Hindawi
Security and Communication Networks
Volume 2022, Article ID 5300253, 14 pages
https://doi.org/10.1155/2022/5300253

Research Article

A Secure IoT and Cloud Computing-Enabled e-Health Management System

Chanapha Butpheng,[1] **Kuo-Hui Yeh**,[1,2] **and Jia-Li Hou**[1]

[1]*Department of Information Management, National Dong Hwa University, Hualien 97401, Taiwan*
[2]*Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan*

Correspondence should be addressed to Kuo-Hui Yeh; khyeh@gms.ndhu.edu.tw

Received 3 November 2021; Revised 19 April 2022; Accepted 11 May 2022; Published 2 June 2022

Academic Editor: Azees M

Utilization of the Internet of Things (IoT) technology is virtually ubiquitous across various life disciplines. When the IoT is integrated into e-health system to enable more real-time on-demand services, it brings significant convenience to the physicians and patients. However, the risk of encountering unreliable information and potential security threat is promptly raised at the same time. Therefore, in this research, we establish a secure IoT and cloud computing-enabled e-health management system in which a distinctive authentication scheme is adopted. The proposed system ensures all major security requirements, such as confidentiality, entity anonymity, repudiation tracking, data integrity, and innate resistance to man-in-the-middle, location spoofing, and replay attacks. Moreover, our system can effectively be executed on low-powered processing units and simultaneously guarantees greater security than historically available alternatives.

1. Introduction

The adoption of smart object technology has rapidly driven a huge advancement in wireless sensor-based application development. These applications deliver efficiency of data transfer, and their nature of ubiquity enables real-time and on-demand Internet of Things- (IoT-) based services into daily life. The most important variable to consider when developing novel IoT-based applications is security during data processing. Potential threats constantly emerge as technology evolves. Hello Barbie is an example of a consumer product with IoT functionality that could be utilized as a possible threat vector by bad actors to spy on consumers utilizing the built-in camera and voice-interaction capabilities. In addition, hackers might be able to use network-monitoring-driven devices to collect personal sensitive information from network attached IoT devices.

IoT continued to emerge as one of the fastest accumulations of communication standards by which smart objects used in our daily lives are integrated with the Internet. The onboard computing and communication capabilities are the drivers of this trend. However, all new technology has potential security vulnerabilities that emerge as the technology matures. Every IoT object may be a potential threat vector in the sense that each IoT device represents a possible vulnerable entry point that can be exploited by malicious actors. There are two security aspects we may consider: (a) the assurance of robust physical protection of IoT devices; (b) the assurance of data availability, confidentiality, integrity, and privacy while engaging in normal operation. Given the advanced nature and rapid innovation of IoT technologies, there is a great demand and expectation that revolutionary security solutions will be adapted specifically to IoT-based objects and their specific applications. The demand stems from the shortcomings of traditional security protocols which are not designed for IoT devices. Firewalls that integrate protocols to manage network traffic at the application level are an example of a solution that is capable of regulating high-level Internet traffic. Nevertheless, firewalls are not a panacea in regard to end-point IoT devices. Moreover, IoT objects are frequently mission-specific and possess limited resources to accomplish their respective

missions. Hence, traditional security solutions must be restructured for the specific security requirements of IoT devices to secure IoT-based applications and systems as a whole. For instance, Yousefnezhad et al. [1] defined fundamental security elements: data availability, confidentiality, and integrity. Moreover, they outlined a lifecycle security solution comprised of three stages: beginning, middle, and end of life stages. Thus, the required privacy and security elements have been classified utilizing the concept of IoT network architecture. More concretely, the authors developed a security architecture which categorizes the primary security requirements for the fundamental layer, the perceptual layer, the network layer, the support layer, and the application layer. The security requirements for each layer are agreement of encryption/key, verification of identification, protected cloud computing, robust antivirus abilities, authentication verification, and data privacy assurance. Furthermore, Hathaliya and Tanwar [2] presented the importance of maintaining privacy/security within a e-health system and an extensive summary of privacy/security solutions. In regard to data communication, with the popularity of IoT applications, the Internet is utilized to communicate a huge amount of generated data. This method of communication is vulnerable to attack from bad actors. An IoT device may be most vulnerable when communicating, storing, or processing data. Gardašević et al. [3] identified security and privacy threats/attacks for IoT smart healthcare systems, such as jamming, eavesdropping, spoofing, man-in-the-middle attacks, and so on. In order to secure the system against these attacks, a solution has been developed to support the advancement and widespread utilization of distributed computing.

Typically, security is implemented and aligned with the target system's operation process, including device authentication, firewall integration, secure booting, efficient network management, access control, and updates/patches. Secure interdevice communication/authentication offers significant security benefit for IoT crucial elements. On the other hand, the academic community has exhaustively dedicated time and resources to utilize cloud computing to solve IoT efficiency and scalability issues. The cloud computing model is an extremely powerful tool because of its ability to be quickly implemented and scaled through the use of virtualized resources management. The utilization of IoT devices and its amalgamation with cloud computing provides network benefits, including communication efficiency, quick access, and intelligent interconnection capabilities, and provides complementary capabilities when integrated as a component in an efficient system that is scalable and flexible. Therefore, our study proposes a secure e-health management system operated with IoT and cloud computing. We begin by defining fundamental required security elements for an e-health system that utilizes IoT-cloud resources. Then, we will introduce a robust authentication mechanism that operates within Body Sensor Networks (BSN) technology to ensure that the fundamental security elements, including attack resistance and data integrity/confidentiality, are satisfied.

Our work is structured accordingly: section 2 summarizes security as it is today and outlines fundamental security elements necessary for e-health systems that utilize IoT-cloud technology. Then, section 3 introduces our proposed system with an authentication protocol, and section 4 examines the robustness of our proposed system. Next, section 5 evaluates the effectiveness of the proposed system through a proof-of-concept implementation. Lastly, the conclusion and suggestions for future research can be found in section 6.

2. Related Works

This section summarizes security as it is today and outlines fundamental security elements necessary for e-health systems that utilize IoT-cloud technology.

2.1. The Current State of the Art. The primary security priorities for e-health systems that utilize IoT-cloud technology are robust preservation of security/privacy in regard to patient health information. Hence, many systems, integrating privacy policies and multilevel security mechanisms, have been proposed to be more capable of managing patient health information. Recently, Butpheng et al. [4] presented a practical amalgamation of cloud computing and IoT-technologies in an e-health setting. The authors demonstrated that integration/implementation of cloud computing and IoT-technologies (CIoT) is achievable in an efficient manner. This integration improves health diagnostics, treatment timeliness, and health outcomes and reduces costs. CIoT is comprised of interconnected smart hardware, custom systems, and custom applications that communicate over the Internet. The authors provided a comprehensive technical architecture with an inclusive list of privacy/security requirements for e-health systems that utilize IoT-cloud technology including identification, authentication, and authorization. Aligning with the proposed architecture, security solutions can be effectively designed to decrease the probability of an attack via examining and identifying vulnerabilities from each security aspect.

Similarly, Wang and Cai [5] proposed a secure communication protocol for incorporating healthcare data and Named Data Networking (NDN) based on IoT technology and Edge-cloud computing (SHNIE) to secure and expedite the delivery of medical data and reduce latency and cost. SHNIE can efficiently deliver aggregated NDN-medical data to multiple users from the nearest edge devices that own the required medical data. The security of communications is realized through a hash ciphertext of the provider's name and IDs as a verification token in the NDN to preserve privacy and security. Moreover, Ray et al. [6] designed a novel solution to deal with issues of interoperability, heterogeneity, and Internet-aware resistances. Proposed devices will collect patient data instantaneously and then process and analyze it with embedded Internet connected electronics. The proposed devices utilize the Internet for all interactions and communication and are capable of remote monitoring and controlling. Then, the authors further propose an e-healthcare architecture that incorporates blockchain technology based on IoT that is classified as follows: IoT e-healthcare; blockchain platform; connectivity;

and IoT devices. Furthermore, Shewale and Sankpal [7] utilized various medical technologies such as real-time monitoring, patient data administration, and healthcare supervision via BSN technology to observe patients' heartbeats and blood pressure. They introduced a monitoring system to detect the status of each patient's health data, collect it, then forward the data to the server using Wi-Fi module based wireless communication. In 2019, Chenthara et al. [8] identified essential privacy/security elements for e-health systems that utilize cloud computing. They engineered a strengthened security and privacy system to manage numerous types of vulnerable/private e-health data including user/patient data, diagnostic data, and treatment data. Next, Koutli et al. [9] outlined a VICINITY security framework that can be integrated into e-health applications that utilize IoT devices to service elderly patients. In the scheme, they demonstrated how Ambient-Assisted-Living (AAL) and mHealth (VICINITY IoT platform) can be used in conjunction to satisfy all required security/privacy elements. Their e-health applications are allowed for the secure processing of personal medical data and remote data storage.

In 2018, Chattopadhyay et al. [10] coined the term Internet of Medical Things (IoMT) since IoT object adoption in healthcare has become ubiquitous. IoMT provides an unparalleled level of patient/doctor accessibility through the utilization of instantaneous management of patient data, treatment support, data processing, and other services. The authors utilized a cryptosystem to safeguard communication and authentication between local processing units (LPU), smart sensors, and transmission gateways. The proposed cryptosystem is integrated with a BSN comprising mainly of wearable and pervasive IoT devices as a healthcare monitoring system. Alihamidi et al. [11] proposed a Blockchain architecture enabling fog computing technology to secure e-health systems. They designed a three-layered IoT-based solution for e-Health systems with a Wireless Sensor Network (WSN). These three major classes are differentiated by their programmed level of access and connectivity between the connected system units. Various vectors of access can be used for evaluating the security effectiveness of any IoT architecture. Rahmani et al. [12] applied fog-computing to e-health systems that utilize IoT-cloud technology by introducing a geographically distributed intermediate intelligence layer in between the cloud devices and the sensor devices. Their solution addresses problems with agility, expandability, and dependability. The prototype, called a smart e-health gateway health monitoring system, was found effective at enhancing system intelligence, energy efficiency, mobility, performance, interoperability of security protocols, and reliability.

2.2. Security Requirement for IOT-Cloud-Based E-Health Systems. In section B, we introduce required security elements for e-health systems that utilize IoT-cloud technology management frameworks. The following are the major security requirements.

2.2.1. Access Control: Required for Secure Communication. Researchers have thoroughly examined dynamic identity-based authentication protocols and identified their advantages, such as user convenience and protocol efficiency, for more than 20 years. In addition, secure communication requires anonymous and unpredictable identities and robust session keys must be predetermined to ensure secure communication between objects. Moreover, simple authentication and login without the use of session keys is easily vulnerable to security threats. Regardless of claims that various security schemes such as Secure Sockets Layer (SSL) and Transport Layer Security (TLS) can be utilized to ensure that security remains strong after authentication, they are ultimately inefficient due to exorbitant computational cost. Therefore, a solution with session key agreement scheme is essential to guarantee robust and secure communication. For the purpose of this study, our team attempts to demonstrate a protected communication protocol for e-health systems that utilize IoT-cloud technology for IoT-cloud-based e-health systems. However, e-health systems that utilize IoT-cloud technology always rely on various technologies such as sensors, actuators, and communication modules for data acquisition, manipulation, transmission, real-time data analysis, data storage, and application functions. Hence, it is essential to develop robust protocols being perfectly capable of operational conditions for IoT medical devices and achieving the required security as well.

2.2.2. Data Encryption. As suggested by Yeh [13] and Yeh [14], the one-way hash operations maintain qualified security in an effective manner, while the exclusive-or protocol may be vulnerable to attackers. As exclusive-or operation is capable of resisting ciphertext only attacks, these protocols are the minimum acceptable level of cryptanalytic security. As a result, the integration of exclusive-or operation must be given careful consideration when designing security protocols. For example, publicly transmitted text must utilize volatile cipher formulae in conjunction with exclusive-or procedures. Moreover, these two constructs must not simply and directly be applied to the cipher. In addition, Ding et al. [15] determined that in order to protect network communication, data encryption must be utilized. Data encryption technology can deliver security within a computer network and its communication. Data encryption is utilized for the transformation of data based on predetermined bespoke rules/algorithms. This enables secure data transmission and the data can be decrypted at the destination through the use of unique unlocking methods. Then, Al-Haija et al. [16] suggested the use of flow cryptography to secure information between communicated parties through the use of encryption to transform plaintext into ciphertext. Subsequently, a designated key can be utilized to decipher the ciphertext into plaintext. The following is a list of standard data encryption algorithms utilized to secure network traffic: Data Encryption Standard (DES) algorithm, MD5 algorithm, and Rivest–Shamir–Adelman (RSA) algorithm [15, 16].

2.2.3. Resistance against Spoofing Attacks. IoT-cloud-based communication architectures are primarily built atop a variant of traditional wireless sensor networks, called body sensor networks (BSN), consisting of many different body biosensors. The primary problem to address is the security of these individual biosensors considering that

they collect and retain private health-related biodata. Hence, our proposed secure e-health system that utilizes IoT-cloud technology should carefully consider several different security requirements such as user identification, secure network switching, data anonymity/untraceability, and spoofing attack resistance at both the device and system level. These required security elements could be addressed via an anonymous authentication protocol. Stepien et al. [17] claimed that the most efficient method to secure IoT-based systems is to embed security solutions into the IoT devices directly. The best practice is that the IoT devices should be capable of accommodating and maintaining authenticity. That is, IoT-based systems must be capable of preventing unauthorized access while allowing flexible interoperability between connected devices under ad hoc network conditions. Moreover, to safeguard the data security at the IoT device end, it is imperative that the devices should be manufactured with the ability to en/decryption functions. This will ensure that malicious entities cannot physically retrieve any raw data from IoT device directly. There are multitudes of security, privacy, and safety risks associated with IoT-device utilization in e-health systems. One example is that personal data can be stolen and used to harm users or patients in different ways. Another example is that bad actors can initiate an insider/outsider attack to acquire patient registration information and biometric data. In this paper, the user authentication process is incorporated into cloud computing to protect against spoofing attacks.

2.2.4. Resistance to Man-in-the-Middle Attacks. One of the primary security vulnerabilities to overcome is man-in-the-middle attack resistance during authentication. An immoral actor could possibly intercept transmitted authentication messages to spoof the communicating devices into believing they are legitimate users. This allows the attacker to communicate with the server as an authorized user and collect or modify data. Spoofing can also be used against real authorized users. For example, the attacker can mimic a legitimate server and communicate with the authorized user to steal their credentials. Embedding all communicating device IDs into the protocol messages for device authentication is an effective method of preventing man-in-the-middle infiltrations. Yu et al. [18] introduced an authentication/key consensus protocol in an IoT-cloud-based setting. They designed registration/authentication stages that utilizes a fuzzy-verifier technique as well as added verification within cloud servers. This method can effectively resist insider attacks and DoS assaults.

2.2.5. Multiple Security and Privacy Properties Must Be Managed Simultaneously. The security/privacy of both data and devices are the primary concerns of any e-health systems that utilize IoT-cloud technology. This is due to the wireless communication protocols utilized by most BSNs, which tend to be inherently insecure. This insecurity leaves the system vulnerable to outside attacks that could cause grievous harm. The following are three of the most common principles for protecting IoT-cloud-based e-Health systems from outside attackers. Firstly, in order to defend the system from spoofing and the unauthorized access of data, a mutual authentication is utilized between commination devices. Secondly, the system must achieve and maintain anonymity and remain untraceable to ensure that data collected by biosensors, private personal data, or patient diagnostic data is not disclosed. Third, the system must be hardened against forgery and replay attacks. In addition, in study [2], the authors presented an extensive security/privacy framework for Healthcare 4.0, which established an integrated blockchain solution for different healthcare applications. Their framework improved patient diagnostics and ensured security/privacy of collected healthcare information.

3. Proposed IOT-Cloud-Based E-Health System

This section introduces a communication environment, a designated trust border, and outline the purpose of our suggested e-health systems that utilize IoT-cloud technology. Then, we illustrate the detailed communication steps for the proposed system that contains both discreet initialization and authentication phases.

As mentioned in the literature review, e-health systems that utilize IoT-cloud technology have the capability of managing a large variety of data, such as heart rate, temperature, diagnosis, and outcomes. This data can be analyzed to identify and calculate the severity of particular diseases. e-health systems leverage the capabilities of various hardware devices and software components to achieve the organized, controlled, and universal cloud computing incorporation. This enables fast/accurate transfer and data analysis. Moreover, it ensures high overall system quality. The analyzed data could be utilized to create graphic visualizations and warnings corresponding to a given patients health condition and stored confidentially. Moreover, data integrity and authentication are ensured through the utilization of cloud networks which implement appropriate security mechanisms for a given system.

3.1. The Communication Environment. Here, we introduce our proposed communication environment for e-health systems that utilize IoT-cloud technology. There are three essential components of our e-health communication: wearable body biosensors, LPU (Local Processing Unit), and BSN (Body Sensor Networks) server. Wearable body biosensors are always fixed onto users and classified as edge devices. These edge devices collect and transmit biodata from a given patient to LPU and BSN server for analysis. This communication is highly efficient and enables fast and accurate diagnosis and treatment in which real-time biodata such as electrocardiograph (ECG), electroencephalograph (EEG), electromyograph (EMG), and blood pressure (BP) are retrieved. After the data is collected, the BSN servers can process the data and provide bespoke treatment recommendations for patients in a timely manner. This reduced delay in diagnosis, and the treatment would greatly improve patient outcomes. Our suggested communication framework for e-health systems that utilize IoT-cloud technology

Table 1: Notations.

Symbol	Definition
P, Q	Principals
X, Y	Statements
K	Long-term secrets or secret keys
P believes X	P believes that X is correct
P sees X	A message is transmitted that contains X to P
P said X	P transmits a message that includes X within the present protocol or beforehand
P controls X	P holds authority over S; P exerts control over X and should be believed
Fresh (X)	X has not been transmitted in a message prior to the present protocol session
$P \overset{K}{\leftrightarrow} Q$	Key K is shared among both P and Q
$\{X\}_K$	Represents formula X, encrypted/secured under key K
BS_i	Identity of wearable biosensor i
LPU_j	Identity of the local processing unit j
Server	Identity of the BSN server (cloud computing)
ID_i	Public identity of BS_i
ID_j	Public identity of LPU_j
ID_s	Public identity of server
$\mathrm{ID}_{\mathrm{dj}}$	Private identity of LPU_j
$\mathrm{ID}_{\mathrm{ss}}$	Private identity of Server
$\mathrm{AID}_{\mathrm{i}}$	One-time alias identity of the wearable biosensor i
SID	A set of unlinkable shadow identities $\mathrm{SID} = \{\mathrm{sid}_1, \mathrm{sid}_2, \ldots, \mathrm{sid}_n\}$
$\mathrm{Tr}_{\mathrm{seq}}$	Track sequence number
R_{is}, R_i, R_j, R_{sid1}, R_{sidn}	Random number
$\oplus$	Bitwise operators
$\|$	Concatenation operation

employs a designated LPU and numerous biosensors to complete advanced registration on a designated BSN server. Once advanced registration is completed, assigned security authorizations shall be distributed and held onboard the biosensors, designated LPU, and designated BSN server. Moreover, assigned security authorizations are utilized to establish protected channels of communication channels and to authenticate devices. This will ensure that data confidentiality and data integrity are maintained.

Our proposed system is comprised of two phases: a system initialization phase and an authentication phase. In the initialization phase, approval will be determined for all required security elements then distributed between the communication objects (body sensors, LPU, and BSN server) through protected channels. After that, the authentication phase will be utilized for the protection of all exchanged communication/data between communicating devices. Before outlining our proposed communication protocols, we comprehensively illustrate the symbols and abbreviations utilized within this paper (Table 1).

3.2. System Initialization. Firstly, body biosensor BS_i transmits its device identity ID_i as a registration request to a designated BSN server (cloud-based server). Once the request is received from BS_i, the designated *Server* will then generate a random number (R_{is}) then utilizes its unique identity ID_s to calculate the secret key, $K_{is} = (ID_s \| R_{is} \| ID_i)$. Subsequently, the designated *Server* will compute multiple unlinkable shadow identities $SID = \{sid_1, sid_2, \ldots, sid_n\}$ for BS_i. Then, to achieve expedient identification of BS_i and avert replay attacks, a track sequence number Tr_{seq} is designated. During each authentication session, Tr_{seq} must be altered and the new Tr_{seq} will be stored both on the BS_i and *Server*. During the authentication session, the freshness of inbound BS_i requests can be verified and directly identified by the *Server* through the use of Tr_{seq} stored in a designated backend database. Then, the *Server* automatically rejects any inbound demands and then terminates communication if the integrity of the Tr_{seq} in the demand is not preserved in the backend database. If the demand remains intact, the *Server* instructs BS_i to transmit a new request while simultaneously embedding a fresh *SID* as the anonymized identity of BS_i . Lastly, the *Server* issues a security certificate $(ID_i, K_{is}, SID, Tr_{seq})$ to BS_i. Meanwhile, the *Server* maintains the same security certificate $(ID_i, K_{is}, SID, Tr_{seq})$ corresponding to each BS_i in the backend database. Moreover, registration between the *Server* and LPU_j is accomplished through a similar protocol; LPU_j transmits its identity ID_j in the form of a registration request to a designated *Server*. The designated *Server* then calculates $K_{is} = (ID_s \| R_{is} \| ID_i)$ utilizing a freshly generated public key R_{js}. It then shares a security certificate (ID_s, K_{js}, ID_j). Lastly, the *Server* maintains the security certificate (ID_s, K_{js}, ID_j) and its corresponding LPU_j in a backed database.

The compiler will verify the token and either deny authorization or add the sensor to a list of authorized devices. The compiler is connected to the physical LAN network or wireless network to receive the token key, save it to the device list, and transmit the encrypted token key and data to authorized network nodes. Once the token key is verified and validated,

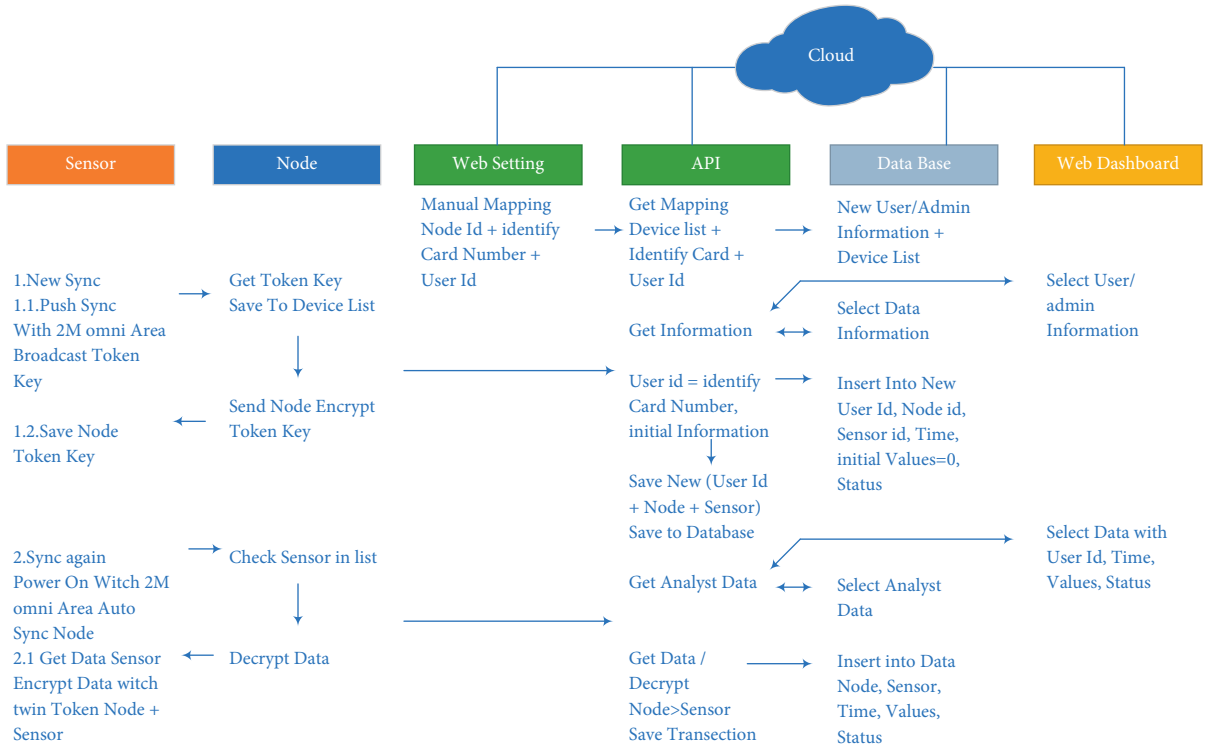


Figure 1: Authentication phase.

the token key is reset to generate a new record. Individual body sensors will receive authorization from the compiler and start gathering data. Gathered data will be encrypted and transmitted wirelessly to the compiler and stored.

3.3. Authentication Phase. In our proposed e-health systems that utilize IoT-cloud technology, we assume healthcare professionals with intelligent devices, endeavor to deliver instantaneous healthcare benefits through the use of an autonomous, noncontact information gathering and recovery apparatus. Given that our communications network based on IoT is public, an authentication protocol that is both strong and resilient is necessary to protect information exchanged between the biosensors, LPU, and BSN server. Due to the public nature of hospital networks, the network is inherently insecure. Accordingly, a robust IoT communication architecture is required for the creation of a secure channel that can be utilized for the safe exchange of data between BS_i, LPU_j, and the *Server*.

The authentication phase communication protocols are detailed in Figure 1.

Stage 1. $BS_i \longrightarrow LPU_j$: $M_{A1} = \{AID_i, (M_1, R_i, Tr_{seq}, ID_j)\}$

BS_i first generates random number R_i and calculates $M_1 = (K_{is} \oplus R_i)$ and $AID_i = (M_1 \| ID_j \| Tr_{seq})$. Next, BS_i sends $M_{A1} = \{AID_i, (M_1, R_i, Tr_{seq}, ID_j)\}$ as an authentication request to LPU_j. If the shared Tr_{seq} value between the *Server* and BS_i is not synchronized, BS_i selects a new sid_1 from the master *SID* list and consequently sets sid_1 as AID_i. Lastly, BS_i transmits an authentication request $(M_{A1} = \{AID_i, (M_1, R_i, ID_j)\})$ to LPU_j.

Stage 2. $LPU_j \longrightarrow Server$: $M_{A2} = \{M_2, R_j, ID_j, V_1, M_{A1}\}$.

Once the authentication request from BS_i is received, LPU_j will produce a random number R_j and calculate $M_2 = (K_{js} \oplus R_j)$ and $V_1 = (M_{A2} \| Tr_{seq} \| ID_j \| R_j \| K_{js})$. Then, LPU_j sends $M_{A2} = \{M_2, R_j, ID_j, V_1, M_{A1}\}$ to the *Server*.

Stage 3. *Server* $\longrightarrow LPU_j$: $M_{A3} = \{R_{s2}, Tr, V_3, V_2\}$.

Once the *Server* obtains $M_{A2} = \{M_2, R_j, ID_j, V_1, M_{A1}\}$, then the *Server* initially confirms if the tracking sequence number Tr_{seq} remains present within a given request. Supposing Tr_{seq} has concluded in M_{A2}, the *Server* executes condition (1), or else condition (2) shall be invoked.

(i) Condition (1): test Tr_{seq} legitimacy; then seek an analogous tuple through Tr_{seq} from a designated backed database. If Tr_{seq} legitimacy is confirmed, the *Server* recovers K_{is}. If legitimacy is rejected, the *Server* will terminate the current session. Alternatively, the *Server* shall validate M_2, V_1, and AID_i utilizing these subsequent algorithms:

(ii) Is the acquired M_2 equivalent to calculate M_2 expressed as $M_2 = (K_{js} \oplus R_j)$?

(iii) Is the acquired V_1 equivalent to calculate M_{A1} expressed as $M_{A1} = (Tr_{seq} \| ID_j \| R_j \| K_{js})$?

(iv) Condition (2): in the event that the *Server* is unable to locate Tr_{seq} within the request (M_{A2}, M_{A1}), the *Server* shall determine the validity/freshness of $AID_i = sid_i$ from *SID*. In the event that the *Server* is unable to recognize sid_i harvested from the backend database, the *Server* eliminates all connections and

then directs BS_i to attempt validation once more utilizing a different shadow identity sid_i.

In the event that at least one inspection agrees, the *Server* shall produce random numbers R_{s1}, R_{s2} and then designate R_{s1} as the new tracking sequence number Tr_{seq} ($Tr_{seq-new} = R_{s1}$). Subsequently, the *Server* computes $Tr = (K_{is} \| R_{s2}) \oplus Tr_{seq-new}$, $V_3 = (Tr \| K_{is})$, $V_2 = (ID_j \| R_{s2} \| K_{js})$, $SK_{is} = (K_{is} \| Tr)$, and $SK_{js} = (K_{js} \| ID_j \| R_{s2} \| R_j)$. It should be noted that the designated session key is SK_{is} and shall be applied to the subsequent protected commination between BS_i and the *Server*. Moreover, SK_{js} is designated as the session key that will be utilized by the LPU_j and *Server*. Thereafter, S transmits $M_{A3} = \{R_{s2}, Tr, V_3, V_2\}$ to the LPU_j as a response.

Stage 4. $LPU_j \longrightarrow BS_i$: $M_{A4} = \{R_{s2}, Tr, V_3\}$.

After receiving $M_{A4} = \{R_{s2}, Tr, V_3, V_2\}$, LPU_j first calculates $(ID_j \| R_j \| K_{js})$; next, it confirms that the accepted value V_2 and the calculated value $(ID_j \| R_{s2} \| K_{js})$ are equivalent. In the event that reciprocal authentication between the LPU_j and the *Server* is confirmed, bothLPU_j and the *Server* approve the utilization of secure session key SK_{js}. Lastly, LPU_j sends $M_{A4} = \{R_{s2}, Tr, V_3\}$ to BS_i . Upon obtaining M_{A4}, BS_i calculates $(Tr \| K_{js})$ and cross-references it with the obtained value V_3 . In the event that the two values are equivalent, BS_i obtains $Tr_{seq-new} = (K_{is} \| R_{s2}) \oplus Tr$ and sets $Tr_{seq} = Tr_{seq-new}$; the next new authentication session will utilize this technique. Lastly, BS_i calculates an $SK_{is} = K_{is} \| Tr$, session key that is utilized in cooperation with the *Server*. This guarantees mutual authentication between the *Server* and BS_i.

4. Security Analysis

In this section, we present our security analysis in terms of the aforementioned system criteria. The analysis of authentication is based on a logic model. Before we conduct the formal logic analysis on the proposed authentication mechanism, we define basic constructs and logic postulates. Hereafter, P and Q range over principals, X and Y range over statements, and K ranges over long-term secret keys (i.e., Table 1).

4.1. Claim 1: Robust Access Control Can Be Achieved through Mutual Authentication among Communication Entities in a BSN. We demonstrate some basic logical assumptions and constraints, where variables P and Q range over principals, X and Y range over statements, and K ranges over long-term secret keys.

(i) Statement 1: Suppose P believes $P \overset{K}{\leftrightarrow} Q$ and P perceives $\{X\}_K$; next, we presume that P believes Q said X.

(ii) Statement 2: Suppose P believes fresh (X) and P believes Q said X; next, we presume that P believes that Q believes X.

(iii) Statement 3: Suppose P believes Q holds dominion over X and P believes that Q believes X; next, we presume that P believes X.

(iv) Statement 4: Suppose P observes (X, Y) then P observes X. In addition, provided that P believes $P \overset{K}{\leftrightarrow} Q$ and P observes $\{X\}_K$, then P observes X.

(v) Statement 5: Supposing a given formula contains one component that is fresh, it is assumed that the entirety of the formula is similarly fresh. Supposing P believes fresh (X), then P believes fresh (X, Y).

Prior to analyzing the authentication process of the BSN, the following constraints must be considered:

(i) Constraint 1: BS_i and Server believe $BS_i \overset{ID_i, K_{is}, SID, Tr_{seq}}{\leftrightarrow} Server$.

(ii) Constraint 2: LPU_j and Server believe $LPU_j \overset{ID_j, K_{js}}{\leftrightarrow} Server$.

(iii) Constraint 3: *Server* believes fresh (R_i, R_j).

(iv) Constraint 4: BS_i and LPU_j believe fresh (R_{s1}, R_{s2}).

(v) Constraint 5: *Server* believes BS_i controls R_i.

(vi) Constraint 6: *Server* believes LPU_j controls R_j.

(vii) Constraint 7: BS_i and LPU_j believe *Server* controls (R_{s1}, R_{s2}).

The tangible BSN authentication protocols and procedures are defined below. Each symbol is defined, and the BSN communication protocol is illustrated in Figure 1.

(i) Stage 1: $BS_i \longrightarrow LPU_j$: $M_{A1} = \{AID_i, (M_1, R_i, Tr_{seq}, ID_j)\}$, where $M_1 = (K_{is} \oplus R_i)$, and $AID_i = (M_1 \| ID_j \| Tr_{seq})$.

(ii) Stage 2: $LPU_j \longrightarrow Server$: $M_{A2} = \{M_2, R_j, ID_j, V_1, M_{A1}\}$, where $M_2 = (K_{js} \oplus R_j)$, $V_1 = (M_{A2} \| Tr_{seq} \| ID_j \| R_j \| K_{js})$, and $M_{A1} = \{AID_i, (M_1, R_i, Tr_{seq}, ID_j)\}$

(iii) Stage 3: $Server \longrightarrow LPU_j$: $M_{A3} = \{R_{s2}, Tr, V_3, V_2\}$, where $Tr = (K_{is} \| R_{s2}) \oplus Tr_{seq-new}$, $V_3 = (Tr \| K_{is})$, and $V_2 = (ID_j \| R_{s2} \| K_{js})$.

(iv) Stage 4: $LPU_j \longrightarrow BS_i$: $M_{A4} = \{R_{s2}, Tr, V_3\}$, where $Tr = (K_{is} \| R_{s2}) \oplus Tr_{seq-new}$ and $V_3 = (Tr \| K_{is})$.

The official analysis of the BSN joint authentication protocol is detailed below:

(1) LPU_j perceives $M_{A3} = \{R_{s2}, Tr, V_3, V_2\}$; based on Stage 3, it is apparent that LPU_j has received and perceived $\{R_{s2}, Tr, V_3, V_2\}$.

(2) LPU_j believes $LPU_j \overset{ID_j, K_{js}}{\leftrightarrow} Server$; based on Constraint 2, LPU_j perceives that it shares ID_j and K_{js} with the designated *server*.

(3) Based on (1) and (2), LPU_j perceives that the *Server* said $\{V_2\}$; therefore, we can derive that LPU_j perceives that the *Server* said $\{V_2\}$ utilizing Statement 1.

(4) LPU_j perceives fresh (R_j); as R_j which is dispensed by LPU_j; LPU_j has the ability to evaluate the freshness of R_j and perceives (R_j) as fresh if it meets the specified criteria.

(5) Based on (3) and (4), LPU_j perceives that the *Server* perceives $\{V_2\}$; if it is confirmed that LPU_j

perceives that the *Server* perceives $\{V_2\}$ based on Rule 2, then the claim is supported.

(6) LPU_j perceives that the *Server* governs $\{R_{s2}\}$; based on Constraint 7, LPU_j perceives that the random number R_{s2} is directly governed by the *Server*.

(7) Based on (5), (6), and Statement 3, LPU_j perceives $\{V_2\}$; therefore, it can be confirmed that LPU_j believes $\{V_2\}$.

(8) BS_i perceives $M_{A4} = \{R_{s2}, Tr, V_3\}$; in Stage 4, BS_i is confirmed to perceive $M_{A4} = \{R_{s2}, Tr, V_3\}$.

(9) Based on Constraint 1, BS_i perceives $BS_i \overset{ID_j, K_{js}, SID, Tr_{seq}}{\leftrightarrow}$ *Server*; therefore, BS_i perceives that ID_i, K_{is}, SID, and Tr_{seq} are shared with the *Server*.

(10) Based on (8) and (9), BS_i perceives that the *Server* transmitted $\{Tr, V_3\}$; therefore, we can determine that BS_i perceives that *Server* transmitted $\{Tr, V_3\}$ utilizing Statement 1.

(11) Based on Constraint 4, BS_i perceives (R_{s2}) as fresh; BS_i accepts the freshness of R_{s2}.

(12) Based on (10) and (11), BS_i perceives that the *Server* perceives $\{Tr, V_3\}$; it is guaranteed that BS_i perceives that the *Server* perceives $\{Tr, V_3\}$ because of Statement 2.

(13) Based on Constraint 7, BS_i perceives that the *Server* governs $\{R_{s2}\}$; BS_i perceives that the random number R_{s2} is directly governed by the *Server*.

(14) Based on (12) and (13), BS_i perceives $\{Tr, V_3\}$; therefore, we can determine that BS_i perceives $\{Tr, V_3\}$.

The finalized results of our simulation are shown below:

(i) LPU_j perceives that the *Server* perceives $\{V_2\}$ based on (5).

(ii) LPU_j perceives $\{V_2\}$ based on (7).

(iii) BS_i perceives that *Server* perceives $\{Tr, V_3\}$ based on (12).

(iv) BS_i perceives $\{Tr, V_3\}$ based on (14).

Assuming the designated *server* is reliable/trustworthy and based on the results of (5), (7), (12), and (14), both BS_i and LPU_j can authenticate each other on the designated *Server*.

4.2. Claim 2: BS_i Anonymity and Untraceability Can Be Ensured. During the authentication phase, we adopt random numbers, R_i, R_j, R_{s2}, and utilize them to randomize any communicated messages, namely, AID_i, M_1, M_2, V_1, Tr_{seq}, Tr, V_2, and V_3, where some are involved with BS_i. Firstly, in the BSN, Tr_{seq} is employed as a unique single-use token that enables the rapid identification of BS_i due to the fact that Tr_{seq} does not contain any data relevant to BS_i. Moreover, Tr_{seq} is actively refreshed following every successfully completed authentication session. Tr_{seq} does not reveal any information about BS_i. Secondly, a secret identity AID_i (occasionally a single-use/valueless *si d* shall be utilized) will be circulated between BS_i and the *Server* and then shall be analyzed by the *Server*. Nobody is capable of tracing or identifying BS_i with the exception of the *Server*. Thirdly, based on the following equation expressed as $M_1 = (K_{is} \oplus R_i) \oplus Tr = (\|K_{is} R_{s2}) \oplus Tr_{seq-new}$ and $V_3 = (Tr \| K_{is})$, it is clear that these three variables are randomized through the use of the random numbers R_i and R_{s2}. The random numbers cannot be recycled between sessions. Given the aforementioned arguments, we believe that the anonymity and untraceability of BS_i can be ensured and maintained.

4.3. Claim 3: Resistance against Spoofing Attacks and Data Confidentiality Can Be Ensured. In the aforementioned unit, basic authentication/login mechanisms that do not utilize session key agreements are insufficient and cannot ensure security in any way. The primary element required to ensure security is the establishment of two session keys. The first session key is approved by BS_i and the *Server*, and the second key is appointed by the LPU_j and the *Server*.

Within the proposed BSN, two unique session keys, $(SK_{is} = (K_{is} \| Tr)$ and $SK_{js} = (K_{js} \| ID_j \| R_{s2} \| R_j))$, will ultimately be created and utilized to secure data transmissions between BS_i, LPU_j, and *Server*. Moreover, we assert that all the secure communication channels between BS_i, LPU_j, and *Server* are provided by the proposed scheme. Additionally, the *Server* selects two highly volatile secrets (K_{is}, K_{js}) in order to secure all communication transpiring within the authentication phase of the proposed scheme. Without knowledge or perception of the two selected secrets, it is exceedingly hazardous for bad actors to disrupt the proposed authentication schemes or salvage any meaningful data from communicated ciphertexts. Therefore, confidentiality of the data is ensured and maintained.

4.4. Claim 4: Resistance against Common Electronic Attacks (Man-in-the-Middle, Location Spoofing, and Replay) Can Be Ensured. Bad actors could intend to deceive/bypass authorized communication objects (BS_i, LPU_j, and *Server*) through the use of imitation messages. The expedient and efficient identification of counterfeit messages and the elimination/mitigation of possible threats within an IoT-based healthcare setting is growing increasingly imperative. Messages counterfeited by bad actors come in a variety of novel forms. Moreover, these malicious tricks could be initiated from within the IoT systems inherently heterogeneous network architectures. It follows that an inspection scheme with the capability of identifying fake communications and then, subsequently, preventing harmful attacks would be essential. In the proposed scheme, it is virtually impossible for a bad actor to forge genuine messages such as AID_i, M_1, M_2, V_1, Tr, V_2, and V_3 in the absence of the knowledge of K_{is} and K_{js}. Therefore, it is exceedingly difficult to launch forgery attacks. Additionally, the *Server* selects two highly volatile secrets because their volatile characteristics are highly resistant to brute-force attacks. Assuming the bad actor successfully harvests and transmits validated recycled messages used in an earlier authentication

session to a potential target. Then, a robust verification protocol utilizes a unique single-use key for all discrete sessions. A cursory inspection of the random number's freshness will determine their verification and validity. Consequently, the proposed scheme has incorporated a high degree of replay attack resistance. Moreover, our proposed scheme, respectively, incorporates details of BS_i and LPU_j into M_1 and M_{A1}. In regard to untraceability, BS_i is secure within M_1. Only the *Server* is capable of executing an untraceable attack on BS_i, given that K_{is} is unknown to other entities. In regard to the *Server*, we present a robust verification protocol that is capable of identifying location-spoofing attacks. Within our proposed scheme, we incorporated a natural resistance to location-spoofing attacks. Lastly, we examine our schemes' toughness and ability to resist man-in-the-middle attacks. Moreover, we investigate if a bad actor could intrude on the authentication session and attempt to deceive authorized communication objects by impersonating legitimate by utilizing forged/scavenged messages. Our proposed scheme incorporates discrete communication object identities into all protocol messages; this enables mutual authentication between authorized communication objects: $K_{is} = (ID_s \| R_{is} \| ID_i)$, $K_{is}' = (ID_s \| R_{is} \| ID_j)$, $M_1 = (K_{is} \oplus R_i)$, $Tr = (K_{is} \| R_{s2}) \oplus Tr_{\text{seq-new}}$, $V_3 = (Tr \| K_{is})$, and $V_2 = (ID_j \| R_{s2} \| K_{js})$ to ensure that such an event does not occur. It is evident that each discrete communication object's identity is contained in all communication within the authentication session. Forged/scavenged messages that contain illegal identities are unusable due to the irrevocability our scheme's authentication protocols. Moreover, previously sent messages cannot be modified or recycled for reuse due to our robust authentication protocol that utilizes two large random numbers with incorporated auxiliary values. Messages can be encrypted by anyone via the public key cryptosystem but can only be decoded by someone who knows the designated random numbers. Therefore, resistance against man-in-the-middle attacks is ensured.

4.5. Claim 5: Simultaneous Management of Security and Privacy Properties Can Be Achieved through the Use of Nonrepudiation Transaction Mechanisms. Within an IoT environment, a protocol with the ability to ensure that a collection of mobile objects, such as gateways and sensors, can coexist within the same time and place is integral at foiling probable fraud events. Our scheme's protocol requires the identity information for both BS_i and LPU_j to be incorporated into all authentication messages (M_{A1} and M_{A2}) to verify all object identities and the location of the ongoing transaction session. According to our proposed scheme, an authorized timestamp is contributed by the *Server*, $M_{A2} = \{M_2, R_j, ID_j, V_1, M_{A1}\}$ which will be utilized as evidence in the ongoing transaction between BS_i and LPU_j after the validity of M_{A2} is established. Therefore, it is evident that the proposed scheme contains robust nonrepudiation transaction characteristics.

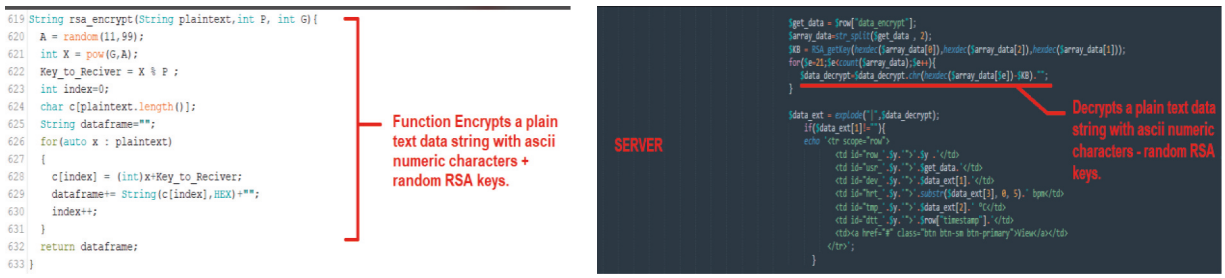
5. Implementation

This section summarizes the implementation of our prototype such that we could validate the practicality and usefulness of our proposed e-health system that utilizes IoT-cloud technology. Our communication process requires coordinated interaction among all the prototype hardware and software components. This communication protocol must be customized based on the intelligent devices, transmissions, interoperability, internal protocols, access control requirements, security encryption/decryption requirements, and privacy requirements of any given system. We implement security protocols at the component level (Raspberry PI, biosensors) and then determine the equivalent computational toll for each component of the system. Table 2 illustrates the implementation environment, in which we utilized a Raspberry PI as the designated local server that is connected to a wireless router/access point and a smart object (body sensor). The essential security elements are all programmed using Arduino Software (C and JavaScript) and executed natively on the Raspberry PI. Lastly, all data is encrypted at the Raspberry PI and transmitted to a cloud API.

First, we construct and execute a private network on a laptop computer with an Intel i5-6200U CPU, Nvidia GeForce 930M GPU, and 12 GB of RAM running Windows 10. Second, we utilize a TP-Link Soft Access Point to connect the laptop and Raspberry PI via a Direct (P2P) WiFi connection. The Raspberry PI has an ESP 8266 microcontroller and 125 MB RAM, which is capable of wakeup and packet transmission in less than 2 milliseconds and standby power consumption of less than 1.0 milliamperes. The Raspberry PI microcontroller can be programmed in C language to act as a hub. Therefore, we utilize the Raspberry PI as an intelligent LPU object that can process and manage communication amongst IoT objects (body sensors) and cloud servers. Third, we designed a client application to be implemented on user facing devices such as smartphones, laptop/desktop computers, or any devices that can run cloud API's. This cloud API stores the data using MySQL to control a database that authorized users can access at any time. Therefore, our API platform is easily customizable, flexible, and scalable and can be combined with our Raspberry PI platform to perform authentication. This system is then tested for practicability, efficiency, and feasibility. Lastly, to balance system efficiency, scalability, and security robustness, a secure RSA algorithm [15, 16] that utilizes a random number generator and a bitwise exclusive operation are integrated into the system. In order to be effective, the communications system channel must be coupled to at least one terminal that has an encoding device and to at least one terminal that has a decoding device. However, during the implementation phase of the system, the values ID_i, ID_s, ID_j, R_{is}, R_{js}, $si\ d$, Tr_{seq}, R_i, R_j, R_{s1}, and R_{s2} are all set to a 32-bit random number generator operation to ensure security density. Bear in mind that within our experiment, IoT object access, is stringently

Table 2: Implementation environment.

Environment	Details
Raspberry PI	32-bit microcontroller Node MCU ESP8266 Micro 26 MHz to 52 MHz, memory 125 MB instruction 4 MB storage, built-in flash, 10 GPIO-Pins, SPI, I2C Integrated TCP/IP protocol stack
Software platform	Arduino software C, JavaScript
Cloud API	PHP, HTML, MySQL



(a) (b)

Figure 2: (a) Client encryption/decryption. (b) Server encryption/decryption.

governed in accordance with the designated privacy policy. In the absence of user consent, access to IoT objects is prohibited.

Our implementation integrates readily available hardware devices, microcontrollers, and communication hardware to transmit data to the cloud system. The demonstrated computation cost required in terms of the cryptosystem using a programmable microcontroller is reduced compared to conventional methods. Our prototype utilizes Arduino software that can be programmed to communicate in a reader/sender configuration via the I2C synchronous serial protocol. The information will be sent bit by bit until the receiving device executes the request and transmits data back while the I2C protocol allows for each authorized device to have its own unique address and for both devices to take turns communicating over a single line. The computation cost of the proposed cryptographic operation in which the experimental conditions are the same as that of [16]. In addition, our Raspberry PI microcontroller board is programmed using Arduino software. Then, we can program control of the physical systems via read/write or analog/digital signals by integrating C programming, JavaScript programming, and RSA encryption and decryption to work in conjunction with Arduino software to significantly improve efficiency and security. Cryptography is defined as a mathematical means by which data can be secured between two parties, whereby the sender and receiver use the same designated algorithm for protection. Therefore, potential intruders can neither understand the content of the messages nor alter them. Any party that knows the designated key (receiver) has the capability of decrypting the received ciphertext. The complete design is divided into stages that start by utilizing a random number generator to produce two large prime numbers. Then, a primality-testing unit is initialized, after which the last stage is the encryption/decryption processes. The system contains text-number conversions that were executed based on ASCII coding/decoding with extended lists that include 255 different codes, as shown in Figures 2(a) and 2(b). Furthermore, we illustrate our communication process utilized in our proposed e-health system that utilizes IoT-cloud technology in Figure 3.

In the following, we simplify our proposed system by dividing it into two main storage types: cloud computing and local storage, as presented in Figure 4.

(i) Local storage: it refers to dedicated IoT devices with on-site or local network memory to store applications, services, and collected data. This design improves reliability, interoperability, processing, and transmission during offline operation. Local storage is a link between the Raspberry PI module storage and host system storage that allows for streamlined protocols and local data processing. Moreover, RSA encryption/decryption algorithms are integrated into the local system to secure it from external attacks, specifically, from attacks against the communication between devices via routers, Wi-Fi networks, LAN networks, or cellular networks.

(ii) Cloud computing storage: cloud API web services are applications, services, and physical devices that can store collected data. Cloud computing reduces the implementation complexity of the system. Moreover, the security of the system is ensured through the utilization of RSA en/decryption at the cloud security layer. This ensures that only authorized users, such as administrative users, developers, and patients, can access or manipulate stored data.

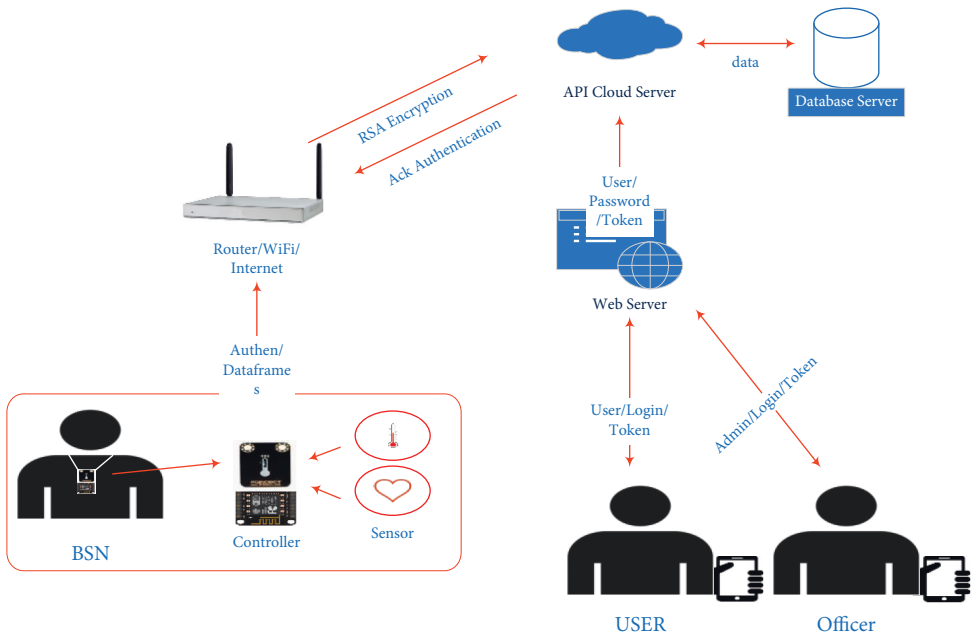


Figure 3: Communication process.

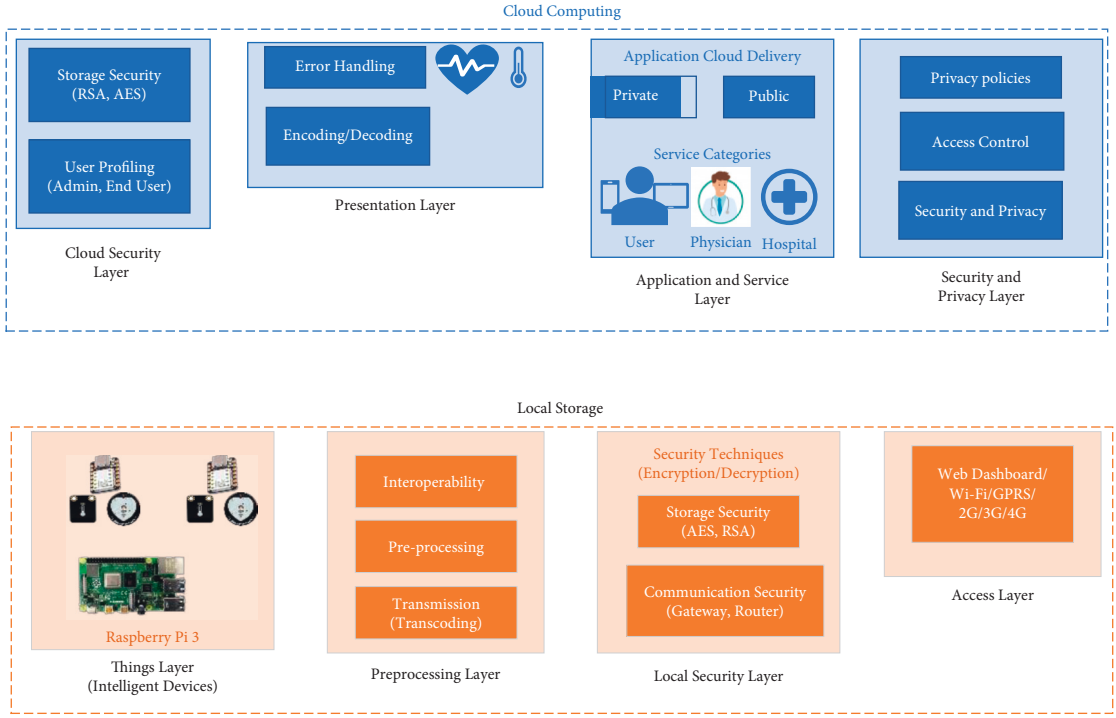


Figure 4: Main storage types for IoT devices.

The presentation layer enables users and patients to access their medical information via a web server or authorized Internet session.

The implementation architecture includes IoT sensor devices, equipment, and application servers. It allows these interconnected devices to communicate while utilizing custom user interfaces responsible for collecting data from network-connected sensors and equipment. When a user session is completed, any updates or modifications to the data or information are synced with the cloud servers. e-Health systems can utilize the aforementioned user interfaces to obtain and assess the health condition of its users then counsel them accordingly. Furthermore, the user interfaces are connected to cloud servers through the use of

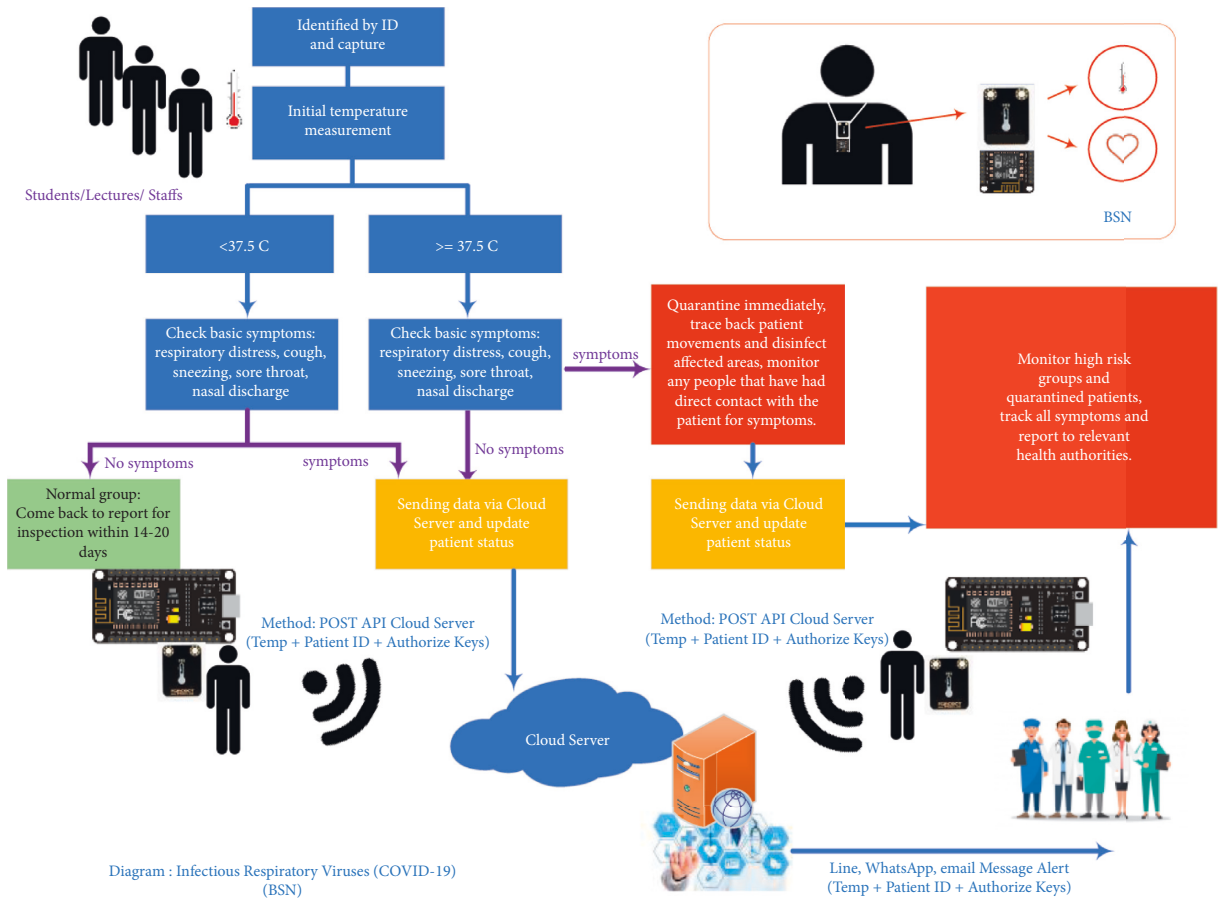


Figure 5: Utilizing IoT-cloud-based e-health system for COVID-19 detection and tracking.

middleware that facilitates intercloud communication with the cloud database. The middleware utilizes cloud API web services to transfer secure information between the web-based interfaces and the cloud servers. Moreover, middleware facilitates communication with the required databases based on user-generated requests. The cloud API is specifically designed to control authentication, authorization, and data processing. Lastly, data flow is consistent throughout the layers and robust cloud security mechanisms, such as RSA, will be implemented at every layer.

We implemented our system for the purpose of IoT-cloud-based e-health management during the worldwide COVID-19 pandemic. The implemented system reduces the complexity of storing information, increases communication efficiency, and improves the performance and accuracy of a healthcare system. Results were verified using COVID-19 spread as a case study scenario. The proposed implementation architecture utilizes various web services and security services which are exclusively cloud-driven and are further implemented as middleware. We analyze the security of communication procedures to identify weaknesses so that improvements can be implemented to achieve the principal security requirements. Intelligent body sensors were issued to patients in the field. This enables nurses/physicians to more effectively monitor and administer care to their patients. Nurses/physicians can utilize their mobile gateways for the collection of live data and to deliver quicker, superior-quality healthcare services to patients. Further patient data mining and analysis can be utilized to build predictive models. It is obvious that the implemented system offers robust security density with a reasonable computational cost. The implementation simplifies the system, from the capture of data to the storage/analysis of data. Our system utilizes embedded sensors and equipment sensors controlled by an LPU node. The LPU processes the health data, displays it locally, and transmits it to the cloud servers. The designated physician will receive daily emails with notifications in the application or webserver, containing the updated health data of their patients. The application enables both the health care personnel and the patients to access and monitor relevant health data. Additionally, security/privacy protocols including the authentication of users and services, encryption of communication, accountability tracking, and data anonymization are utilized to secure the system from attacks. Finally, users who interacted with the e-health system that utilizes IoT-cloud technology are allowed to track potential COVID-19 symptoms. In addition, users can receive alarms and visualizations of the collected live data and react to developing situations, as shown in Figure 5.

In our implemented system, first, the system identifies a user's ID and collects initial baseline measurements of the user's temperature, heart rate, blood pressure, and so on.

Then, it uploads the data to a designated private network and sends it to the cloud API database. We utilize wearable body sensors to collect temperature data over time. If a user's data is abnormal, that is, body temperature measures greater than 37.5°C, then the user will be classified as possibly having contracted COVID-19, based on observed symptoms. We separate symptoms into two categories: the normal group (no symptoms during the 14-day reporting period) and the abnormal group (displaying symptoms during the 14-day reporting period). In the event that symptoms are confirmed in a user, all that user's data will be transferred to designated Cloud API servers and their health status will be updated. Once a user displays COVID-19 symptoms, we recommend immediate quarantine and contact-tracing. Additionally, disinfection of affected areas, active monitoring of any people who had direct contact with the patient, and possible quarantine of contacted individuals are required to control the potential spread of the virus. All patient data will be continually collected, transmitted, and updated on our cloud servers. Lastly, at-risk groups, such as physicians who have direct contact with many possibly infected individuals, should track all symptoms and report them to relevant health authorities in a timely manner. Many users, including physicians, nurses, and other stakeholders can access collected data from the public cloud and private cloud via authorized web portals.

6. Conclusion and Future Work

The growth/development of IoT applications has been exceptionally fast. This growth has proven to be useful in a multitude of industries. Moreover, there have been progressive changes in the administration and operation models of medical organizations across the world. Numerous intelligent IoT objects are all interconnected in order to create a widespread IoT-based network architecture. However, a new network helps expose new security challenges. Our research has established a secure e-health management system utilizing cloud computing and IoT-based BSN structures. We propose an authentication mechanism to ensure that major security requirements are satisfied. Based on our implementation, our proposed system can be effectively executed on low-powered LPU testbeds, such as a Raspberry PI, and the system ensures greater security than historically available alternatives. Moreover, we performed a rigorous formal analysis to confirm the security, privacy, efficiency, and robustness. The implementation results prove that our IoT-cloud-based e-health management system is practicable. We used Arduino microcontroller units to implement and test RSA encryption and decryption keys. Therefore, it is evident that the security system is considered beneficial for sensor networks, as it can be connected while near a sink, particularly if it supports rechargeable batteries with easily interchangeable cells. However, system efficiency could be further improved with the future development of more efficient leaner codes, algorithms, and architectures. In the future, this design could be modified to utilize a larger bit size for encryption and decryption by modifying the algorithm. Lastly, numerous public and symmetric key cryptographic algorithms may be designed and verified based on design sets, such as DES, AES, and many others.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan, grant nos. MOST 109-2221-E-259-011-MY2, MOST 110-2926-I-259-501, MOST 110-2629-E-259-001, MOST 110-2218-E-011-007-MBK, 111-2218-E-011-012-MBK, and 110-2634-F-A49-004.

References

[1] N. Yousefnezhad, A. Malhi, and K. Främling, "Security in product lifecycle of IoT devices: a survey," *Journal of Network and Computer Applications*, vol. 171, p. 102779, 2020.

[2] J. J. Hathaliya and S. Tanwar, "An exhaustive survey on security and privacy issues in Healthcare 4.0," *Computer Communications*, vol. 153, no. 1, pp. 311–335, 2020.

[3] G. Gardašević, K. Katzis, D. Bajić, and L. Berbakov, "Emerging wireless sensor networks and internet of Things technologies-foundations of smart healthcare," *Sensors*, vol. 20, no. 13, p. 3619, 2020.

[4] C. Butpheng, K.-H. Yeh, and H. Xiong, "Security and privacy in IoT-cloud-based e-health systems-A comprehensive review," *Symmetry*, vol. 12, no. 7, pp. 1191–1235, 2020.

[5] X. Wang and S. Cai, "Secure healthcare monitoring framework integrating NDN-based IoT with edge cloud," *Future Generation Computer Systems*, vol. 112, pp. 320–329, 2020.

[6] S. Ray, H. Kathuria, K. Chakravarty et al., "Seronegative panencephalitis complicated by viral encephalomyelitis in a case of Good's syndrome - a neuropathological report," *International Journal of Neuroscience*, vol. 99, pp. 1–6, 2020.

[7] M. A. D. Shewale and S. V. Sankpal, "IOT based smart and secure health care system Analysis & data comparison," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 1, pp. 394–398, 2020.

[8] S. Chenthara, K. Ahmed, H. Wang, and F. Whittaker, "Security and privacy-preserving challenges of e-health solutions in cloud computing," *IEEE Access*, vol. 7, pp. 74361–74382, 2019.

[9] M. Koutli, N. Theologou, A. Tryferidis et al., "Secure IoT e-health applications using VICINITY framework and GDPR guidelines," in *Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 263–270, IEEE, Santorini, Greece, August 2019.

[10] A. K. Chattopadhyay, A. Nag, D. Ghosh, and K. Chanda, "A secure framework for IoT-based healthcare system," in *Proceedings of the International Ethical Hacking Conference 2018: EHaCON 2018*, October 2019.

[11] I. Alihamidi, A. Ait Madi, and A. Addaim, "Proposed architecture of e-health IOT," in *Proceedings of the 2019, The international conference on Wireless Networks and*

Mobile Communications (WINCOM), pp. 1–7, Fez, Morocco, November 2019.

[12] A. M. Rahmani, T. N. Gia, B. Negash et al., "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: a fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[13] K.-H. Yeh, "A secure transaction scheme with certificateless cryptographic primitives for IoT-based mobile payments," *IEEE Systems Journal*, vol. 12, no. 2, pp. 2027–2038, 2018.

[14] K.-H. Yeh, "A secure IoT-based healthcare system with body sensor networks," *IEEE Access*, vol. 4, pp. 10288–10299, 2016.

[15] L. Ding, Z. Wang, X. Wang, and D. Wu, "Security information transmission algorithms for IoT based on cloud computing," *Computer Communications*, vol. 155, no. 1, pp. 32–39, 2020.

[16] Q. A. Al-Haija, M. A. Tarayrah, H. Al-Qadeeb, A. Al-Lwaimi, and A. Al-Lwaimi, "A tiny RSA cryptosystem based on Arduino microcontroller useful for small scale networks," *Procedia Computer Science*, vol. 34, pp. 639–646, 2014.

[17] K. Stepien, A. Poniszewska-Maranda, and W. Marańda, "Securing connection and data transfer between devices and IoT cloud service," *Integrating Research and Practice in Software Engineering: Studies in Computation Intelligence*, vol. 851, pp. 83–96, 2020.

[18] Y. Yu, L. Hu, and J. Chu, "A secure authentication and key agreement scheme for IoT-based cloud computing environment," *Symmetry*, vol. 12, no. 1, pp. 150–215, 2020.