applied sciences

MDPI

Article

BSNCare+: A Robust IoT-Oriented Healthcare System with Non-Repudiation Transactions

Kuo-Hui Yeh

Department of Information Management, National Dong Hwa University, Hualien 97401, Taiwan; khyeh@mail.ndhu.edu.tw; Tel.: +886-3-863-3117

Academic Editor: Christos Verikoukis
Received: 30 September 2016; Accepted: 6 December 2016; Published: 9 December 2016

Abstract: Recently, the rapid advancement in technologies of modern intelligent objects has led to a new network paradigm, called the Internet of Things (IoT), in which every networked and automated object has been connected in a pervasive manner. New types of IoT-based application services are thus presented. In a healthcare oriented environment, the usage of IoT has brought opportunities for assisting physicians (or nurses) to provide on-demand and real-time body-care services to patients with higher accuracy and better efficiency. However, while IoT-oriented techniques deliver such advantages, they may encounter system security vulnerabilities and patient privacy threats not seen in the past. In this paper, we propose a robust IoT-based healthcare system, called BSNCare+, in which body sensor networks (BSNs) are adopted as the underlying communication architecture. In the proposed healthcare system, we exploit lightweight crypto-primitives to construct a secure communication mechanism that does achieve data confidentiality and entity authentication among intelligent body sensors, the mobile gateway and the backend BSN-Care server. In addition, we evaluate the performance of the proposed healthcare system using the Raspberry PI series platform. The results show the practicability and feasibility of BSNCare+.

Keywords: authentication; healthcare; Internet of Things (IoT); privacy; security

1. Introduction

The ever-increasing advancement in information and communication technologies has led to a new era in the successful development of IoT (Internet of Things)-based services, which extends the concept of the Internet and delivers versatile types of pervasive computing applications not seen in the past. In particular, in the modern healthcare environment, IoT-oriented technologies have brought novel opportunities for hospital administrators and physicians to provide on-demand and real-time body-care services to those (usually the patient) in need of higher accuracy and better efficiency. The architecture of the body sensor networks (BSNs) is a collection of tiny and low-powered wireless body sensors for monitoring the patient's health status in a real-time manner. BSNs have become one of the most popular core technologies in developing IoT-based healthcare applications. For example, the patient's bio-features can be monitored in real time via bodily equipped or embedded bio-sensors. The physicians (or nurses) can thus assess the personal health status of their patients more effectively. Medical errors can be avoided, and the treatments provided by the hospital will be more on-demand in nature. Furthermore, with pre-defined system settings tailored to the patient, potential critical emergencies experienced by the patient can be prevented more effectively. Bio-sensors for detecting electrocardiography (ECG), electroencephalography (EEG), electromyography (EMG), blood pressure (BP), temperature and motion have been commercialized for end users in this IoT concept-oriented generation. It is highly exciting to see the opportunity for developing and deploying an IoT-based healthcare system providing significant benefits to patients in the real world. Healthcare

has thus advanced to a new era with a whole new perspective on the application development of IoT-based architectures.

While IoT-oriented techniques have brought the opportunity for innovative application development, this opportunity has also brought new security challenges to IoT-based applications. For instance, *Hello Barbie*, a novel IoT-based commercial product for children, reveals a potential privacy threat, which allows the attacker to spy on us, our family and everything via camera and voice-interaction functionalities in the house [1]. In general, every sensor in the IoT represents a potential risk of system vulnerability. That is, each sensor may be a vulnerable entry point for malicious attacks. The security issue, i.e., how to effectively protect an IoT system with an appropriate security level, i.e., providing data confidentiality, integrity and privacy during data collection, has been promptly focused on during these years. The academic community has put great efforts into this important and promising research field. There seems to be a general expectation for a new and revolutionary security solution tailored to IoT-based sensors. However, because the sensors adopted in IoT applications usually have a specific defined mission with limited resources available to accomplish it, the traditional security solutions, such as firewalls, Intrusion Detection System (IDS) and Intrusion Prevention System (IPS), are not suitable for implementation on these kinds of sensors. Hence, the refinement of traditional security solutions to fit specific security requirements of IoT-based sensors is one of the most promising ways for securing IoT-based application systems.

Based on our observations, great efforts have been put into the development of secure IoT-based systems. In 2013, Yao et al. [2] proposed a lightweight authentication scheme for secure multicast functionality in small-scale IoT networks. The proposed mechanism mainly utilizes two effective properties, i.e., (1) the absorbency property and (2) the one-way or quasi-communicative property, of a fast accumulator refined by them to establish a lightweight multicast authentication scheme. The authors then evaluated seven principal criteria, required by multicast authentications for resource-constrained applications. Next, Kawamoto et al. [3] presented a location based authentication scheme embedded with an effective data collection mechanism for IoT networks. In the proposed system, the parameters related to network control can be adjusted dynamically to satisfy the real-time requirements of the system and the corresponding network environments. This property improves the authentication accuracy of the proposed system. In addition, the optimization of the authentication accuracy is investigated. Furthermore, Hernández-Ramos et al. [4] developed lightweight authentication and authorization procedures, which are compliant with the Architectural Reference Model (ARM) from the EU FP7 IoT-A project, for constrained IoT-based sensors. The proposed schemes can also be combined with other standard technologies and can be utilized to form security plans for the life cycle of IoT device development.

In 2014, Tennina et al. [5] introduced a healthcare system architecture, which is fit for wireless sensor networks. The authors proposed a communication protocol stack for WSN (wireless sensor networks)-based pervasive healthcare applications in which four layers embedded with various technologies, such as IEEE 802.15.4 MAC (medium access control) layer, network layer, middleware services and application layer, are introduced. Implementations including network coding and distributed localization scheme are demonstrated on a WSN testbed as a proof of concept of their proposed architecture. Next, Kartsakli et al. [6] presented a cloud-assisted MAC protocol, called CLNC-MAC (cloud-assisted RLNC-based MAC protocol), which exploits random linear network coding (RLNC) techniques to guarantee the completeness of the data transmission in healthcare applications regardless of the channel conditions of the relay network. Analyses on CLNC-MAC are then performed via queuing theory in which essential metrics, i.e., data throughput, data delay and energy efficiency, are examined through Monte Carlo simulations. With the performance assessment, the authors demonstrated that their proposed protocol is superior to existing methods in the presence of channel errors for a two-hop relay network. In the following year, Kartsakli et al. [7] further presented a framework for cloud-assisted ambient assisted living (AAL) applications. The proposed framework provides two planes for control operation and data operation and thus delegates network coordination

tasks to the cloud. An efficient MAC protocol is developed in which the RLNC technique is adopted in a cooperative relay network to pursue better performance efficiency. Later, Kartsakli et al. [8] discussed the opportunity and challenge for Machine-to-Machine (M2M) mHealth systems in terms of a wireless communications perspective. Eight major challenges, such as heterogeneous devices and traffic, wireless propagation characteristics, energy efficiency, quality of service, reliability, interference and coexistence, topology, security and technology integration, must be thoroughly considered for the development of M2M mHealth systems. Recently, Gope and Hwang [9,10] presented two authentication schemes, i.e., BSN-Care [9] and USM-IoT [10], for IoT networks. These two schemes are designed to fit the architecture (and its specific security requirements) of body sensor networks and distributed wireless sensor networks (WSN), respectively. Both of these two schemes have a similar concept in the design of the proposed authentication mechanisms. That is, lightweight crypto-modules, such as one-way hash function, random number generator operation and bitwise exclusive-or operation, are adopted for providing system efficiency and security robustness at the same time. In 2015, the authors [10] first presented an authentication protocol for distributed wireless sensor networks to satisfy important security properties such as mutual authentication, sensor anonymity and un-traceability, system scalability, and resistances to impersonation attack, replay attack and cloning attack. In 2016, the authors [9] further proposed an authentication mechanism for IoT-based healthcare system, which is based on body sensor networks consisting of lightweight and healthcare-oriented body sensors. The authors then investigated the security density and protocol efficiency via BAN logics analysis and computation cost comparison.

Even though Gope and Hwang's authentication schemes, i.e., BSN-Care [9] and USM-IoT [10], are designed to fit the security requirements of body sensor networks and distributed wireless sensor networks, respectively, the underlying communication architectures are client-server based and client-client-server based from the point of view of authentication analysis. Unfortunately, there still exist space for improvements of these two protocols in terms of the authentication viewpoint. First, the two schemes lack session key agreement and a man-in-the-middle attack may be launched. Based on the authentication process of BSN-Care [9] and USM-IoT [10], these two proposed systems provide user login functionality without session key agreement. A man-in-the middle adversary may simply interrupt the login request and response message and pretend that it is as a legal BSN-Care server or legal client. A fake and phishing website counterfeited as a legal BSN-Care server is a possible technique to achieve a man-in-the-middle attack. Second, we find that bitwise exclusive-or module is inappropriately used in their schemes. That is, the form of "$M \oplus key$" may be more vulnerable than the form of "$H(M \oplus key)$", where key is a secret and M is a message. From this point of view, we find that there exist insecure designs of protocol messages in Gope and Hwang's schemes, i.e., $N_x = K_{ls} \oplus N_1$, $N_x = K_{sh} \oplus N_S$, $N_y = K_{ch} \oplus N_C$. The security density is based the robustness of random number and the exclusive-or function. This design preserves the robustness resisting to only "cipher-text only" level attacks and we thus suggest the form of "$H(M \oplus key)$" when designing an robust authentication scheme. Third, their schemes are not secure against a location spoofing attack. Apparently, in Gope and Hwang's schemes legitimate but malicious cluster heads (or servers) can tell anyone that the user, authenticated at previous sessions, is now in his/her range even though the user is not there.

Due to the tradeoff among network heterogeneity, system security and computation efficiency, in this paper we would like to develop a robust IoT-oriented healthcare system for body sensor networks, called the BSNCare+, which is based on improvements on the methods presented in BSN-Care and USM-IoT. The rest of this paper is organized as follows. In Section 2, we introduce the underlying IoT communication architecture of the proposed BSNCare+ system. Section 3 illustrates the details of our proposed BSNCare+ system, and Section 4 performs the security analysis and related discussions of the BSNCare+. Then, we present performance evaluations through a demo system implementation with an IoT-oriented programming platform, i.e., Raspberry PI 2. The practicability of the BSNCare+ is thus demonstrated. Finally, we provide concluding remarks in Section 6.

2. The Underlying IoT Communication Architecture

In this section, we present the IoT communication architecture, which our proposed BSNCare+ system is modeled on. Figure 1 illustrates the applied scenarios, such as the IoT-based ward, the IoT-based rehabilitation environment and the IoT-based healthcare place, as well as the detailed communication procedures of the underlying network architecture. In these applied scenarios, fixed sensors and intelligent wearable devices are deployed in the field and on the patient, respectively, to support the nurse/doctor in activities of patient care, rehabilitation and even healthcare. There are three indispensable components in the proposed IoT communication architecture: a BSN-Care server, mobile gateway, and edge devices (i.e., body sensors or wearable bio-devices). These edge devices are responsible for collecting environment parameters and bio-data from the patient. The nurse/doctor is then able to retrieve this data via his/her intelligent handheld device as a mobile gateway, which is responsible for retrieving data from the edge devices. With the specific bio-data and environment parameters, the nurse can recognize and satisfy the patient's need in a faster and more efficient way. For example, with the patient's bio-data, i.e., ECG, EEG, EMG and BP, the nurse can provide more accurate and timely treatment services and reduce medical delay accordingly. In the proposed IoT communication architecture, all the edge devices and the mobile gateway need to perform registrations at the BSN-Care server. After registration, security credentials are shared and stored among the edge device, the mobile gateway and the BSN-Care server. When the BSNCare+ system is in operation, all interactions between any two operating entities must be based on a robust identification and authentication mechanism for establishing a secure communication channel. The security credentials are exploited to achieve the goal of entity authentication, and, in addition, data confidentiality and data integrity can thus be guaranteed via the secure communication.

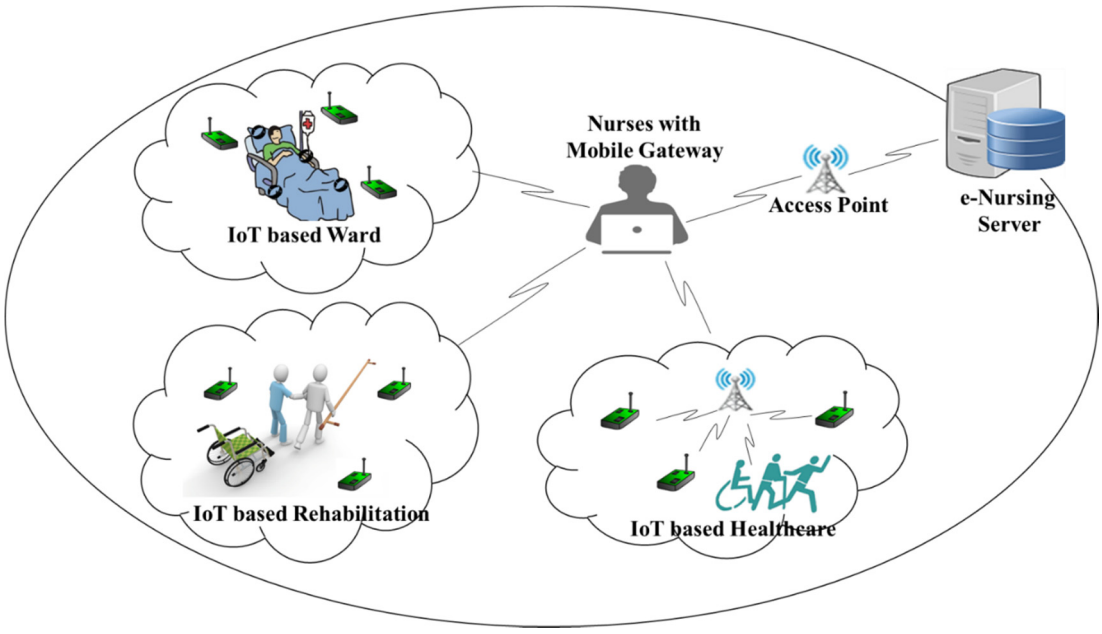

Figure 1. The proposed IoT-based communication architecture.

3. The Proposed Healthcare System: BSNCare+

This section introduces the proposed BSNCare+ system, which consists of a registration phase and an authentication phase. In addition, we present the trust boundary and the desired objectives of the proposed scheme.

By means of the characteristics of contactlessness and efficiency of data retrieval, intelligent objects are broadly deployed to build various IoT-based application systems for the purpose of pursuing

a better quality of services. In this section, we introduce the proposed BSNCare+ system in which IoT-based healthcare, rehabilitation and biomedical related equipment are adopted (or embedded) for both the patient and the involved environment as the edge devices (e.g., Figure 2). The nurse can utilize his/her mobile gateway for real-time data collection and provide better-quality healthcare services to the patient. All the sensing data will be forwarded to the BSN-Care server and maintained for the purpose of further data analysis and the mining of the patient's needs. In our proposed BSNCare+ system, two communication channels, i.e., "Sensors to Gateway" and "Gateway to BSN-Care server", are focused on because the openness of these two channels cannot guarantee that all the data transmissions on it are secure. An attacker (or hacker) may be attracted to launch malicious behaviors, such as bio-data eavesdropping on a specific person and entity counterfeiting for spoofing, on these insecure channels. This will result in huge and unpredictable losses for the hospital. In summary, the assumptions of the trust boundary of the proposed BSNCare+ system are listed below:

- The security credentials received during the registration phase are under a secure channel.
- The mobile gateway is equipped with secure storage.
- The "Sensors to Gateway" channel and "Gateway to BSN-Care Server" channel are both insecure, i.e., the transmitted data may be sniffed out.
- The BSN-Care Server is trusted and all the database accesses are safe.

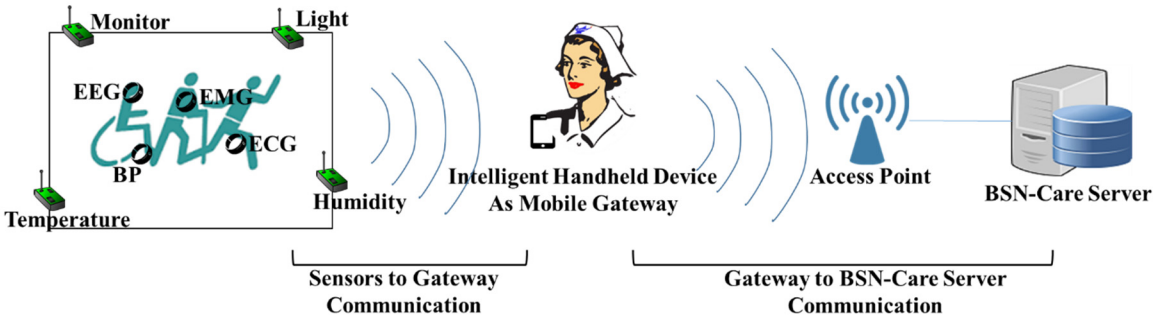

Figure 2. The proposed BSNCare+ system. EEG, electroencephalography; EMG, electromyography; BP, blood pressure; ECG, electrocardiography.

In the following, we will introduce the communication procedures of the proposed BSNCare+ system. The BSNCare+ consists of two phases: the registration phase and the authentication phase. In the registration phase, the security credentials will be agreed upon by the communication entities, i.e., body sensors, the mobile gateway and the BSN-Care server, via a secure channel. Next, an authentication phase is demonstrated for secure communication. The objectives of BSNCare+ are as follows:

- To achieve mutual authentication among communication entities.
- To guarantee anonymity and un-traceability for each body sensor in case of the disclosure of personal health status or private information.
- To resist against man-in-the-middle attack, location spoofing attack, forgery attack and replay attack.
- To achieve transaction non-repudiation property.
- To establish secure communication channels among the body sensor, the mobile gateway and the BSN-Care server to preserve data confidentiality. That is, two session keys will eventually be agreed upon by both the body sensor (or the mobile gateway) and the BSN-Care server.

Before illustrating the BSNCare+, we present the symbols, abbreviations and cryptographic functions throughout the communication process of the BSN-Care+ in Table 1.

Table 1. Notations throughout the communication process of the BSN-Care+.

Symbol	Definition
BS_i	Body sensor i
MG_j	Mobile gateway j (operated by the nurse or the doctor)
Server	The BSN-Care server
ID_i	Private identity of BS_i
ID_j	Public identity of MG_j
ID_s	Public identity of *Server*
AID_i	One-time-alias identity of BS_i
SID	A set of un-linkable shadow identities $SID = \{sid_1, sid_2, \dots\}$
K_{is}	The secret key shared between BS_i and *Server*
K_{js}	The secret key shared between MG_j and *Server*
Tr_{seq}	Track sequence number
N_{is}, N_h, N_{js}, N_i, N_j, N_{s1}, N_{s2}	Random numbers
GPS_k	The GPS information of entity k
$H(.)$	Secure one-way hash function, i.e., SHA-3
$\oplus$	Bitwise exclusive-or operation
$\|\|$	Concatenation operation

3.1. The Registration Phase

At first, the body sensor BS_i sends its identity ID_i to the BSN-Care server, i.e., *Server*, as a registration request. Upon receiving the request from BS_i, the *Server* generates a random number N_{is} and uses its identity ID_s to compute a secret key $K_{is} = H(ID_s || N_{is} || ID_i)$. Next, the *Server* calculates a set of un-linkable shadow identities $SID = \{sid_1, sid_2, \dots\}$ for BS_i, where each $sid_h \in SID$ and $sid_h = H(N_h || K_{is})$. Note that N_h is a random number used for deriving each sid_h value. In addition, a track sequence number Tr_{seq} is created for the fast identification of BS_i and to prevent replay attacks as well. Tr_{seq} will be renewed and stored on both the *Server* and the BS_i sides at each authentication session. In that case, the *Server* is able to check the freshness of an incoming request from BS_i, and to directly identify BS_i via Tr_{seq} in the backend database during the authentication session. If Tr_{seq} in the request is not maintained in the backend database, the *Server* will reject the incoming request and terminate the connection. The *Server* then asks BS_i to send a new request embedding one of the fresh shadow identities sid_h from the list of *SID*s as an anonymous identity of BS_i. Note that the used sid_h must be removed from the *SID* list at both the *Server* side and the BS_i side after the authentication session.

Finally, the *Server* issues a security credential, i.e., (ID_i, K_{is}, SID, Tr_{seq}, $H(.)$) to BS_i. Meanwhile, the *Server* maintains a tuple (ID_i, K_{is}, SID, Tr_{seq}, $H(.)$) corresponding to BS_i at the backend database. Note that $H(.)$ denotes a secure one-way hash function. On the other hand, the registration phase between the mobile gateway MG_j and the *Server* is performed in a similar way. The MG_j sends its identity ID_j to the *Server* as a registration request. Then, the *Server* computes $K_{js} = H(ID_s || N_{js} || ID_j)$ with a newly generated random number N_{js}, and shares a security credential, i.e., (ID_j, K_{js}, $H(.)$). The *Server* maintains a tuple (ID_j, K_{js}, $H(.)$) corresponding to the MG_j in the backend database.

3.2. The Authentication Phase

Under the public and insecure IoT communication architecture, an authentication procedure is needed to establish a secure communication channel for robust data exchange among BS_i, the MG_j and the *Server*. The detailed communication procedures of the authentication phase are as shown in Figure 3.

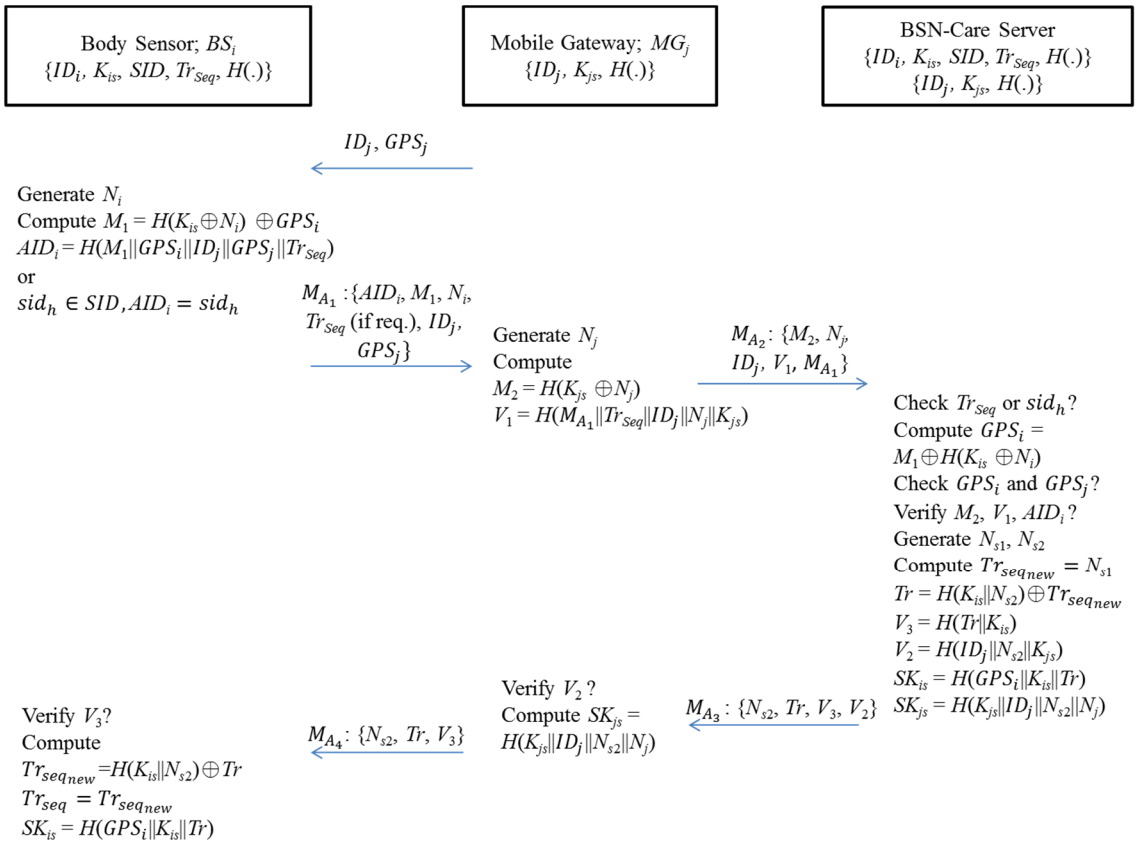

Figure 3. The authentication process of BSNCare+.

Step 1, $BS_i \rightarrow MG_j$: $M_{A_1} = \{AID_i, M_1, N_i, Tr_{seq}$ (if req.), $ID_j, GPS_j\}$

The BS_i first generates a random number N_i and calculates $M_1 = H(K_{is} \oplus N_i) \oplus GPS_i$ and $AID_i = H(M_1 || GPS_i || ID_j || GPS_j || Tr_{seq})$. Next, BS_i sends $M_{A_1} = \{AID_i, M_1, N_i, Tr_{seq}$ (if req.), $ID_j, GPS_j\}$ as an authentication request to the MG_j. Note that if the value Tr_{seq} shared between BS_i and the *Server* is out of synchronization, BS_i then chooses a fresh shadow identity sid_h from SID and, consequently, sets sid_h as AID_i. After that, BS_i sends $M_{A_1} = \{AID_i, M_1, N_i, ID_j, GPS_j\}$ as an authentication request to the MG_j.

Step 2, $MG_j \rightarrow$ *Server*: M_{A_2}: $\{M_2, N_j, ID_j, V_1, M_{A_1}\}$

Upon receiving the authentication request from BS_i, the MG_j generates a random number N_j and computes $M_2 = H(K_{js} \oplus N_j)$ and $V_1 = H(M_{A_1} || TR_{seq} || ID_j || N_j || K_{js})$. Then, the MG_j sends $M_{A_2} = \{M_2, N_j, ID_j, V_1, M_{A_1}\}$ to the *Server*.

Step 3, *Server* $\rightarrow MG_j$: $M_{A_3} = \{N_{s2}, Tr, V_3, V_2\}$

Once the *Server* obtains $M_{A_2} = \{M_2, N_j, ID_j, V_1, M_{A_1}\}$, the *Server* first checks whether the track sequence number Tr_{seq} is in the request. If Tr_{seq} is included in M_{A_2}, the *Server* performs condition (1). Otherwise, condition (2) is invoked.

- Condition (1): Check the validity of Tr_{seq}, and look for the corresponding tuple via Tr_{seq} from the backend database. If Tr_{seq} is valid, the *Server* retrieves K_{is} and derives $GPS_i = M_1 \oplus H(K_{is} \oplus N_j)$. Then, the *Server* compares the computed GPS_i with the received GPS_j for the purpose of checking if there is a location spoofing attack. If so, the *Server* stops the session. Otherwise, the *Server* verifies M_2, V_1 and AID_i via the following equations.

- ○ Are the received value M_2 and the computed value $M_2 = H(K_{js} \oplus N_j)$ equal?
- ○ Are the received value V_1 and the computed value $H(M_{A_1} || Tr_{seq} || ID_j || N_j || K_{js})$ equal?
- ○ Are the received value AID_i and the computed value $H(M_1 || GPS_i || ID_j || GPS_j || Tr_{seq})$ equal?

- Condition (2): If the *Server* cannot find any Tr_{seq} in the request, i.e., M_{A_2} and M_{A_1}, the *Server* then examines the freshness and validity of $AID_i = sid_h$. If the *Server* cannot identify the sid_h from the backend database, the server terminates the connection and requests BS_i to try with another valid shadow identity sid_h.

If one of the two examinations is passed, the *Server* then generates two random numbers N_{s1} and N_{s2}, and sets N_{s1} as the new track sequence number Tr_{seq}, i.e., $Tr_{seq_{new}} = N_{s1}$. Subsequently, the *Server* computes $Tr = H(K_{is} || N_{s2}) \oplus Tr_{seq_{new}}$, $V_3 = H(Tr || K_{is})$, $V_2 = H(ID_j || N_{s2} || K_{js})$, $SK_{is} = H(GPS_i || K_{is} || Tr)$ and $SK_{js} = H(K_{js} || ID_j || N_{s2} || N_j)$. Note that SK_{is} is the session key, which will be utilized for the next secure communication between BS_i and the *Server*, where SK_{js} is the session key agreed by the MG_j and the *Server.* After that, S sends M_{A_3}: $\{N_{s2}, Tr, V_3, V_2\}$ to the MG_j as a response.

Step 4, $MG_j \rightarrow BS_i$: $M_{A_4} = \{N_{s2}, Tr, V_3\}$

After receiving $M_{A_3} = \{N_{s2}, Tr, V_3, V_2\}$, the MG_j first computes $H(ID_j || N_j || K_{js})$ and checks if the received value V_2 is equal to the computed value $H(ID_j || N_{s2} || K_{js})$. If it holds, the mutual authentication between the MG_j and the *Server* is achieved. A session key SK_{js} is securely agreed by the MG_j and the *Server*. Finally, the MG_j forwards $M_{A_4} = \{N_{s2}, Tr, V_3\}$ to BS_i. Upon obtaining M_{A_4}, BS_i calculates $H(Tr || K_{is})$ and compares it with the received value V_3. If these two values are the same, BS_i derives $Tr_{seq_{new}} = H(K_{is} || N_{s2}) \oplus Tr$ and sets $Tr_{seq} = Tr_{seq_{new}}$, which will be used for the next new authentication session. Finally, BS_i computes a session key $SK_{is} = H(GPS_i || K_{is} || Tr)$, which is shared with the *Server*. Obviously, mutual authentication between BS_i and the *Server* is guaranteed as well.

4. Security Analysis of BSNCare+

In this section, we analyze the security robustness of the major communication procedures of BSNCare+ and, based on the analyses, we have indicated that the BSNCare+ can achieve the principal security requirements as follows:

- Claim 1: Mutual authentication can be achieved among communication entities in the BSNCare+.

The mutual authentication of the BSNCare+ system is proven through BAN logic analysis [11]. Before the analysis, we present the basic constructs and logic postulates, where the symbols P and Q range over principals, X and Y range over statements, and K ranges over long-term secrets keys. All the symbols and abbreviations are presented in Table 2.

Table 2. Basic constructs utilized in the analysis of Claim 1.

Symbol	Definition
P, Q	Principals
X, Y	Statements
K	Long-term secrets key or secrets
P believes X	The principal P believes that X is true
P sees X	Someone has sent a message containing X to P, who can read and repeat X (possibly after doing some decryption)
P said X	P has sent a message containing statement X in the current session of the protocol or before
P controls X	P has jurisdiction over X, i.e., the principal P is an authority on X and this matter should be trusted
fresh(X)	X has not been sent in a message before the current session of the protocol
$P \overset{K}{\leftrightarrow} Q$	The key K is shared between the principals P and Q
$\{X\}_K$	This symbol represents the formula X encrypted or protected under the key K

Logical postulates:

- Rule 1 (the message-meaning rules): If P believes $P \overset{K}{\leftrightarrow} Q$ and P sees $\{X\}_K$, then we postulate P believes Q said X.
- Rule 2 (the nonce-verification rule): If P believes fresh(X) and P believes Q said X, then we postulate P believes Q believes X.
- Rule 3 (the jurisdiction rule): If P believes Q controls X and P believes Q believes X, then we postulate P believes X.
- Rule 4: If P sees (X, Y) then P sees X. In addition, if P believes $P \overset{X}{\leftrightarrow} Q$ and P sees $\{X\}_K$, then P sees X.
- Rule 5: If one part of a formula is fresh, then the entire formula must also be fresh. If P believes fresh(X), then P believes fresh (X, Y).

Before we analyze the authentication process of the BSNCare+, the assumptions are given.

- Assumption 1: BS_i and *Server* believe $BS_i \overset{ID_i,\ K_{is},\ SID,\ Tr_{seq}}{\leftrightarrow} Server$
- Assumption 2: MG_j and *Server* believe $MG_j \overset{ID_j,\ K_{js}}{\leftrightarrow} Server$
- Assumption 3: *Server* believes fresh (N_i, N_j)
- Assumption 4: BS_i and MG_j believe fresh (N_{s1}, N_{s2})
- Assumption 5: *Server* believes BS_i controls (N_i)
- Assumption 6: *Server* believes MG_j controls (N_j)
- Assumption 7: BS_i and MG_j believe *Server* controls (N_{s1}, N_{s2})

The concrete realization of the authentication procedures of the BSNCare+ is presented as follows. Please refer to Table 1 for the definition of each symbol and Figure 1 for the detailed communication procedures of the BSNCare+.

- Step 1, $BS_i \rightarrow MG_j$: $M_{A_1} = \{AID_i, M_1, N_i, Tr_{seq}$ (if req.), ID_j, $GPS_j\}$, where $M_1 = H(K_{is} \oplus N_i) \oplus GPS_i$ and $AID_i = H(M_1 \,||\, GPS_i \,||\, ID_j \,||\, GPS_j \,||\, Tr_{seq})$.
- Step 2, $MG_j \rightarrow Server$: $M_{A_2} = \{M_2, N_j, ID_j, V_1, M_{A_1}\}$, where $M_2 = H(K_{js} \oplus N_j)$ and $V_1 = H(M_{A_1} \,||\, Tr_{seq} \,||\, ID_j \,||\, N_j \,||\, K_{js})$ and $M_{A_1} = \{AID_i, M_1, N_i, Tr_{seq}$ (if req.), ID_j, $GPS_j\}$.
- Step 3, $Server \rightarrow MG_j$: $M_{A_3} = \{N_{s2}, Tr, V_3, V_2\}$, where $Tr = H(K_{is} \,||\, N_{s2}) \oplus Tr_{seq_{new}}$, $V_3 = H(Tr \,||\, K_{is})$, $V_2 = H(ID_j \,||\, N_{s2} \,||\, K_{js})$.
- Step 4, $MG_j \rightarrow BS_i$: $M_{A_4} = \{N_{s2}, Tr, V_3\}$, where $Tr = H(K_{is} \,||\, N_{s2}) \oplus Tr_{seq_{new}}$ and $V_3 = H(Tr \,||\, K_{is})$.

The formal analysis of mutual authentication of the BSNCare+ is demonstrated in the following.

- (1) MG_j sees $M_{A_3} = \{N_{s2}, Tr, V_3, V_2\}$: Based on Step 3, it is obvious that MG_j has received and seen $M_{A_3} = \{N_{s2}, Tr, V_3, V_2\}$.
- (2) MG_j believes $MG_j \overset{ID_j,\ K_{js}}{\leftrightarrow} Server$: With the assumption 2, MG_j believes that it actually shares ID_j and K_{js} with the *Server*.
- (3) MG_j believes *Server* said $\{V_2\}$: Based on the above two results, i.e., (1) and (2), we can derive that MG_j believes *Server* said $\{V_2\}$ via Rule 1 (the message-meaning rules).
- (4) MG_j believes fresh (N_j): As the N_j is issued by the MG_j itself, MG_j can tell the freshness of N_j, and believes fresh (N_j) if it is.
- (5) MG_j believes *Server* believes $\{V_2\}$: Based on the results (3) and (4), the fact of MG_j believing that the *Server* believes $\{V_2\}$ is confirmed because Rule 2 (the nonce-verification rule) support this claim.
- (6) MG_j believes *Server* controls $\{N_{s2}\}$: Based on the assumption 7, MG_j believes *Server* actually controls the random number N_{s2}.

- (7) MG_j believes $\{V_2\}$: With the results (5) & (6) and Rule 3 (the jurisdiction rule), we can derive that MG_j believes $\{V_2\}$.
- (8) BS_i sees $M_{A_4} = \{N_{s2}, Tr, V_3\}$): With Step 4, BS_i has actually seen $M_{A_4} = \{N_{s2}, Tr, V_3\}$.
- (9) BS_i believes $BS_i \overset{ID_i,\ K_{is},\ SID,\ Tr_{seq}}{\leftrightarrow}$ *Server*: Based on the assumption 1, BS_i believes that it actually shares $ID_i,\ K_{is},\ SID$ and Tr_{seq} with the *Server*.
- (10) BS_i believes *Server* said $\{Tr,V_3\}$: Based on the above two results (8) and (9), we can derive that BS_i believes *Server* said $\{Tr, V_3\}$ via Rule 1 (the message-meaning rules).
- (11) BS_i believes fresh (N_{s2}): Based on the assumption 4, BS_i believes the freshness of N_{s2}.
- (12) BS_i believes *Server* believes $\{Tr,\ V_3\}$: Based on the results (10) and (11), it is guaranteed that BS_i believes *Server* believes $\{Tr,\ V_3\}$ since Rule 2 (the nonce-verification rule).
- (13) BS_i believes *Server* controls $\{N_{s2}\}$: Based on the assumption 7, BS_i believes *Server* actually controls the random number N_{s2}.
- (14) BS_i believes $\{Tr,\ V_3\}$: With the results (12) & (13) and Rule 3 (the jurisdiction rule), we can derive that BS_i believes $\{Tr,\ V_3\}$.

The final results are as follows:

MG_j believes *Server* believes $\{V_2\}$ (From (5))
MG_j believes $\{V_2\}$ (From (7))
BS_i believes *Server* believes $\{Tr,\ V_3\}$ (From (12))
BS_i believes $\{Tr,\ V_3\}$ (From (14))

With the four results (5), (7), (12) and (14), and the assumption of the trustworthiness of *Server*, both BS_i and the MG_j can be authenticated by each other via *Server*.

- Claim 2: The anonymity and un-traceability of BS_i can be guaranteed.

In the normal communication processes of the BSNCare+, we adopt three random numbers N_i, N_j and N_{s2} to randomize the transmitted messages, such as AID_i, M_1, M_2, V_1, Tr, V_2 and V_3, where some of them are involved with BS_i. They are AID_i, M_1, N_i, TR_{seq}, Tr and V_3. First, in the BSNCare+, TR_{seq} is specifically utilized to be the one-time-use token for fast identification of BS_i because this value, i.e., TR_{seq} contains no information revelant to BS_i and will be updated after each successful authentication session. The value TR_{seq} does not reveal any information about BS_i. Second, an anonymous identity AID_i (or sometimes, a one-time use and meaningless value sid_h will be used) to be exchanged between BS_i and the *Server*, and will be examined by the *Server*. No one can trace and identify BS_i except the *Server*. Third, from the expression of equations of $M_1 = H(K_{is} \oplus N_i) \oplus GPS_i$, $Tr = H(K_{is} \,||\, N_{s2}) \oplus Tr_{seq_{new}}$ and $V_3 = H(Tr \,||\, K_{is})$, it is obvious that these three values are randomized by the two random numbers N_i and N_{s2}, which cannot be re-used from session to session. Based on the above arguments, we claim that the anonymity and un-traceability for BS_i can be guaranteed.

- Claim 3: Secure channels among BS_i, the MG_j and the *Server* are successfully established and data confidentiality can be achieved.

As mentioned before, a simple authentication and login activity without session key agreement is not enough to guarantee any kind of security. One of the most important security requirements is to securely establish two session keys in which one session key is agreed by BS_i and the *Server*, and the other one is established by the MG_j and the *Server*. In the BSNCare+, two session keys, i.e., $SK_{is} = H(GPS_i \,||\, K_{is} \,||\, Tr)$ and $SK_{js} = H(K_{js} \,||\, ID_j \,||\, N_{s2} \,||\, N_j)$, will eventually be generated and exploited to protect the data transmissions among among BS_i, the MG_j and the *Server*. In that case, we claim that the secure channels among BS_i, the MG_j and the *Server* are provided by the BSNCare+. In addition, during each normal session of the BSNCare+, all the transmitted messages are either well-protected via the robust one-way hash function, i.e., SHA-3 (512 bits) or meaningless random

sequences, which do not reveal any useful information regarding the communication entities or the BSNCare+ itself. Furthermore, two high-entropy secrets, i.e., K_{is} and K_{js}, are chosen by the *Server* for the purpose of protecting the communication during the authentication phase of BSNCare+. Without knowing these two secrets, it is theoretically difficult for attackers to break the SHA-3 hash function and retrieve any useful information from transmitted cipher texts owing to the irreversibility of the one-way hash function. Data confidentiality is thus guaranteed.

- Claim 4: Resistance against man-in-the-middle attack, location spoofing attack, forgery attack and replay attack is provided.

Attackers may intend to deceive the legal communication entities, such as BS_i, the MG_j and the *Server*, via fake messages. How to efficiently identify such counterfeit messages and eliminate new potential threats in an IoT-based healthcare environment has thus become increasingly important. Messages forged by the malicious attackers may exist in any kind of expressions and new cheating tricks may be launched through the heterogeneous network architectures of IoT. Hence, an examination scheme for identifying the counterfeit messages and accordingly for preventing malicious attacks is indispensable for an authentication scheme. In the BSNCare+, without the knowledge of K_{is} and K_{js}, it is difficult for the attacker to counterfeit legitimate messages such as AID_i, M_1, M_2, V_1, Tr, V_2 and V_3, and to launch a forgery attack. Note that these two secrets are chosen by the *Server* and have a high-entropy to resist against brute-force attacks. Even if the attacker sends a valid-but-used message, eavesdropped from the previous authentication session, to a victim party, the verification of these previously-used messages will fail. This is because the random numbers, i.e., N_i, N_j and N_{s2}, have a one-time-use for each session. That is, the examination of the freshness of these used random numbers will make the verification invalid. Therefore, the resistance to replay attack is embedded in the BSNCare+. Next, in the BSNCare+, we embed the GPS information of BS_i and MG_j into M_1 and M_{A_1}, respectively. For the un-traceability for BS_i, the GPS information of BS_i is well-protected in M_1. Nobody except the *Server* can perform a un-traceability attack on BS_i due to the unknown K_{is}. On the *Server* side, a verification process for examining location spoofing attacks is presented. The resistance to location spoofing attacks is naturally embedded into the BSNCare+. Finally, we analyze the robustness against man-in-the-middle attacks. In the BSNCare+, an attacker may interrupt authentication sessions and fool the legal communicating entities that he/she is the other legitimate side via eavesdropped (or counterfeited) messages. To prevent such a phenomenon, we embed all the entities' identities into the protocol messages for mutual authentication among communication entities; for example, $K_{is} = H(ID_s||N_{is}||ID_i)$, $K_{js} = H(ID_s||N_{js}||ID_j)$, $M_1 = H(K_{is} \oplus N_i) \oplus GPS_i$, $AID_i = H(M_1||GPS_i||ID_j||GPS_j||TR_{seq})$, $M_2 = H(K_{js} \oplus N_j)$, $V_1 = H(M_{A_1}||TR_{seq}||ID_j||N_j||K_{js})$, $Tr = H(K_{is}||N_{s2}) \oplus Tr_{seq_{new}}$, $V_3 = H(Tr||K_{is})$ and $V_2 = H(ID_j||N_{s2}||K_{js})$. It is obvious that all the messages contain the entities' identities actually communicating in the authentication session. Accordingly, to counterfeit a forged message containing other invalid entities' identities is difficult because of the irreversibility of the one-way hash function. On the other hand, the previously used message cannot be reused (or modified) under the protection of the one-way hash function with one-time use random numbers. The resistance against man-in-the-middle attack is actually provided.

- Claim 5: The non-repudiation transaction property is provided.

In an IoT-wide universe, a mechanism capable of proving a group of mobile objects (or mobile gateway and sensors) existing at the same place and at the same time is important for preventing possible fraud event. In our proposed BSNCare+ system, the identities and the GPS information of BS_i and MG_j is embedded into the authentication messages, i.e., M_{A_1} and M_{A_2}, to demonstrate the involved entities and the location of current transaction session. With a legal timestamp provided by the *Sever*, M_{A_2}: $\{M_2, N_j, ID_j, V_1, M_{A_1}\}$ can be as a evidence for current transaction involving BS_i and MG_j once the verification of M_{A_2} is confirmed. Hence, we can claim that the BSNCare+ system exhibits the non-repudiation transaction property.

5. Performance Evaluation of BSNCare+

To evaluate the performance of the BSNCare+, we implement a demo system of the BSNCare+ as a proof of concept of our proposed idea. The implementation environment is as shown in Table 3, where the Raspberry PI 2 platform is simulated as the operating entities during an authentication session of the BSNCare+. The Raspberry PI is a credit card-sized single-board computer, which has been recognized as one of the most popular key platforms for IoT-oriented technique development. It offers a complete Linux server in a tiny platform at a very low cost and thus is suitable to evaluate the performance of any kind of IoT-oriented protocols. Therefore, in the performance evaluation we adopt the Raspberry PI 2 platform to perform the major authentication process of the BSNCare+ for the purpose of testing the practicability of the BSNCare+ in terms of its efficiency and feasibility. In addition, in order to pursue the balance of the protocol efficiency, security robustness and system scalability, a secure one-way hash function, i.e., SHA-3 (512 bits) [12,13], a bitwise exclusive-or operation and a random number generator are implemented in the demo system. Note that in the system implementation, the values ID_i, ID_s, ID_j, N_{is}, N_h, N_{js}, sid_h, Tr_{seq}, N_i, N_j, GPS_i, GPS_j, N_{s1} and N_{s2} are all set to 96-bits for appropriate security density. Each time the SID contains 100 sid_h values. All the experiments are programmed via Oracle Java 8 and Eclipse 3.8, and are realized with the Bouncy Castle Crypto APIs [13].

Table 3. Implementation Environment.

Environment	Description
Raspberry PI 2	Broadcom BCM2836 @ 1GHz Quad-Core ARM Cortex-A7 Architecture. 1GB DDR2 RAM SanDisk 16GB Class 10 SD Card
Operating System	Raspbian2016/03
Programming Language	Oracle Java 8 ARM
Programming IDE	Eclipse 3.8
Crypto API	The Bouncy Castle Crypto APIs [13]

Table 4 demonstrates the computation cost required in our proposed BSNCare+ in terms of the execution time of the major communication scheme. In the registration phase, we need $(2 + k)$ times of 96-bit random number generator operations and $(2 + k)$ times the SHA-3 (512 bits) hash function. In detail, twice the SHA-3 hash function with 288-bits input sequence, i.e., $K_{is} = H(ID_s || N_{is} || ID_i)$, k time of SHA-3 hash function with 608-bits input sequence, i.e., $sid_h = H(N_h || K_{is})$, are evaluated in the system implementation. Note that k is the size of SID and we set k as 100 to balance between the security and efficiency of the BSNCare+. In total, we require 47.34 ms to complete the registration phase, which generates 102 random numbers with length of 96-bits, and executes 102 times of SHA-3 hash functions, where 7.14 ms is for generating 102 random numbers and 40.2 ms is for executing 102 times the SHA-3 hash functions. Based on our implementation results, the performance bottleneck occurs at the execution of the SHA-3 hash function with a 608-bit input sequence.

Next, we examine the execution cost of performing the authentication phase involved with Tr_{seq}. In brief, 4 times the random number generations, 8 times the bitwise exclusive-or operations and 18 times the SHA-3 hash function are needed to complete the authentication process with an execution time of 9.634 ms (Table 3). In addition, in Table 5 we present the details of the implementation of the authentication phase with Tr_{seq}. We find that the computation cost is dominated by the SHA-3 hash function as the execution time of the random number generation and the exclusive-or operation are comparatively slight when compared to the SHA-3 hash function. We see two interesting results. First, the ratio of "the execution time of performing all required SHA-3 functions" to "the total computation time required for executing the authentication phase" is 94.95%. Second, the execution time of performing twice the SHA-3 function with a 2208-bit input sequence accounts for 35.81%

of the total computation cost. Based on these two observations, we find that although the SHA-3 hash function dominates the efficiency of the BSNCare+, it has space for efficiency improvement if the protocol message during the authentication phase can be more-carefully designed to fit the characteristics of the SHA-3 hash function. For example, we find that the efficiency of the SHA-3 hash function will be downgraded to a lower level once the input sequence exceeds the multiple of 576 bits, which is one of the defaulted block sizes of the SHA-3 hash function. In that case, an interesting result, i.e., the SHA-3 hash function may be more suitable for communication protocols with short messages, is observed. Finally, the computation cost for implementing the authentication phase without Tr_{seq} is investigated in which, in total, an execution time of 7.540 ms is needed for performing 4 times the random number generations, 8 times the bitwise exclusive-or operations and 16 times teh SHA-3 hash function. Similarly, Table 6 presents the detailed implementation of the authentication phase without Tr_{seq}. The same trend is obtained as that observed in Table 5.

Table 4. Execution Time of the BSNCare+.

Phase	Computation Cost	Execution Time (ms)
Registration Phase	(2+k) RN + (2+k) H [1]	47.34 ms (with k = 100)
Authentication Phase with Tr_{seq}	BS_i: 1RN + 3XOR + 5H	1.969 ms (20.44%)
	MG_j: 1RN + 1XOR + 4H	2.848 ms (29.56%)
	Server: 2RN + 4XOR + 9H	4.817 ms (50.00%)
	Total: 4RN + 8XOR + 18H	9.634 ms
Authentication Phase without Tr_{seq}	BS_i: 1RN + 3XOR + 4H	1.451 ms (19.24%)
	MG_j: 1RN + 1XOR + 4H	2.319 ms (30.76%)
	Server: 2RN + 4XOR + 8H	3.770 ms (50.00%)
	Total: 4RN + 8XOR + 16H	7.540 ms

[1] k is the size of *SID*, which contains $ksid_h$ values, in the implementation, k = 100. RN means a random number. XOR means a bitwise exclusive-or operation. H means the one-way hash function, i.e., SHA-3 (512 bits).

Table 5. The details of the implementation of the authentication phase with Tr_{seq}.

Operations in the System Implementation	Execution Time (ms)
4 times of 96-bit random number generations, i.e., N_i, N_j, N_{s1} and N_{s2}	0.2783
8 times of bitwise exclusive-or operations	0.2078
4 times of SHA-3 function with 512-bit input sequence, i.e., M_1 and M_2	0.02
2 times of SHA-3 function with 608-bit input sequence, i.e., Tr	0.802
2 times of SHA-3 function with 704-bit input sequence, i.e., V_2	1.04
2 times of SHA-3 function with 800-bit input sequence, i.e., SK_{js}	1.006
2 times of SHA-3 function with 896-bit input sequence, i.e., AID_i	1.038
2 times of SHA-3 function with 1024-bit input sequence, i.e., V_3	0.866
2 times of SHA-3 function with 1120-bit input sequence, i.e., SK_{is}	0.928
2 times of SHA-3 function with 2208-bit input sequence, i.e., V_1	3.45

Table 6. The details of the implementation of the authentication phase without Tr_{seq}.

Operations in the System Implementation	Execution Time (ms)
4 times of 96-bit random number generations, i.e., N_i, N_j, N_{s1} and N_{s2}	0.2783
8 times of bitwise exclusive-or operations	0.2078
4 times of SHA-3 function with 512-bit input sequence, i.e., M_1 and M_2	0.02
2 times of SHA-3 function with 608-bit input sequence, i.e., Tr	0.802
2 times of SHA-3 function with 704-bit input sequence, i.e., V_2	1.04
2 times of SHA-3 function with 800-bit input sequence, i.e., SK_{js}	1.006
2 times of SHA-3 function with 1024-bit input sequence, i.e., V_3	0.866
2 times of SHA-3 function with 1120-bit input sequence, i.e., SK_{is}	0.928
2 times of SHA-3 function with 1600-bit input sequence, i.e., V_1	2.392

Performance evaluation is one of the most indispensable issues while designing a robust and efficient communication scheme. In general, the evaluation is able to reflect the practicability and feasibility of deploying the proposed mechanism into the real world. In the above sections, we have demonstrated the practicability and feasibility of our proposed system, i.e., the BSNCare+, through a common IoT-oriented developing platform, i.e. the Raspberry PI 2 platform. To further investigate the advantage of the BSNCare+, we compare it with BSN-Care [9], USM-IoT [10] and scheme in [14] in terms of performance efficiency. The comparisons among BSNCare+ and the other two schemes are listed in Table 7. The evaluation metrics are the one-way hash function (HF), random number generation (RN) and bitwise exclusive-or operation (XOR). Although our proposed BSNCare+ scheme additionally requires at most 3 times the random number generation, at most 2 times the bitwise exclusive-or operation and at most 7 times the one-way hash function, our scheme can achieve the same order of computation complexity as these three schemes do. The efficiency is guaranteed to have better security robustness than BSN-Care [9] and USM-IoT [10].

Table 7. Performance comparison among BSN-Care [9], USM-IoT [10], scheme in [14] and BSNCare+ [1].

Method	Underlying Architecture	Phase	Computation Cost
BSN-Care [9]	Client-Server	Registration Phase Authentication Phase Total	(1 + *k*) RN + (1 + *k*) H 2RN + 8XOR + 12H (3 + *k*) RN + 8XOR + (13 + *k*) H
USM-IoT [10]	Client-Client-Server	Registration Phase Authentication Phase Total	(1 + *k*) RN + (1 + *k*) H* 3RN + 6XOR + 12H (4 + *k*) RN + 6XOR + (13 + *k*) H
Scheme in [14]	Client-Client-Server	Registration Phase Authentication Phase Total	(2k + 2) RN + (3k + 7) XOR + (2k + 8) H 4RN + 12XOR + 24H (6 + 2k) RN + (3k + 19) XOR + (32 + 2k) H
BSNCare+	Client-Client-Server	Registration Phase Authentication Phase Total	(2 + *k*) RN + (2 + *k*) H 4RN + 8XOR + 18H (6 + *k*) RN + 8XOR + (20 + *k*) H

[1] *k* is the size of *SID*, which contains $ksid_h$ values, in the implementation, *k* = 100. RN means a random number not including the user identity and chosen password. XOR means a bitwise exclusive-or operation. H means the one-way hash function, i.e., SHA-3 (512 bits). Note that the generation of the secret key, i.e., K_{ch}, shared by the cluster head and the H IoTS is not explained in [10]. We skip the corresponding generation cost here.

On the other hand, the BSN-Care [9], USM-IoT [10], scheme in [14] and BSNCare+ utilize an elegant re-synchronization mechanism consisting of a set of un-linkable shadow identities $SID = \{sid_1, sid_2, \dots\}$ to conquer the Denial of Services (DoS) attack. Note that from a sensor point of view, the definition of *DoS* attack in the context of IoT-based authentication is that the secrets held by the sensor and the *Server* are out of synchronization due to abnormal process conditions such as malicious interruption or operation interference [15,16]. Consequently, in the next authentication session this sensor can no longer be successfully validated by the *Server*. The pre-shared un-linkable shadow identities *SID* can be adopted as a fresh and legitimate secret when an abnormal session occurs, and the *DoS* attack can thus be prevented. In the proposed BSNCare+ system, we require a cost of performing *k* times random numbers and hash functions for the re-synchronization mechanism. The cost is similar to that in BSN-Care [9] and USM-IoT [10], and is lower than scheme in [14].

In our implementation, we have realized our proposed BSNCare+ system on a common IoT-based testbed, i.e., raspberry PI 2 platform, as a proof of concept. What we focus on is the healthcare relevant scenario identified in Sections 3.1 and 3.2. In the applied scenarios, intelligent body sensors are deployed in the field and on the patient, respectively, to support the nurse/doctor in activities of patient care, rehabilitation and even healthcare. That is, the nurse/doctor can utilize his/her mobile gateway for real-time data collection and provide better-quality healthcare services to the patient with further data analysis and the mining of the patient's needs. It is obvious that the BSNCare+ system locates in a very-reasonable computation cost level because at most 9.634 ms is required in executing a

normal protocol session of the BSNCare+ system on a common IoT-based testbed. We thus believe the practicality of the BSNCare+ system as it guarantees better security density with a user-acceptable computation cost.

6. Conclusions

With the rapid growth of the IoT application development and its potential, there have been subsequent changes in administration and operation models of medical (or hospital) organizations across the world. Numerous intelligent devices are connected and thus form a pervasive IoT-based network architecture, which brings new security challenges. While the proactive solutions, such as entity authentication, have performed well, and security enhancement has promptly been adopted for IoT-oriented networks, the reactive security mechanism, such as malicious event detection and prevention [17,18], still play an important role in terms of the complete-and-integrated security perspective. The security is not just protection from a single point view. Complementary security techniques, such as firewalls, intrusion detection and prevention, entity authentication and access control, must be thoroughly considered and integrated to provide a complete and robust security for IoT-oriented networks. In the future, a complete framework integrating various types of security mechanisms with specific user (or system) demands from IoT-based healthcare system in the real world will be one of the most promising research directions.

To efficiently protect an IoT-based service system is a particular challenge owing to the tradeoff among network heterogeneity, system security and computation efficiency. One of the most promising solutions is to implement an efficient authentication scheme for IoT-oriented architecture. In this paper, we present a robust IoT-based healthcare application system, called BSNCare+, which is based on improvements on the techniques presented in BSN-Care and USM-IoT. With formal analysis and performance evaluation, the security robustness and computation efficiency of the BSNCare+ can be guaranteed. The execution time of at most 9.634 ms is derived for performing the major communication process of the BSNCare+ on a common IoT-oriented development platform, i.e., Raspberry PI 2 platform. The computation cost is reasonable and user-acceptable. Therefore, we believe that our proposed healthcare system, i.e. BSNCare+, is practical and feasible for IoT-based BSN network architecture.

Acknowledgments: This work was supported by the Academia Sinica, the Taiwan Information Security Center (TWISC) and the Ministry of Science and Technology, Taiwan, under the grants numbered MOST 105-2221-E-259-014-MY3, MOST 105-2221-E-011-070-MY3, MOST 105-2923-E-182-001-MY3, MOST 104-2218-E-001-002 and MOST 105-2218-E-001-001.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Gibbs, S. Hackers Can Hijack Wi-Fi Hello Barbie to Spy on Your Children. 2015. Available online: http://www.theguardian.com/technology/2015/nov/26/hackers-can-hijack-wi-fi-hello-barbie-to-spy-on-your-children (accessed on 19 September 2016).
2. Yao, X.; Han, X.; Du, X.; Zhou, X. A lightweight multicast authentication mechanism for small scale IoT applications. *IEEE Sens. J.* **2013**, *13*, 3696–3701. [CrossRef]
3. Kawamoto, Y.; Nishiyama, H.; Kato, N.; Shimizu, Y.; Takahara, A.; Jiang, T. Effectively collecting data for the location-based authentication in the Internet of Things. *IEEE Sens. J.* **2015**. [CrossRef]
4. Hernandez-Ramos, J.L.; Pawlowski, M.P.; Jara, A.J.; Skarmeta, A.F.; Ladid, L. Toward a lightweight authentication and authorization framework for smart objects. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 690–702. [CrossRef]
5. Tennina, S.; Di Renzo, M.; Kartsakli, E.; Graziosi, F.; Lalos, A.S.; Antonopoulos, A.; Mekikis, P.V.; Alonso, L. WSN4QoL: A WSN-oriented healthcare system architecture. *Int. J. Distrib. Sens. Netw.* **2014**, *2014*, 503417. [CrossRef]

6. Kartsakli, E.; Antonopoulos, A.; Alonso, L.; Verikoukis, C. A cloud-assisted random linear network coding medium access control protocol for healthcare applications. *Sensors* **2014**, *14*, 4806–4830. [CrossRef] [PubMed]
7. Kartsakli, E.; Antonopoulos, A.; Lalos, A.S.; Tennina, S.; Di Renzo, M.; Alonso, L.; Verikoukis, C. Reliable MAC design for ambient assisted living: Moving the coordination to the cloud. *IEEE Commun. Mag.* **2015**, *53*, 78–86. [CrossRef]
8. Kartsakli, E.; Lalos, A.S.; Antonopoulos, A.; Tennina, S.; Renzo, M.D.; Alonso, L.; Verikoukis, C. A survey on M2M systems for mHealth: A wireless communications perspective. *Sensors* **2014**, *14*, 18009–18052. [CrossRef] [PubMed]
9. Gope, P.; Hwang, T. BSN-Care: A secure IoT-based modern healthcare system using body sensor network. *IEEE Sens. J.* **2016**, *16*, 1368–1376. [CrossRef]
10. Gope, P.; Hwang, T. Untraceable sensor movement in distributed IoT infrastructure. *IEEE Sens. J.* **2015**, *15*, 5340–5348. [CrossRef]
11. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
12. Dworkin, M.J. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, NIST FIPS-202. August 2015. Available online: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf (accessed on 8 December 2016).
13. The Bouncy Castle Crypto APIs. 2016. Available online: https://www.bouncycastle.org/ (accessed on 8 December 2016).
14. Gope, P.; Hwang, T. A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. *IEEE Trans. Ind. Electron.* **2016**. [CrossRef]
15. Gope, P.; Lee, J.; Quek, T.Q.S. Resilience of DoS attack in designing anonymous user authentication protocol for wireless sensor networks. *IEEE Sens. J.* **2016**. [CrossRef]
16. Wang, C.H.; Lin, C.Y. An efficient delegation-based roaming payment, protocol against denial of service attacks. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 September 2011; pp. 4136–4140.
17. Althunibat, S.; Antonopoulos, A.; Kartsakli, E.; Granelli, F.; Verikoukis, C. Countering intelligent dependent malicious nodes in target detection wireless sensor networks. *IEEE Sens. J.* **2016**, *16*, 8627. [CrossRef]
18. Antonopoulos, A.; Verikoukis, C. Misbehavior detection in the Internet of Things: A network-coding-aware statistical approach. In Proceedings of the IEEE INDIN 2016, Poitiers, France, 18–21 July 2016.