig. 6, the proposed method consistently maintains superior efficiency with the lowest execution times across all throughput levels, especially at higher data rates, where other methods like ECC-based, PUF-based, Hash-based, Blockchain-based, Three-factor-based, and Hybrid Cryptosystem-based show significant performance degradation. The execution times of Three-Factor authentication based and ECC-based grow disproportionately with increasing data rates, suggesting poor scalability despite their moderate efficiency at lower throughputs. Critical issues exist in PUF-based, Hash-based, Blockchain-based, and Three-factor authentication methods, as they exhibit longer processing times than the proposed scheme. Among these, the Hybrid Cryptosystem-based approach is particularly inefficient, demonstrating significantly lower performance. Specifically, when throughput is high, transitioning from a Hybrid Cryptosystem-based to a Three-factor-based authentication model leads to ineffective resource allocation under stress. In contrast, the proposed method's consistent, nearly linear performance curve validates its resilient and optimized design.

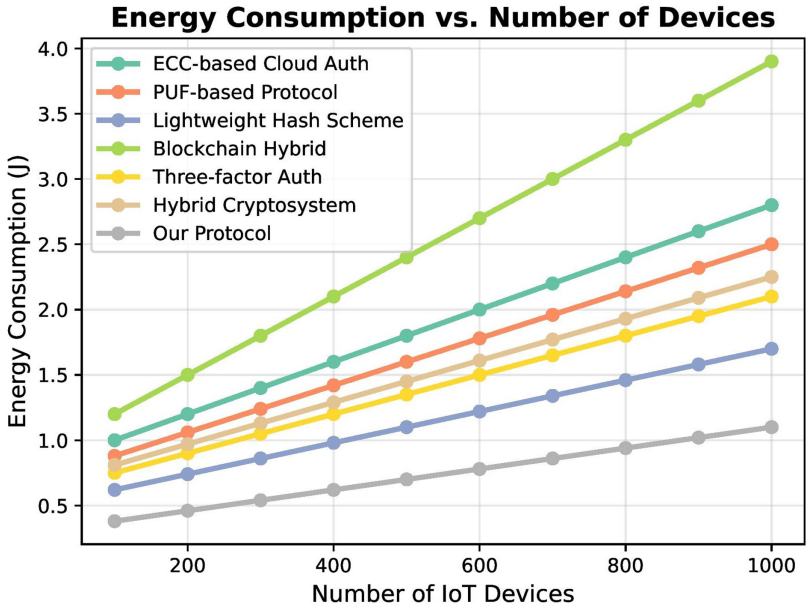

Figure 4: Energy consumption vs. number of devices.

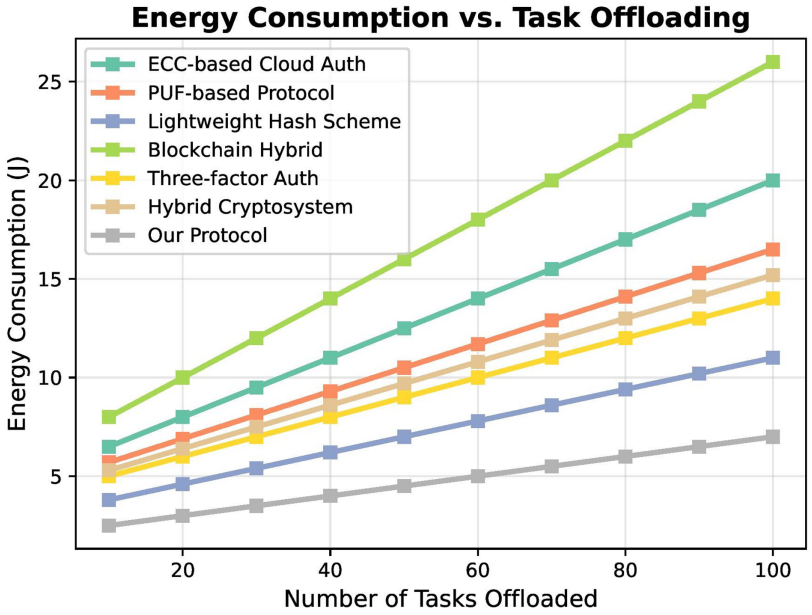

Figure 5: Energy consumption vs. number of task-offloading.

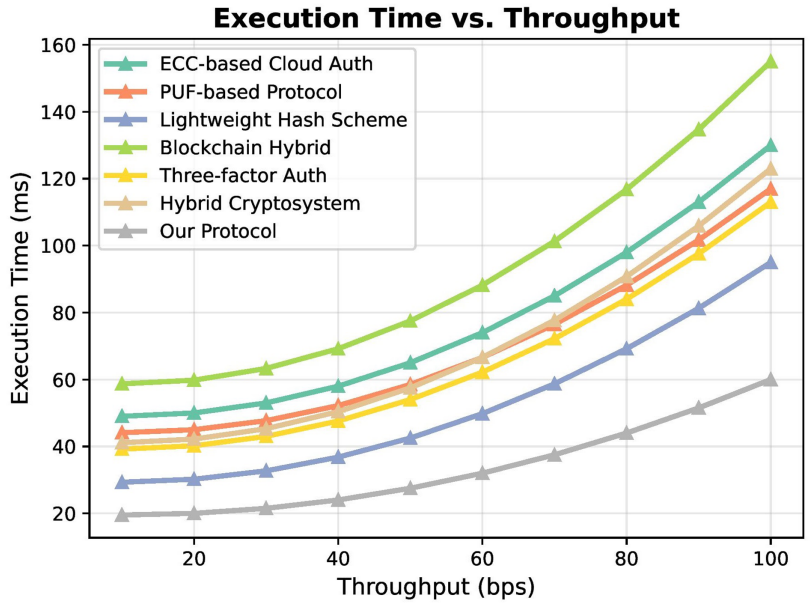

Figure 6: Execution time vs. throughput.

6.8 Comparative Analysis

(1) **Comparative Analysis (Performance Metrics):** When comparing the proposed protocol with state-of-the-art protocols [11,12,26,27,50,51] in terms of communication and computation costs, the result demonstrated in Table 8 showed that the communication cost of [50,51] is slightly better than the proposed authentication scheme; however, the proposed protocol is much better in computation cost from all the mentioned schemes, as shown in Fig. 7.

Table 8: Comparative analysis (Performance Metrics).

Metrics → Protocols↓	Communication Cost in Bits	Computation Cost in ms
Chandrakar et al. [11]	9440	350.3
Deebak and Al-Turjman [12]	7648	1200
Li et al. [26]	5712	77.94
Mohit et al. [27]	5312	208.6
Keshta [50]	4558	27.269
Alghamdi [51]	4408	27.57
Proposed	4948	16.79

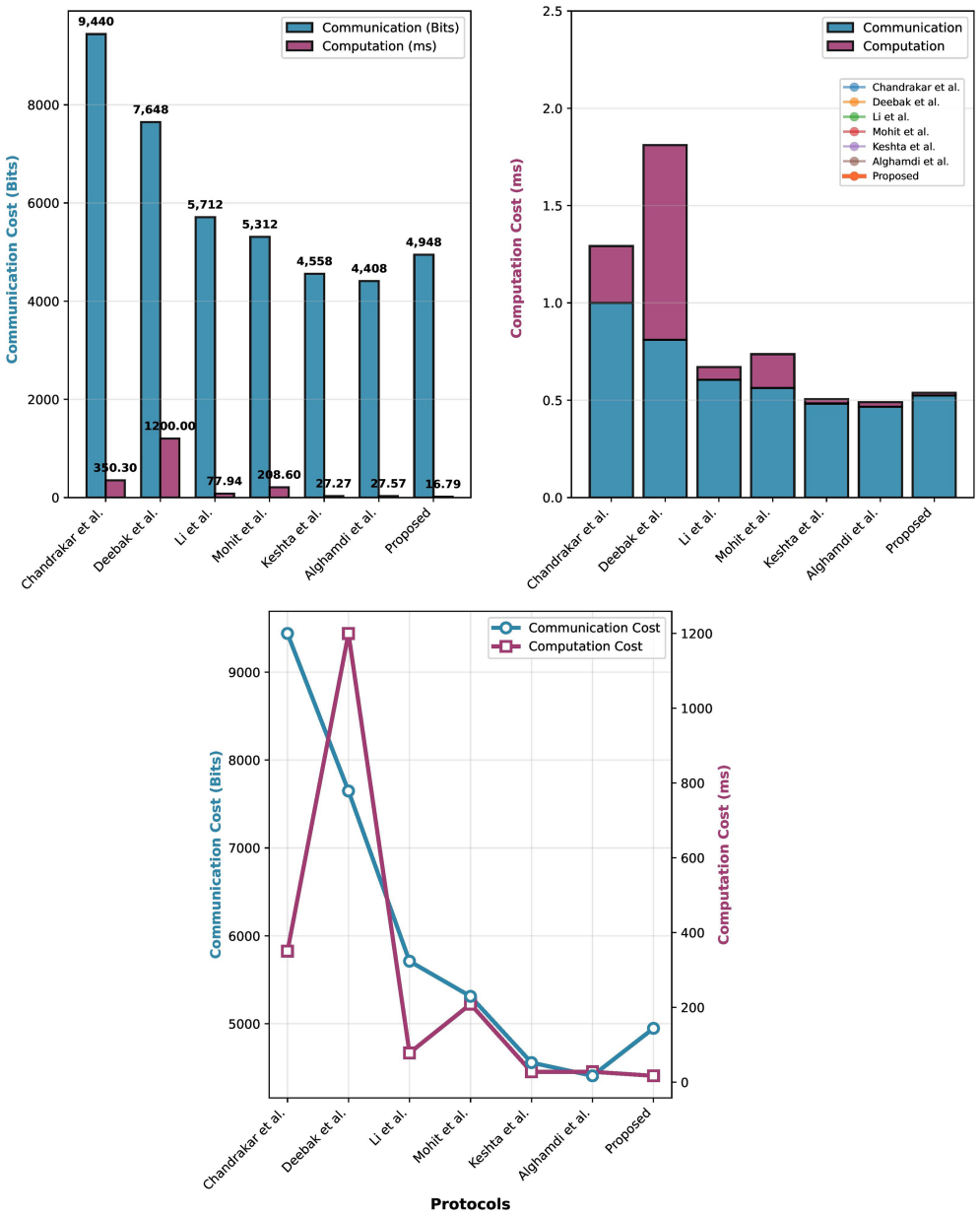

Figure 7: Performance metrics comparison with state-of-the-art schemes.

Fig. 7 demonstrates the differences in computation and communication costs among various protocols. For example, [51] leads the pack at 4408 bits, closely followed by [50] of communication cost of 4558 bits, while the proposed approach has 4948 bits. Also, [11] incurs the highest overhead at 9440 bits, over twice as much as the protocol scheme. Similarly, the proposed technique, with a mere 16.79 ms, significantly reduces computing costs compared to the worst-performing protocols, such as [12] at 1200 ms and [11] at 350.3 ms, surpassing its closest rivals [26,27], which are both up to 27 ms by improving the proposed protocol up to 38%. It means the proposed technique emerges as the superior overall solution due to its remarkable computing efficiency, the crucial criterion for real-time performance, despite its slightly higher communication cost (12%) than [51]. The formula used for reduction in the proposed scheme is shown in Eq. (64):

$$Reduction\ (\%\,Improvement) = \frac{Baseline\ Cost - Proposed\ Cost}{Baseline\ Cost} \times 100 \tag{64}$$

- Best Communication Cost: [51] (4408 bits) and Best Computation Cost: Proposed (16.79 ms)
- Communication cost difference: +540 bits (12.3% higher)
- Computation cost improvement: 0.0% better
- Combined Performance Score (The Proposed one is better):
- Overall [11] is 0.646, [12] is 0.905, [26] is 0.335, [27] is 0.368, [50] is 0.253, [51] is 0.245, and the proposed is 0.269, which achieves the best balance between communication and computation costs!

(2) **Comparative Analysis (Security Functionalities):** The proposed protocol is compared with the state-of-the-art schemes in security functionalities, including at least five attacks and five security features. The result demonstrated that the proposed protocol not only meets but exceeds expectations, as it is safe against all the attacks and avails all the mentioned features, as depicted in Table 9, which ✓ means the mentioned scheme is resisting the mentioned attack and availing the mentioned security feature, χ which means contrary to ✓. The security features of several protocols [11,12,21,26,27,38] are contrasted in Table 9 with widespread cryptographic attacks and security attributes. All of the threats listed—including Man-in-the-Middle, Password Guessing, Impersonation, Insider, Key Disclosure, and Replay attacks—are successfully mitigated by the proposed protocol, which also preserves all of the security features—including anonymity, untraceability, forward/backward secrecy, and session key secrecy. Notably, [11,21,27,51,52] demonstrate susceptibility to Man-in-the-Middle assaults, but only the Proposed procedure and [38] provide defense against impersonation attempts. Except [26,27,38,51], most protocols are resistant to password guessing attacks. Forward/Backward Secrecy is absent in [11,27,50,53]. As the existing protocols have serious flaws, especially [11,21], which do not offer anonymity, or untraceability, and [53], which are susceptible to a key disclosure attack, the proposed protocol stands out as the sole alternative offering both untraceability and total attack resistance.

(3) **Energy Consumption through Different Infrastructure:** If we compare the energy efficiency of four different computational strategies, including local, edge, cloud offloading, and the proposed hybrid approach, the result depicted in Fig. 8 demonstrates that as the number of offloaded tasks increases, the energy consumption rises for all strategies, but at different rates. However, the proposed scheme maintains the lowest energy consumption across the entire range of tasks, which is one of the most energy-efficient solutions.

Table 9: Comparative analysis (security functionalities).

Protocols → Functionalities↓	[11]	[12]	[21]	[26]	[27]	[38]	[52]	[50]	[53]	[51]	Proposed
Man-in-the-Middle Attack	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓
Password Guessing Attack	✗	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓
Impersonation Attack	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✓
Insider Attack	✓	✓	✗	✓	✗	✓	✓	✓	✗	✓	✓
Key Disclosure Attack	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓
Replay Attack	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓
Anonymity	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓
Untraceability	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓
Forward Backward Secrecy	✗	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓
Session Key Secrecy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

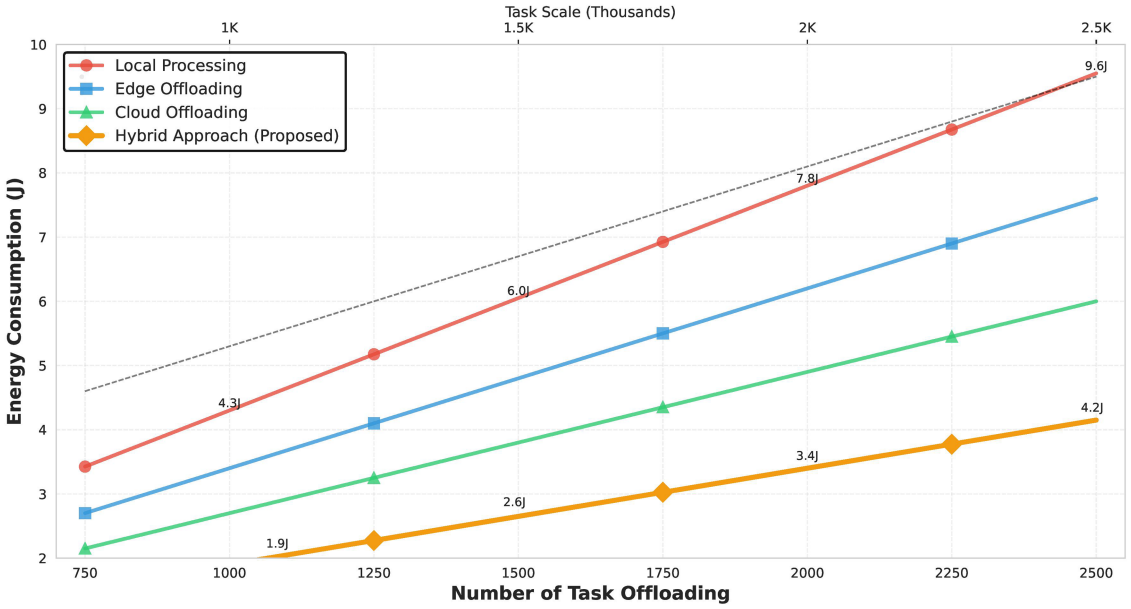

Figure 8: Energy consumption vs. number of task offloaded into differing strategies.

7 Conclusion

This article presented an authentication protocol for the cloud-centric and IoT-driven e-healthcare communication system based on a simple hash cryptographic function and XOR operation. The security of the proposed authentication scheme has been scrutinized via a widely used BAN logic, ProVerif simulation, and pragmatic discussion. In contrast, the performance metrics have been measured by considering computation cost, storage overhead, communication cost, energy consumption, and other trade-offs, such as energy consumption vs. the number of devices, energy consumption vs. task offloading, and runtime vs. throughput. The comparative analysis demonstrated that the proposed protocol outperforms all its competitors and is recommended for practical implementation in the real-world e-healthcare communication system. Future research will extend the proposed work by integrating blockchain technology, identity distribution, AI-enhanced, and quantum key derivation.

Acknowledgement: The authors are thankful to the Deanship of Graduate Studies and Scientific Research at the University of Bisha for supporting this work through the Fast-Track Research Support Program.

Funding Statement: The authors received no specific funding.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Saeed Ullah Jan, Fahad Algarni; data collection: Fahad Algarni; analysis and interpretation of results: Fahad Algarni, Saeed Ullah Jan; draft manuscript preparation: Saeed Ullah Jan, Fahad Algarni. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data will be available on request.

Ethics Approval: Not applicable to this research work.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Algorithm implementation code

```
import hashlib
import time
import random
from django.http import JsonResponse
from django.utils.crypto import get_random_string
def h(data):
“““ Hashing function h() “““
return hashlib.sha256(data.encode()).hexdigest()
def generate_timestamp():
“““ Generates a timestamp (T) “““
return int(time.time())
def generate_random_number():
“““ Generates a random number N “““
return random.randint(1000, 9999)
def xor(a, b):
“““ XOR operation on two strings of equal length “““
return ‘’.join(chr(ord(x) ^ ord(y)) for x, y in zip(a, b))
def protocol_view(request):
IDP = request.GET.get(‘IDP’, ‘client’) # Client’s ID
IDPM = request.GET.get(‘IDPM’, ‘server’) # Server’s ID
N = generate_random_number()
h_func = h # Hash function
SK = h(h_func(‘initial’) + str(N))
RP = h(SK + str(N)) # Response key RP
HPIDP = h(IDP) # Hash of IDP
HPIDPM = h(IDPM) # Hash of IDPM
KP = get_random_string(16) # Generate random key for KP
T = generate_timestamp() # Generate timestamp T
μ = ‘mu_value’ # constant or dynamic value
s = ‘some_value’ # This could also be dynamic
XPM = str(N) + μ + str(T) # Combining values to form XPM
```

```
T1 = generate_timestamp()
response_1 = {
'KP': KP,
'HPIDPM': HPIDPM,
'XPM': XPM,
'T1': T1
}
Tc = generate_timestamp() # Current timestamp
if T1 − Tc <= 10: # ΔT = 10 s (example)
KP_new = xor(XPM, h_func(HPIDPM + KP))
YP = XPM + xor(XPM, str(N)) + KP_new
T2 = generate_timestamp()
response_2 = {
'HPIDP': HPIDP,
'KP': KP_new,
'YP': YP,
'T2': T2
}
if T2 − Tc <= 10:
KP_new2 = xor(XPM, h_func(HPIDPM + KP))
if KP_new2 == KP_new:
YP_new = XPM + xor(XPM, str(N)) + KP_new2
if YP_new == YP:
SK = h(SK + XPM) # Final SK update
RP = h(SK + XPM) # Final RP update
T3 = generate_timestamp()
response_3 = {
'KP*': KP_new2,
'YP*': YP_new,
'RP': RP,
'T3': T3
}
if T3 − Tc <= 10:
return JsonResponse({'status': 'Success', 'SK': SK})
```

Appendix B

ProVerif implementation Code

```
free c: channel.
free dS: bitstring [private].
free r1: bitstring [private].
free IDW: bitstring [private].
free IDPM: bitstring [private].
```

```
free SK: bitstring [private].
fun h(bitstring): bitstring.
fun xor(bitstring, bitstring): bitstring.
fun concat(bitstring, bitstring): bitstring.
event begin_PM(bitstring).
event begin_CS(bitstring).
event end_PM(bitstring).
event end_CS(bitstring).
query attacker(SK).
query x:bitstring;
event(end_CS(x)) ==> event(begin_PM(x)).
query x:bitstring;
event(end_PM(x)) ==> event(begin_CS(x)).
let PM(Q1:bitstring, Q2:bitstring, PIDPM:bitstring) =
new r2:bitstring;
new TPM:bitstring;
let S1 = xor(h(concat(Q1, r2)), IDPM) in
let S2 = xor(Q2, r2) in
let S3 = h(concat(PIDPM,
concat(IDPM,
concat(r2, TPM)))) in
event begin_PM(PIDPM);
out(c, (PIDPM, TPM, S1, S2, S3));
in(c, (W3:bitstring, W4:bitstring, W5:bitstring, W6:bitstring, TCS:bitstring));
let SK = xor(W3, h(concat(IDPM, concat(r2, TCS)))) in
let PIDPM2 = xor(W4, h(concat(IDPM, concat(r2, TCS)))) in
let Q1new = xor(W5, h(concat(IDPM, concat(r2, TCS)))) in
let W6p =
h(concat(IDPM,
concat(PIDPM2,
concat(r2,
concat(SK,
h(concat(dS, concat(PIDPM2, TCS)))))))) in
if W6p = W6 then
event end_PM(PIDPM2)
else
0.
let W() =
in(c, (PIDPM:bitstring, TPM:bitstring, S1:bitstring, S2:bitstring, S3:bitstring));
new r3:bitstring;
new TW:bitstring;
let U1 = xor(h(concat(r1, TW)), r3) in
let U2 =
h(concat(PIDPM,
```

```
concat(IDW,
concat(S1,
concat(S2,
concat(TPM,
concat(TW, r3))))))) in
out(c, (PIDPM, IDW, S1, S2, S3, U1, U2, TPM, TW));
in(c, (W1:bitstring, W2:bitstring, W3:bitstring,
W4:bitstring, W5:bitstring, W6:bitstring, TCS:bitstring));
let SK = xor(W1, h(concat(r3, concat(r1, TCS)))) in
let W2p = h(concat(IDW, concat(r3, concat(TCS, SK)))) in
if W2p = W2 then
out(c, (W3, W4, W5, W6, TCS))
else
0.
let CS(Q1:bitstring, Q2:bitstring) =
in(c, (PIDPM:bitstring, IDW:bitstring,
S1:bitstring, S2:bitstring, S3:bitstring,
U1:bitstring, U2:bitstring,
TPM:bitstring, TW:bitstring));
let r3 = xor(U1, h(concat(r1, TW))) in
let U2p =
h(concat(PIDPM,
concat(IDW,
concat(S1,
concat(S2,
concat(TPM,
concat(TW, r3))))))) in
if U2p = U2 then
let r2 = xor(S2, Q2) in
let IDPMr = xor(S1, h(concat(Q1, r2))) in
let S3p = h(concat(PIDPM,
concat(IDPMr,
concat(r2, TPM)))) in
if S3p = S3 then
new PIDPM2:bitstring;
new TCS:bitstring;
let SK =
h(concat(IDPMr,
concat(IDW,
concat(r2,
concat(r3,
concat(TPM, TW)))))) in
let W1 = xor(h(concat(r3, concat(r1, TCS))), SK) in
let W2 = h(concat(IDW, concat(r3, concat(TCS, SK)))) in
```

```
let W3 = xor(h(concat(IDPMr, concat(r2, TCS))), SK) in
let W4 = xor(h(concat(IDPMr, concat(r2, TCS))), PIDPM2) in
let W5 = xor(h(concat(IDPMr, concat(r2, TCS))), h(concat(PIDPM2, dS))) in
let W6 =
h(concat(IDPMr,
concat(PIDPM2,
concat(r2,
concat(SK,
h(concat(dS, concat(PIDPM2, TCS)))))))) in
event begin_CS(PIDPM2);
out(c, (W1, W2, W3, W4, W5, W6, TCS));
event end_CS(PIDPM2)
else
0
else
0.
process
new PIDPM:bitstring;
let Q1 = h(concat(PIDPM, dS)) in
let Q2 = h(concat(IDPM, dS)) in
(PM(Q1, Q2, PIDPM)
| W()
| CS(Q1, Q2)
)
```

References

1. Dzung D, Naedele M, Von Hoff TP, Crevatin M. Security for industrial communication systems. Proc IEEE. 2005;93(6):1152–77. doi:10.1109/jproc.2005.849714.
2. Chen L, Gong G. Communication system security. Boca Raton, FL, USA: CRC Press; 2012.
3. Rahman MG, Imai H. Security in wireless communication. Wirel Pers Commun. 2002;22(2):213–28.
4. Burg A, Chattopadhyay A, Lam KY. Wireless communication and security issues for cyber–physical systems and the Internet-of-Things. Proc IEEE. 2017;106(1):38–60. doi:10.1109/jproc.2017.2780172.
5. Stavroulakis P, Stamp M. Handbook of information and communication security. Berlin/Heidelberg, Germany: Springer; 2010.
6. Roy KS, Deb S, Kalita HK. A novel hybrid authentication protocol utilizing lattice-based cryptography for IoT devices in fog networks. Digit Commun Netw. 2024;10(3):989–1000. doi:10.1016/j.dcan.2022.12.003.
7. Liu W, Park EK. E-healthcare security solution framework. In: Proceedings of the 30th International Conference on Computer Communications and Networks; 2012 Jul 19–22; Athens, Greece. p. 1–6.
8. Zeadally S, Isaac JT, Baig Z. Security attacks and solutions in electronic health (E-health) systems. J Med Syst. 2016;40(12):263. doi:10.1007/s10916-016-0597-z.
9. Masud M, Gaba GS, Choudhary K, Alroobaea R, Hossain MS. A robust and lightweight secure access scheme for cloud based E-healthcare services. Peer-to-Peer Netw Appl. 2021;14(5):3043–57.
10. Padmaja K, Seshadri R. A real-time secure medical device authentication for personal E-Healthcare services on cloud computing. Int J Syst Assur Eng Manag. 2022;13(Suppl 1):S186–96. doi:10.1007/s13198-021-01148-1.

11. Chandrakar P, Sinha S, Ali R. Cloud-based authenticated protocol for healthcare monitoring system. J Ambient Intell Humaniz Comput. 2020;11(8):3431–47.
12. Deebak BD, Al-Turjman F. Smart mutual authentication protocol for cloud based medical healthcare systems using internet of medical things. IEEE J Sel Areas Commun. 2021;39(2):346–60. doi:10.1109/jsac.2020.3020599.
13. Chiou SY, Ying Z, Liu J. Improvement of a privacy authentication scheme based on cloud for medical environment. J Med Syst. 2016;40(4):101. doi:10.1007/s10916-016-0453-1.
14. Qadir GA, Hussan BK. An authentication and access control model for healthcare based cloud services. J Eng. 2023;29(9):15–26. doi:10.31026/j.eng.2023.03.02.
15. Okikiola FM, Mustapha AM, Akinsola AF, Sokunbi MA. A new framework for detecting insider attacks in cloud based e-health care system. In: Proceedings of the 2020 International Conference in Mathematics, Computer. Engineering and Computer Science (ICMCECS); 2020 Mar 18–21; Lagos, Nigeria. p. 1–6.
16. Benil T, Jasper J. Cloud based security on outsourcing using blockchain in E-health systems. Comput Netw. 2020;178(3):107344. doi:10.1016/j.comnet.2020.107344.
17. Jan SU, Ali S, Abbasi IA, Mosleh MA, Alsanad A, Khattak H. Secure patient authentication framework in the healthcare system using wireless medical sensor networks. J Healthc Eng. 2021;2021(4):9954089. doi:10.1155/2021/9954089.
18. Jan SU, Tariq MU, Khashan OA, Alzahrani N, Ghani A. FEELAP: fuzzy extractor-based efficient lightweight authentication protocol for edge-IoT ecosystem in e-healthcare. IEEE Open J Comp Soc. 2025;6:1727–40. doi:10.1109/OJCS.2025.3616014.
19. Ahmim M, Ouafi N, Ullah I, Ahmim A, Chefrour D, Almukhlifi R. LSAP-IoHT: lightweight secure authentication protocol for the internet of healthcare things. Comput Mater Contin. 2025;85(3):5093–116. doi:10.32604/cmc.2025.067641.
20. Anandhi T, Sangari AS. Privacy preserving authentication protocol with optimized Elliptic Curve Cryptography for healthcare domain in cloud. Cluster Comput. 2026 Apr;29(2):77. doi:10.1007/s10586-025-05854-4.
21. Tanveer M, Chelloug SA, Alabdulhafith M, El-Latif AAA. Lightweight authentication protocol for connected medical IoT through privacy-preserving access. Egypt Inform J. 2024;26(7):100474. doi:10.1016/j.eij.2024.100474.
22. Mir O, Nikooghadam M. A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. Wirel Pers Commun. 2015;83(4):2439–61. doi:10.1007/s11277-015-2538-4.
23. Ni S, Kang B, Li A, Huo Y, Zuo X. Analysis and improvement of a privacy-preserving authentication scheme for telecare medical information system environment. Wuhan Univ J Nat Sci. 2023;28(6):531–40. doi:10.1051/wujns/2023286531.
24. Yu S, Park K. Sals-tmis: secure, anonymous, and lightweight privacy-preserving scheme for iomt-enabled TMIS environments. IEEE Access. 2022;10:60534–49. doi:10.1109/ACCESS.2022.3181182.
25. Zheng L, Zhang Y, Zhang R, Chen J, Cui M, Song C. An improved authentication protocol in telemedicine system. Lect Notes Comput Sci. 2018;11337(4):177–84. doi:10.1007/978-3-030-05234-8_22.
26. Li CT, Shih D, Wang C. Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems. Comput Methods Programs Biomed. 2018;157:191–203. doi:10.1016/j.cmpb.2018.02.002.
27. Mohit P, Amin R, Karati A, Biswas GP, Khan MK. A standard mutual authentication protocol for cloud computing based health care system. J Med Syst. 2017;41(5):83. doi:10.1007/s10916-017-0699-2.
28. Amin R, Islam SK, Gope P, Choo KR, Tapas N. Anonymity preserving and lightweight multimedical server authentication protocol for telecare medical information system. IEEE J Biomed Health Inform. 2019;23(4):1749–59. doi:10.1109/jbhi.2018.2870319.
29. Setianto, Dwi YB, Wahyuningrum, Estri S. Multitier model with JSON-RPC in telemedicine devices authentication and authorization protocol. In: Proceedings of the 2021 7th International Conference on Engineering, Applied Sciences and Technology (ICEAST); 2021 May 1–3; Pattaya, Thailand. p. 213–6.
30. Son S, Lee J, Kim M, Yu S, Das AK, Park Y. Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain. IEEE Access. 2020;8:192177–91. doi:10.1109/ACCESS.2020.3032680.

31. Lei C, Chuang Y. Privacy protection for telecare medicine information systems with multiple servers using a biometric-based authenticated key agreement scheme. IEEE Access. 2019;7:186480–90. doi:10.1109/access.2019.2958830.
32. Hamed NM, Yassin AA. Secure patient authentication scheme in the healthcare system using symmetric encryption. Iraqi J Electr Electron Eng. 2022;18(2):94–105. doi:10.37917/ijeee.18.1.9.
33. Yao H, Yan Q, Fu X, Zhang Z, Lan C. ECC-based lightweight authentication and access control scheme for IoT E-healthcare. Soft Comput. 2022;26(10):4441–61. doi:10.21203/rs.3.rs-210016/v1.
34. Lee TF, Lin KW, Hsieh YP, Lee KC. Lightweight cloud computing-based RFID authentication protocols using PUF for e-healthcare systems. IEEE Sens J. 2023;23(6):6338–49. doi:10.1109/jsen.2023.3242132.
35. Zhang L, Zhu Y, Ren W, Zhang Y, Choo KR. Privacy-preserving fast three-factor authentication and key agreement for IoT-based E-health systems. IEEE Trans Serv Comput. 2023;16(2):1324–33. doi:10.1109/TSC.2022.3149940.
36. Sun L, Liu D, Li Y, Zhou D. A blockchain-based e-healthcare system with provenance awareness. IEEE Access. 2024;12(3):39532–44. doi:10.1109/access.2024.3440170.
37. Sunitha MJ, Asendra C, Kumar BB, Harshith Goud E, Basha SH. User authentication scheme and identity management for e-health systems using blockchain technology. In: Proceedings of the 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS); 2024 Apr 18–19; Chikkaballapur, India. p. 1–7.
38. Alzahrani A. RLKS-TMS: a robust and lightweight key agreement scheme for telemedicine system. IEEE Access. 2024;12:24312–27. doi:10.1109/ACCESS.2024.3422038.
39. Dolev D, Yao A. On the security of public key protocols. IEEE Trans Inf Theory. 1983;29(2):198–208. doi:10.1109/tit.1983.1056650.
40. Burrows M, Abadi M, Needham R. A logic of authentication. ACM Trans Comput Syst. 1990;8(1):18–36. doi:10.1145/77648.77649.
41. Canetti R, Goldreich O, Halevi S. The random oracle methodology, revisited. J ACM. 2004;51:557–94. doi:10.1145/1008731.1008734.
42. Blanchet B, Smyth B, Cheval V, Sylvestre M. ProVerif 2.00: automatic cryptographic protocol verifier, user manual and tutorial. 2018 [cited 2026 Jan 1]. Available from: https://bblanche.gitlabpages.inria.fr/proverif/manual.pdf.
43. Suzuki K, Tonien D, Kurosawa K, Toyota K. Birthday paradox for multi-collisions. In: International Conference on Information Security and Cryptology. Berlin/Heidelberg, Germany: Springer; 2006. p. 29–40.
44. Upton E, Halfacree G. Raspberry Pi user guide. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2016.
45. Rahaman S, Cai H, Chowdhury O, Yao D. From theory to code: identifying logical flaws in cryptographic implementations in C/C++. IEEE Trans Dependable Secur Comput. 2021;19(6):3790–803. doi:10.1109/TDSC.2021.3108031.
46. Kilinc HH, Yanik T. A survey of SIP authentication and key agreement schemes. IEEE Commun Surv Tutorials. 2014;16(2):1005–23. doi:10.1109/surv.2013.091513.00050.
47. Patil P, Narayankar P, Narayan DG, Meena SM. A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish. Procedia Comput Sci. 2016;78:617–24. doi:10.1016/j.procs.2016.02.108.
48. Althebyan Q, Yaseen Q, Jararweh Y, Al-Ayyoub M. Cloud support for large scale e-healthcare systems. Ann Telecommun. 2016;71(9–10):503–15. doi:10.1007/s12243-016-0496-9.
49. Prelov VV. Communication channel capacity with almost Gaussian noise. Theory Probab Appl. 1989;33(3):405–22. doi:10.1137/1133068.
50. Keshta I. A CRC-based authentication model and ECC-based authentication protocol for resource-constrained IoT applications. IEEE Access. 2024;12(1):24328–44. doi:10.1109/access.2024.3482991.
51. Alghamdi AM. Design and analysis of lightweight and robust authentication protocol for securing the resource constrained IIoT environment. PLoS One. 2025;20(2):e0318064. doi:10.1371/journal.pone.0318064.
52. Alzahrani A. Developing a provable secure and cloud-centric authentication protocol for the e-healthcare system. IEEE Access. 2024;12(2):39545–61. doi:10.1109/access.2024.3500216.
53. Alzahrani A, Alzahrani HA. A privacy-preserving and energy efficient authentication protocol for the cloud-based e-healthcare system. Alex Eng J. 2025;118(6):59–90. doi:10.1016/j.aej.2025.01.051.