Efficient Authentication Protocol for Continuous Monitoring in Medical Sensor Networks

BIDI YING\textsuperscript{1,2}, NADA RADWAN MOHSEN\textsuperscript{2}, AND AMIYA NAYAK\textsuperscript{2}

\textsuperscript{1}School of Information & Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
\textsuperscript{2}School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa K1N 6N5, Canada

CORRESPONDING AUTHOR: BIDI YING. (e-mail: yingbid@gmail.zjgsu.edu.cn)

ABSTRACT Currently, continuous monitoring on patients with the help of small devices (or sensors), is easy for doctors/nurses to check patients. Due to privacy issues, data collected from devices should be protected. Thus, a lightweight mutual authentication and key agreement protocol is required among doctors/nurses, trusted servers, sensors and patients. In this paper, we provide a secure protocol which could support continuous monitoring on patients. Firstly, user’s biometrics will be used to verify users by means of continuous monitoring of physiological data (e.g., ECG signals) in which verification of the patient identity. This could prevent device theft attacks. In addition, dynamic identity is taken to provide user anonymity and mitigate against user traceability. Later, we provide informal and formal security analysis to prove that our protocol can establish a session key between the user and sensor after successfully mutually authentication. Performance analysis proved our scheme to be competitive in comparison to existing schemes relative to the added security benefits it provides.

INDEX TERMS Biometrics, wearable medical sensor networks, continuous monitoring.

I. INTRODUCTION
With the growth in the number of patients suffering chronic and cardiovascular diseases in advanced countries and the general aging of the population, demand for medical care and patient remote monitoring or telehealth rose [1]. Wearable medical sensor networks (WMSN) which includes sensors attached to the patient’s body such as ECG electrodes, pulse oximeters, temperature or blood pressure sensors is provided in E-health care systems due to the growth in the number of patients and with the development of new wireless technologies [2]. In this system, doctors/nurses could monitor patients’ heartbeat rate, temperature, blood pressure or blood oxygen level at any time without the need of being within physical proximity.

Data related to patients is highly sensitive, and needs to be protected. However, due to insecure communication channels, data collected from patients is easy to be listened, modified, and thus results in an incorrect diagnosis [3]. Therefore, strict precautions must be taken to prevent illegal access to the patient’s private data through a strong user authentication mechanism. Thus, all users need to be authenticated before accessing the sensitive data. In addition, data needs to be encrypted by a session key among communication between doctors/nurses provider and patients. Such a protocol should be designed in wearable medical sensor networks to provide a balance between security, privacy and computational cost [1].

A. RELATED WORK
Recently, long-term secret keys and smart cards were introduced to supplement passwords and create two-factor user authentication schemes [4], in order to improve security. However, due to resource limitations, long-term keys are usually stored into the gateway [5]. This protocol design restriction resulted in the proposal of a large number of two-factor authentication and key agreement protocols using the smart card and password.
In 2009, Das et al. [6] designed a two-factor authentication scheme to authenticate user and sensor nodes using smart card and password; however, the scheme did not establish a session key at all. To improve on this scheme, Vaidya et al. [7] proposed a two-factor mutual user authentication scheme with key agreement for WSNs. In 2014, Kim et al. [8] showed that [7] was susceptible to user-impersonation and gateway node bypassing so introduced a two-factor user authentication and key agreement protocol for WSN claiming to resist both attacks. Chang et al. [4] was shown to be vulnerable to offline password guessing attacks and to lack perfect forward secrecy by Park et al. [9]. Once an attacker is successful at guessing the correct password, he can easily perform an impersonation attack, stolen verifier attack and lost smart card attack [9].

To address some weaknesses of two-factor authentication protocols including their failure to resist offline password guessing attacks and inability to update user passwords, authentication schemes introduced biometric keys as a third factor. Biometrics has several advantages in that they cannot be lost or forgotten and are difficult to copy, forge or break [10]. These properties make biometric-based schemes more reliable and more secure than conventional schemes [10].

In 2015, A.K. Das et al. [11] introduced a biometric based authentication scheme resisting well known security threats in WSNs including stolen smart card attack, impersonation attack, offline-password guessing attack and man-in-the-middle attack. Maurya et al. [12] found that the scheme is in fact susceptible to stolen smart card attack, [13] proved its susceptibility to impersonation attack, and [14] showed it allowed user-forgery attack and offline-password guessing attacks. In 2016, Choi et al. [15] proposed another biometric based scheme addressing the issues of lack of accuracy of biometric recognition, user verification difficulty. However, this scheme is vulnerable to user impersonation attack and known plain-text attack in addition to not providing user anonymity.

A three-factor user authentication and key agreement protocol for WSN is proposed by Park et al. [9] with the help of the elliptic curve cryptosystem and fuzzy extraction. Later, Wang et al. [1] proved that Park et al. [9] lacks resistance to offline password guessing due to its incorrect application of ECC in the protocol design. This is because all the parameters in the verification parameter can be computed using static knowledge readily accessible to an attacker sniffing the public channel with access to the smart card and biometrics. Moon et al. [16] improved on Park et al.’s scheme and claimed to address its weaknesses but was found to be susceptible to impersonation attack by Maurya et al. [12]. In 2017, an another three-factor user authentication scheme which uses smart card, password and biometrics protected through utilization of a bio-hash function is proposed [17]. Wang et al. [1] also proved that Jung et al. [17] is not resistant to impersonation and offline password guessing attacks in addition to not providing forward secrecy and user anonymity. Besides, utilizing biometrics are also introduced by [18] and [19], however, they are lack of off-line password update.

The above schemes belong to one-time (static) authentication which means that authentication is only invoked at the beginning of a communication session. However, if an attacker is able to gain access to the system, he/she can continue to use it for a long period of time without the need to re-authenticate [20]. Thus, static authentication does not protect against session hijacking. Due to the above issue, continuous authentication has been introduced as a supplemental means of automatically verifying the legitimacy of a user with the help of prior knowledge of their motion state collected from mobile devices or wearable motion sensors [21]–[23]. For example, using behavioral or physiological data including electroencephalogram (EEG), electrocardiogram (ECG) and photoplethysmography (PPG) can implement continuous authentication.

B. CONTRIBUTION

A lightweight biometric based authentication protocol is proposed in this paper. This protocol will utilize ECC and dynamic identity with user’s biometrics to verify users. Fuzzy extraction which consists of two randomized operations is used to protect biometrics. The principle of fuzzy extraction is out of the paper, details can be seen in [12].

To provide a continuous monitoring, our scheme mutually authenticates the doctor/nurse and trusted server through three-factor authentication, and generates a session key between doctor/nurse and sensor. Similarly to [21]–[23], we take patient’s ECG signals to achieve a continuous patient’s identity verification occurred in the trusted server. Note that, to reduce overhead in sensors, we do not perform continuous authentication on the sensor; rather, we chose to append our static sensor authentication with continuous verification.

The main contributions are:

1) End-to-end authentication: for the first time, providing a complete end-to-end scheme that can be deployed in a real-time environment across the doctor/nurse, trusted server, sensor and patient while utilizing biometrics on both ends to enhance security. Compared to existing scheme [1], [9], it could reduce lots of communication/computational overhead.

2) Continuous monitoring: our protocol could provide continuous monitoring for patients by verifying their physiological data.

3) Anonymity and resist sensor theft attack: our scheme could resist to sensor theft attack, and protect patient’s identity. Also, Dynamic identities are introduced to provide the anonymity and untraceability of mutual authentications as these identities cannot be retrieved by adversaries without knowing secret random numbers and update in each round authentication process.

4) Security: formally validating that our protocol establishes a shared session key and achieves mutual authentication using BAN logic, and simulation results based AVISPA also prove that our protocol can resist replay attack, man-in-the-middle attack.
FIGURE 1. Wearable medical sensor network.

5) Efficiency: evaluation comparisons show that our protocol has better performances (e.g., communication overhead, computational overhead).

The remainder of this paper is organized as follows: Section II goes over some preliminaries and section III describes our authentication scheme and Sections IV and V perform formal and informal security analysis on it. Section VI analyzes the performance of our scheme and finally, section VII concludes the paper.

II. PRELIMINARIES

We are applying wireless wearable sensor networks to the IoT-based telemedicine system to enable doctors, nurses and caregivers to monitor patients dynamically and in real time (shown in Fig. 1). Patients are monitored dynamically in that while the doctor is on the move, he always has access to the patient’s real-time sensor readings from any location through an application on his mobile device. The dynamic nature also implies that the patient can continue to live actively while constantly being tracked through the wireless sensors hooked on to limited-resource wearable devices. Patients are monitored in real-time meaning that whenever the doctors log on to the system, they have access to the current sensor readings and not static data like in medical information systems.

In designing our scheme, we assume that any adversary has the following capabilities when accessing our WMSN:

1) An adversary can conduct power analysis attack to obtain the information stored in the smart card and sensors [24].
2) An adversary can intercept, modify, delete and replay all messages transmitted over public channels.
3) An adversary can obtain user fingerprint through the use of putty and gelatin or a high-quality scanner.

The notations used throughout our scheme are presented in Table 1 below.

III. PROPOSED AUTHENTICATION AND KEY AGREEMENT SCHEME

Our scheme consists of five phases described in this section. These include doctor/nurse and sensor registration, doctor/nurse login, authentication and key agreement, continuous monitoring and the password change phase.

A. REGISTRATION

1) DOCTOR/NURSE REGISTRATION

Through the registration phase, a legal user $U_i$ obtains his/her smart card from the trusted server. Communications between $U_i$ and the trusted server take place over a secure channel as it is a one-time process. Details are as follows:

Step 1: $U_i$ inserts his/her identity $ID_i$, chooses a password $PW_i$ and imprints fingerprint $BIO_i$. $U_i$ computes $(R_i, P_i) = \text{Gen}(BIO_i)$ and $HPW_i = h(PW_i || R_i)$. He/she then sends $\{ID_i, PW_i, R_i\}$ to the trusted server.

Step 2: The trusted server chooses a random number $r_0$ and uses its secret key $x$ to compute $X_{S_i} = h(ID_i, PW_i, R_i)$, $A_i = X_{S_i} \oplus HPW_i$ and $L_i = h(R_i || X_{S_i} || ID_i)$. The server then generates another random number $r_1$ to compute $B_i = h(r_1 || L_i)$, $C_i = r_1 \oplus X_{S_i}$ and $Y = xP$. It then chooses a third random number $r_2$ to compute the dynamic identity $TID_i = h(r_2 P_{ID_i})$. The trusted server stores $\{ID_i, h(TID_i P_x), r_0 \oplus x, Honey_{List}\}$ in its database where the Honey_{List} is meant to track the number of failed logins to block a user exceeding a specific threshold as is done in [1], and is set to 0 at the beginning.

Step 3: The trusted server issues a smart card $SC$ to $U_i$ containing $SC = \{A_i, B_i, C_i, Y, P, P_i, TID_i\}$.

2) SENSOR REGISTRATION

Similarly, communications between the sensor and the trusted server take place over a secure channel as it is a one-time process. Details are as follows:

Step 1: The patient is hooked up to the ECG sensor which collects data. The data is segmented and processed as in [24] to

| Parameter | Meaning |
|-----------|---------|
| $U_i, S_i$ | User $i$ (could be a nurse, or a doctor…) and wearable sensor $j$, respectively |
| $ID_i, PW_i$ | User $i$’s real identity and real password |
| $TS$ | A trusted server |
| $P$ | A Point in the graph $G$ |
| $r_i$ | Random number $(i=0,1,2,3,4)$ |
| $x$ | Secret key of the trusted server |
| $\text{Gen}(\bullet), \text{Rep}(\bullet)$ | Fuzzy extraction function |
| $SID_i$ | $S_i$’s real identity |
| $PID$ | Patient’s real identity |
| $\text{FeatureSet}_{ECG}$ | ECG feature set that has been preprocessed and auto-correlated using techniques like [25] |
| $TID_i$ | User $i$’s dynamic identity |
| $X$ | A secret value known by $U_i$ and $TS$ |
| $X_A$ | A secret value known by $U_i$ and $TS$ |
| $X_B$ | A secret value known by $S_i$ and $TS$ |
| $SK_{i,j}$ | A secret session key for data communications between user $i$ and the sensor $j$ |
generate a FeatureSet\textsubscript{ECG}. This is sent to the trusted server together with the PID which the patient enters directly in the secure channel.  

Step 2: The sensor sends its SID\textsubscript{j} to the trusted server.  

Step 3: The trusted server uses PID and FeatureSet\textsubscript{ECG} as training data for the convolutional neural network inside the server as is used in [17]. The server also stores \( \{ h(SID\textsubscript{j}|x), PID \oplus x \} \) in its database and computes \( C_0 = h(x) \). It then sends \( \{ C_0 \} \) to the sensor.

**B. DOCTOR/NURSE LOGIN**  

In order to establish a connection with the sensor through the trusted server, \( U_i \) must login to the system. The following steps need to be executed:  

**Step 1:** \( U_i \) inserts ID\(_i\) and PW\(_i\), and imprints fingerprint B\(_{i}\).  

**Step 2:** The smart card computes \( R_i' = Rep(B\(_{i}\), P\(_i\)) \), \( X_{S_i}' = A_i \oplus h(PW\(_i\)||\( R_i' \)), \( r_i' = C_i \oplus h(X_{S_i}'\text{|}|ID\(_i\)) \) and \( B_i' = h(r_i'|L_i') \). It then checks if \( B_i' = B_i \). If they are equal, it selects a random number \( \alpha \in Z_{p^\ast} \) and computes \( X_i = \alpha P, X = \alpha Y, \text{ DID}_i = TID_i \oplus h(X,PX) \) and \( M_{U_i,G} = h(X_{S_i}'\text{|}|PX,PX) \). Next, the smart card terminal sends \( \{ \text{DID}_i, X_i, M_{U_i,G} \} \) to the trusted server.

**C. AUTHENTICATION AND KEY AGREEMENT**  

In this phase, the trusted server receives the login request message from \( U_i \) and mutually authenticates itself with the user and the sensor. After successful mutual authentication, \( U_i \) and \( S_j \) establish a common secret session key \( SK_{i,j} \), which is used for future secure communications between them. Details follow:  

**Step 1:** The trusted server computes \( X' = xX_i \) and \( TID'_i = \text{DID}_i \oplus h(X,PX) \). It then uses \( h(TID'_i/PX) \) to lookup the corresponding \( r_0, ID_i \) and \( \text{Honey\_List} \) from its database.  

If \( \text{Honey\_List} \) threshold > threshold, the server thinks the smart card has been suspended and rejects the request. If it cannot find an entry for \( h(TID'_i/PX) \) in its database, it also rejects the login request. Otherwise, it proceeds to compute \( X'_{S_j} = h(ID_i\text{|}|x)|r_0) \). If \( M_{U_j,G} \neq h(X'_{S_j}\text{|}|PX,PX') \), then the server increments the value in the \( \text{Honey\_List} \) by 1. Otherwise, it looks up PID using \( h(SID\textsubscript{j}\text{|}|x) \) and chooses \( \beta \in Z_{p^\ast} \) to compute \( Y_i = \beta P, X_{S_j} = h(C_0\text{|}|SID\textsubscript{j}\text{|}|PID) \), \( M_{G,S_j} = h(X_{S_j}\text{|}|PX,PSID\textsubscript{j}\text{|}|PTG) \), and \( X_{S_j} = X_{S_j} \oplus X_{S_j} \). The server sends \( \{ X_i, M_{G,S_j}, X_{S_j}, Y_i, T_j, \} \) to the sensor, where \( T_j \) is a timestamp.  

**Step 2:** At this time, the patient would have been hooked up to the ECG sensor and entered his PID to the sensor. The sensor checks the freshness of \( T_C \) and computes \( X'_{S_j} = h(C_0\text{|}|PID)\text{|}|SID\textsubscript{j} \). If \( M_{G,S_j} \neq h(X_{S_j}\text{|}|PX,PSID\textsubscript{j}\text{|}|PTG) \), it rejects it. Otherwise, the sensor computes \( \beta' = X_{S_j} \oplus X'_{S_j} \), \( k_j = h(X'_{S_j}\text{|}|PT) \), \( M_{S_j,G} = h(k_j\text{|}|Y_j\text{|}|X_{S_j}\text{|}|X_j\text{|}|T_j\text{|}|PID) \), \( Z = \beta'X_i \), and \( SK_{i,j} = h(Z\text{|}|T_j) \). After that, it sends \( \{ M_{S_j,G}, T_j, E[\text{FeatureSet}\textsubscript{ECG}\text{|}|SK_{i,j}] \} \) to the trusted server, where \( T_j \) is a timestamp.  

**Step 3:** The trusted server checks the freshness of the timestamp \( T_j \) and computes \( k_j' = h(X_{S_j}\text{|}|PT) \), \( Z' = \beta X_i \) and \( SK_{i,j}' = h(X,PZ'||Y_j) \). It then decrypts \( E[\text{FeatureSet}\textsubscript{ECG}\text{|}|SK_{i,j}] \) and gets \( \text{FeatureSet}\textsubscript{ECG} \). The neural network uses \( \text{FeatureSet}\textsubscript{ECG} \) to find PID'. Next, the server checks if \( M_{S_j,G} = h(k'_j\text{|}|Y_j\text{|}|X_{S_j}\text{|}|X_j\text{|}|T_j\text{|}|PID') \). If they are not equal, the server rejects the session because this means that the neural network did not match the received signals with the correct PID. Otherwise, it generates a new random number \( r_3 \), computes \( M_{G,U_i} = h(X'_{S_j}\text{|}|Z_{U_i}\text{|}|\text{TID}_i) \), \( TID'_{new} = h(r_3\text{|}|PID_i) \) and updates the database entry to \( h(TID'_{new}\text{|}|x) \). The server sends \( \{ Y_j, M_{G,U_i}, r_3 \} \) to the user.  

**Step 4:** The user checks if \( M_{G,U_i} = h(X'_{S_j}\text{|}|Z_{U_i}\text{|}|\text{TID}_i) \). If they are not equal, authentication fails. Otherwise, it computes \( SK'_{i,j} = h(X_i\text{|}|\alpha Y_j\text{|}|Y_j) \) and updates \( TID_i = h(r_3\text{|}|PID_i) \) in its smart card.

**D. CONTINUOUS MONITORING**  

Data collected from sensor is continuous sent to the doctor after the one-time authentication has completed successfully. This data is encrypted using a session key \( SK'_{i,j} = h(X_i\text{|}|\alpha Y_j\text{|}|Y_j) \). Also, data collected from sensor should be sent to the trusted server periodically, e.g., 30minutes. This could help to verify patient through comparisons of the cached the PID and trained results from the trust server using neural network to train data. If both of them are matched, it means the patient is the registered one and the session continues. If they don’t match, the server warns the doctor that the patient might have changed and the doctor may or may not choose to terminate the session.

**E. PASSWORD CHANGE**  

This phase enables the legal user \( U_i \) to change his/her password and biometric without communication with the server. Details are seen as follows:  

**Step 1:** \( U_i \) inserts ID\(_i\) and PW\(_i\), and imprints fingerprint B\(_{i}\).  

**Step 2:** The smart card computes \( R_i* = Rep(B\(_{i}\), P\(_i\)) \), \( X_{S_i}* = A_i \oplus h(PW\(_i\)||\( R_i\text{|}|\text{SID}_i\text{|}|\text{PID}_i\text{|}|\text{X}_i\text{|}|\text{Y}_i\text{|}|\text{T}_i\text{|}|\text{ID}_i\) \), \( L_i* = h(R_i*\text{|}|X_{S_i}\text{|}|\text{SID}_i\text{|}|\text{PID}_i\text{|}|\text{X}_i\text{|}|\text{Y}_i\text{|}|\text{T}_i\text{|}|\text{ID}_i\) \). It then checks if \( B_i* \neq B_i \). If they are not equal, the terminal rejects the smart card. Otherwise, it prompts the user to insert a new password.  

**Step 3:** The user inserts his/her new password PW\(_{new}\).  

**Step 4:** The smart card computes \( A_i* = X_{S_i}\text{|}|h(PW_{new}\text{|}|\( R_i* \) \) and generates a new random number \( r_4 \) to compute \( B_{new} = h(r_4\text{|}|L_i* \) and \( C_{new} = h(r_4\text{|}|h(X_{S_i}\text{|}| PTG) \). It then replaces \( \{ A_i, B_i, C_i \} \) in the smart card with \( \{ A_{new}, B_{new}, C_{new} \} \).

**IV. SECURITY ANALYSIS**  

**A. FORMAL PROOF OF AUTHENTICATION AND KEY AGREEMENT USING BAN LOGIC**  

Table 2 below shows the notation used to describe the protocol logic as well as the BAN logic postulates [26].  

We used BAN logic to formally prove the security of our scheme. The proposed scheme should satisfy the following
TABLE 2. BAN Logic Notation and Postulates

| Notation / Postulate | Meaning |
|----------------------|---------|
| $A \Rightarrow B$    | $A$ believes $B$ |
| $A \Rightarrow B$    | $A$ has jurisdiction over $B$ (A can generate or compute $B$) |
| $A \equiv B$          | $A$ once said $B$ |
| $∥B∥$                | Fresh $B$ (generated in current session) |
| $A \triangleright B$ | $A$ has jurisdiction over $B$ (A can generate or compute $B$) |
| $A \triangleright B$ | $A$ once said $B$ |
| $(A)_k$              | $A$ is hashed using key $K$ |
| $(A)_k$              | $A$ is encrypted using key $K$ |
| $A \vdash \#(X)$     | $A$ is a message meaning rule |
| $B \vdash X$         | $B$ is a jurisdiction rule |
| $A \vdash \#(X)$     | $A$ is a freshness conjugation rule |

security goals for the session key $SK_{i,j}$: (1) $G_1$: $U_i \equiv S_j \Rightarrow U_i \leftrightarrow S_j$; (2) $G_2$: $S_j \equiv U_i \equiv S_j \equiv S_j \equiv S_j \equiv S_j$ (3) $G_3$: $U_i \equiv S_j \equiv S_j \equiv S_j$.

The following represents the idealized version of our scheme:

M1: $U_i \rightarrow TS: ((DID_i, X_i, U_i \leftrightarrow TS))_{X_{S_j}}$
M2: $TS \rightarrow S_j: (X_i, SID_j, T_G)_{X_{S_j}}$
M3: $S_j \rightarrow TS: (k_j, Y_j, T_j, PID)_{X_{S_j}}$
M4: $TS \rightarrow U_i: (\alpha Y_j, X_i, T_IID)_{X_{S_j}}$

The following defines the initial assumptions we made about the state of the scheme: $H_1: S_j \equiv \#(T_G)$; $H_2: TS \equiv \#(X)$; $H_3: TS \equiv \#(T_j)$; $H_4: U_i \equiv \#(X); H_5: TS \equiv U_i \equiv S_j \equiv S_j \equiv S_j \equiv S_j$; $H_6: TS \equiv S_j \leftrightarrow TS; H_7: S_j \equiv S_j \leftrightarrow TS; H_8: U_i \equiv U_i \equiv S_j \rightarrow U_i \leftrightarrow S_j$; $H_9: U_i \equiv U_i \leftrightarrow S_j$; $H_{10}: S_j \equiv U_i \leftrightarrow S_j$.

Formal security analysis of the idealized scheme is as follows:

From M1, we get S1: $TS < (DID_i, X_i, U_i \leftrightarrow TS)_{X_{S_j}}$. Using $H_5$ and $S_1$, we can apply the message meaning rule:

$$TS \equiv U_i \leftrightarrow TS, TS \equiv (DID_i, X_i, U_i \leftrightarrow TS)_{X_{S_j}}$$

Then we can get S2: $TS \equiv U_i \sim (DID_i, X_i, U_i \leftrightarrow TS)$. Using $H_2$ and M1, we can apply the freshness conjugation rule:

$$TS \equiv \#(X)$$

Thus we can get S3: $TS \equiv \#(DID_i, X_i, U_i \leftrightarrow TS)$. Using S3 and S2, we can apply the none verification rule:

$$TS \equiv \#(DID_i, X_i, U_i \leftrightarrow TS), TS \equiv U_i \sim (DID_i, X_i, U_i \leftrightarrow TS)$$

Therefore, we have $TS \equiv U_i \equiv (DID_i, X_i, U_i \leftrightarrow TS)$.

From M2, we get S4: $S_j < (X_i, SID_j, T_G)_{X_{S_j}}$. Using $H_7$ and S5, we can apply the message meaning rule:

$$S_j \equiv S_j \leftrightarrow TS, S_j \sim (X_i, SID_j, T_G)_{X_{S_j}}$$

$$S_j \equiv TS \sim (X_i, SID_j, T_G)$$

Similar, using $H_1$ and M2, we can apply the freshness conjugation rule:

$$S_j \equiv \#(T_G)$$

Then, we apply the none verification rule:

$$S_j \equiv \#(X_i, SID_j, T_G), S_j \equiv TS \sim (X_i, SID_j, T_G)$$

Thus, we have $S_j \equiv TS \equiv (X_i, SID_j, T_G)$.

From M3, we get S5: $TS < (k_j, Y_j, T_j, PID)_{X_{S_j}}$. Using $H_6$ and S5, we can apply the message meaning rule:

$$TS \equiv S_j \leftrightarrow TS, TS \sim (k_j, Y_j, T_j, PID)_{X_{S_j}}$$

By using H3, the freshness conjugation rule and the none verification rule, we have $U_i \equiv TS \equiv (X_i, Y_j, T_j, PID)_{X_{S_j}}$.

Similarly, from M5 and using $H_8$ and $H_4$, the freshness conjugation rule and the none verification rule, we have $U_i \equiv TS \equiv (X_i, Y_j, T_j, PID)_{X_{S_j}}$.

Since $SK_{i,j} = h(\alpha Y_j | \alpha Y_j | Y_j)$ and given $S_j \equiv TS \equiv (X_i, SID_j, T_G)$ and $TS \equiv S_j \equiv (k_j, Y_j, T_j, PID)$, we have $U_i \equiv S_j \equiv U_i \leftrightarrow S_j$. Therefore, we have achieved goal G1.

Similarly, we have $S_j \equiv U_i \equiv U_i \leftrightarrow S_j$, and then achieves goal G2.

Using $H_9$ and $U_i \equiv S_j \equiv U_i \leftrightarrow S_j$ and the jurisdiction rule, we have $U_i \equiv U_i \leftrightarrow S_j$. Thus, we achieve goal G3.

Using $H_{10}$ and $S_j \equiv U_i \equiv U_i \leftrightarrow S_j$ and the jurisdiction rule, we have $S_j \equiv U_i \leftrightarrow S_j$. Thus, it achieved goal G4.

We have successfully proved goals G1, G2, G3 and G4. Therefore, we can conclude that our scheme ensures that the user $U_i$ and server $S_j$ have been mutually authenticated and have established a shared session key $SK_{i,j}$.

B. INFORMAL SECURITY ANALYSIS

1) USER (DOCTOR/NURSE) IMPERSONATION ATTACK

If an attacker accesses to a smart card and obtains its content, it cannot compute $M_{U_i,C} = h(X_{S_i} | PX | PX)$ since it does not
know $X_i'$ from $A_i \oplus h(PW_i||R_i')$. The reason is that $R_i'$ and $PW_i'$ are never stored in the server database, and are never computed due to one-way hash function feature. Thus, resists impersonation attack.

2) WEARABLE MEDICAL SENSOR IMPERSONATION ATTACK
If an attacker accesses to $C_0$ in the sensor, it still cannot be able to compute $M_{X_i,G} = h(k_i,PT_j, PX_i, PX_j, PID)$ due to without known of $X_i$ and $Y_j$. Computing of two of $X_i$ and $Y_j$ requires PID and $\beta$. Similarly, the attacker cannot construct a valid $SK_{i,j}$.

If the attacker accesses to the trusted server database, but it needs need $x$ to get $PID$. This could avoid the drawback in ang et al. [1] where once the sensor is compromised, carrying out an impersonation attack and computing the session key is simple.

3) PATIENT IMPERSONATION ATTACK
Since the patient’s PID is never stored as plain text, it is not possible for attacker to get PID without knowing $x$ since the server stores $\{h(SID_j)||x, PID \oplus x\}$. Thus, it could resist patient impersonation attack.

4) PHYSICAL SENSOR THEFT
Since with the help of prior knowledge about data collected from patient (for example, continuously monitoring the patient and verifying every 30 minutes that the patient identity cached on the server matches the result computed by the neural network using the detected ECG signals), it could provide a protection mechanism for cases where an attacker steals the sensor and hooks it up to a different patient. Thus, it could resist to physical sensor theft.

5) LOST/STOLEN SMART CARD ATTACK
If an adversary obtains $\{A_i, B_i, C_i, Y_i, P_i, PID\}$ from the smart card, it cannot construct a valid login message $\{DID_i, X_i, M_{U_i,G}\}$. This is because it is hard to get $X_i$, without knowing user’s $R_i$ and $PW_i$ or the random numbers $r_0$ and $x$. All these parameters are never stored in the smart card. Therefore, our scheme resists lost/stolen smart card attack.

6) REPLAY ATTACK
If an adversary obtains contents in the smart card, and then is eavesdropping over the public channel, it cannot construct a valid login message. If he/she tries to replay a valid login message, the trusted server will compute $TID_j' = DID_i \oplus h(X,PX)$' and then lookup $h(TID_j'Px)$ in its database. This is because that $TID_j$ is a dynamic identity that is recomputed with a new randomly generated number each time when user is successfully authenticated. Therefore, the login request will be rejected. Also, using the timestamp staleness, replaying messages will be detected by the trusted sever and the sensor.

7) USER ANONYMITY AND PROTECTION AGAINST TRACEABILITY
If an attacker monitors the public channel, it will not be capable of determining which user the messages correspond to with the help of the randomness of $DID_i(=TID_i \oplus h(X, PX))$ and the flesh of $X_i$ and $X_j$ per each login session. Thus, it is computationally infeasible for an attacker to determine which $ID_i$ the transmitted $DID_i$ belongs to. It is very difficult for an attacker to know if two messages belong to the same $ID_i$. What’s more, PID is never exposed at all over the public channel. Thus, our protocol could provide user anonymity and un-traceability.

8) OFFLINE PASSWORD GUESSING ATTACK
If the attacker obtains the information $\{A_i, B_i, C_i, Y_i, P_i, PID\}$ stored in the smart card of a legal user, it cannot guess the correct password. This is because the password is protected by the one-way hash function $HPW_i = h(PW_i||R_i)$. It is impossible to guess these four parameters correctly at the same time. Thus, our protocol can prevent offline password guessing attack.

9) SYMMETRIC KEY PROTECTION
As discussed above, the session key is computed as $SK_{i,j} = h(Z||X_i||Y_j)$. Given $X_i$ and $Y_j$, an intruder cannot compute $Z$ without knowing $\beta$. Thus, it is hard to compute the session key $SK_{i,j} = h(Z||X_i||Y_j)$.

10) PROTECTION OF BIOMETRIC TEMPLATE
Biometric templates are protected using fuzzy extraction function. In addition, the patient’s ECG is always encrypted before being sent over the public channel. As previously discussed, it is computationally infeasible for an intruder to compute the session key.

V. FORMAL SECURITY VALIDATION: SIMULATION USING AVISPA TOOL
The Automated Validation of Internet Security Protocols and Applications, or AVISPA, is a push-button tool utilizing industrial-strength technology to build and analyze formal models of large-scale security sensitive protocols and detect both active and passive attacks they may be susceptible to [27]–[29]. Protocol schemes and their security properties are defined using a High-Level Protocol Specification Language (HLPSL). The architectural structure of the AVISPA tool constitutes several components. HLPSL is first translated into an intermediate format (IF) through the HLPSL2IF translator. IF is a low-level language that can be directly fed to the integrated verification back-ends which include On-the-fly Model-Checker (OFMC), CL-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC) and the Tree Automata-based Protocol Analyzer (TA4SP). All of them are used to measure whether a protocol is SAFE or UNSAFE and return a trace of the potential attack. Because the analysis method used by each of these tools is different, they may yield
different results in terms of the safety of the protocol and the sequence of events leading up to the trace [29].

Results from both OFMC and CL-AtSe backends proved our protocol to be safe against passive and active attacks (like replay and man-in-the-middle attack) under the Dolev-Yao model. Simulation results are shown in Fig. 2 and Fig. 3.

VI. PERFORMANCE ANALYSIS

We compare the performance and computational complexity of our protocol with [1], [9], [17], and [23]. Note that [1], [9] and [17] are user-sensor three-factor authentication schemes whereas [23] is a continuous authentication scheme between a sensor and gateway node.

Table 3 shows the the security properties of our protocol with aforementioned schemes and observe that none of them satisfy all 11 security requirements. In contrast, our protocol satisfies all 11 security requirements our protocol in addition to providing continuous patient monitoring.

Table 4 compares the smart card’s computational overhead where $T_h$, $T_F$, $T_E$, $T_s$ represent the time complexity of the one-way hash function operation, fuzzy extraction operation, ECC multiplication operation, and the symmetric key encryption/decryption operation respectively. According to Maurya et al. [12], the time complexity (in ms) on a windows 7 operating system with Intel (R) core (TM) 2 Quad CPU Q8300, @2.50 Hz and 2 GB RAM is $T_h \approx 0.5$, $T_F \approx 0.5$, $T_E \approx 50.3$ and $T_s \approx 8.7$. The fuzzy extraction execution time is assumed to be equal to that of the one-way function since it can typically be constructed using universal hash functions or error-correcting codes requiring lightweight operations [12]. Because the XOR operation’s running time is negligible, it is ignored in our analysis.

The total computational overhead for the static authentication phase is $2T_h + 8T_E + 2T_F + 2T_s$ in our protocol, $21T_h$ in [17], $25T_h + 4T_F$ in [9] and $29T_h + 6T_E + T_F$ in [1]. Although [17] has the smallest computational cost, Table 4 shows that it has the weakest security and is not fit for practical applications. In comparison to [1] and [9], our protocol demonstrates acceptable overhead while maintaining stronger security.

Table 5 lists the computational overhead incurred by the sensor. It shows that the sensor computational overhead in our protocol is almost half that in [1] and [9]. This makes it much easier to deploy our protocol on sensors which have very limited resource capacity.

Fig. 4 provides the comparison of computational overhead. From the figure, we can see that our protocol needs 61.5 ms...
TABLE 5. Comparisons of Computational Overhead for Sensor

| Phase                  | Our protocol | [17] | [9] | [1] | [23] |
|------------------------|--------------|------|-----|-----|------|
| Static phase           | $5T_h + T_s$ | $3T_H$ | $4T_H + 2T_s$ | $6T_h + 2T_s$ | $20T_s$ |
| Continuous auth. / mon. | $T_s$        | N/A  | N/A | N/A | $10T_s$ |

FIGURE 4. Comparisons of communication overhead.

TABLE 6. Comparisons of Computational Overhead for Trusted Server

| Phase                  | Our protocol | [17] | [9] | [1] | [23] |
|------------------------|--------------|------|-----|-----|------|
| Authentication phase   | $9T_h + 3T_s + T_s$ | $8T_H$ | $11T_H$ | $13T_h + T_s$ | $10T_H$ |
|                        | $\approx 164.1$ms | $\approx 5.5$ms | $\approx 56.8$ms | $\approx 5$ms |

FIGURE 5. Comparisons of communication overhead.

VII. CONCLUSION

In this paper, a new ECC based lightweight static and continuous mutual authentication and key agreement protocol is proposed to protect data privacy and provide mutual authentication between the doctor/nurse, trusted server, sensor and patient. Continuous authentication is provided with the help of prior knowledge of their data collected from patients. Security analysis showed that our protocol is resistant to user and sensor impersonation attacks, physical sensor theft and more.

REFERENCES

[1] C. Wang, G. Xu, and J. Sun, “An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks,” Sensors, vol. 17, no. 2, 2017, Art. no. 2946.
[2] P. Kumar, S. G. Lee, and H. J. Lee, “E-SAP: Efficient strong authentication protocol for healthcare applications using wireless medical sensor networks,” Sensors, vol. 12, no. 2, pp. 1625–1647, 2012.
[3] Y. Deng, C. Chen, W. Tsuar, Y. Tang, and J. Chen, “Internet of Things based design of a secure and lightweight body area network (BAN) healthcare system,” Sensors, vol. 17, no. 12, pp. 1–18, 2017.
[4] I. Chang, T. Lee, T. Lin, and C. Liu, “Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks,” Sensors, vol. 15, no. 12, pp. 29841–29854, 2015.
[5] A. S. K. Pathan, H. W. Lee, and C. S. Hong, “Security in wireless sensor networks: Issues and challenges,” in Proc. 8th Int. Conf. Adv. Commun. Technol., 2006, pp. 1043–1048.
[6] M. L. Das, “Two-factor user authentication scheme in wireless sensor networks,” IEEE Trans. Wireless Commun., vol. 8, no. 3, pp. 1086–1090, 2009.
[7] B. Vaidya, D. Makrakis, and H. Mouftah, “Two-factor mutual authentication with key agreement in wireless sensor networks,” Secur. Commun. Netw., vol. 9, no. 2, pp. 171–183, 2012.
[8] J. Kim, D. Lee, M. Jeon, Y. Lee, and D. Won, “Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks,” Sensors, vol. 14, no. 4, pp. 6443–6462, 2014.
[9] Y. Park and Y. Park, “Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks,” Sensors, vol. 16, no. 12, 2016, Art. no. 2123.
[10] C. T. Li and M. S. Hwang, “An efficient biometric-based remote authentication scheme using smart cards,” J. Netw. Comp. Appl., vol. 33, pp. 1–5, 2010.
[11] A. K. Das, “A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor,” Int. J. Commun. Syst., vol. 30, no. 1, Jan. 2015, doi: 10.1002/dac.2933.
[12] A. K. Maurya and V. N. Sastry, “Fuzzy extractor and elliptic curve based efficient user authentication protocol for wireless sensor networks and Internet of Things,” Information, vol. 8, no. 136, 2017, doi: 10.3390/info8040136.
[13] J. Ryu, H. Lee, H. Kim, and D. Won, “Secure and efficient three-factor protocol for wireless sensor networks,” Sensors, vol. 18, no. 12, 2018, Art. no. 4481.
[14] F. Wu, L. Xu, S. Kumari, and X. Li, “An improved and provably secure three-factor user authentication scheme for wireless sensor networks,” Peer-to-Peer Netw. Appl., vol. 11, pp. 1–20, 2018.
[15] Y. Choi, Y. Lee, and D. Won, “Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction,” Int. J. Distrib. Sens. Netw., vol. 12, no. 1, pp. 1–16, 2016.
YING ET AL.: EFFICIENT AUTHENTICATION PROTOCOL FOR CONTINUOUS MONITORING IN MEDICAL SENSOR NETWORKS

[16] J. Moon, D. Lee, Y. Lee, and D. Won, “Improving biometric-based authentication schemes with smart card revocation/revissue for wireless sensor networks,” Sensors, vol. 17, no. 5, Apr. 2017, Art. no. 940.

[17] J. Jung, J. Moon, D. Lee, and D. Won, “Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks,” Sensors, vol. 17, 2017, doi: 10.3390/s17030644.

[18] L. Zhang, Y. Zhang, S. Tang, and H. Luo, “Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement,” IEEE Trans. Ind. Electron., vol. 65, no. 3, pp. 2793–2805, Mar. 2018.

[19] D. Xu, J. Chen, S. Zhang, and Q. Liu, “Privacy-preserving and efficient truly three-factor authentication scheme for tele-care medical information systems,” J. Med. Syst., vol. 42, no. 11, 2019, doi: 10.1007/s10916-018-1047-x.

[20] G. Wu, J. Wang, Y. Zhang, and S. Jiang, “A continuous identity authentication scheme based on physiological and behavioral characteristics,” Sensors, vol. 18, no. 1, Jan. 2018, Art. no. 1104.

[21] G. Cola, M. Avvenuti, F. Musso, and A. Vecchio, “Gait-based authentication using a wrist-worn device,” in Proc. 13th Int. Conf. Mobile Ubiquitous Syst.: Comput., Netw. Serv., 2016, pp. 208–217.

[22] S. Mondal and P. Bours, “Continuous authentication in a real world setting,” in Proc. 5th Int. Conf. Adv. Pattern Recognit., Jan. 2015, pp. 1–6.

[23] Y. Chuang, N. Lo, C. Yang, and S. Tang, “A lightweight continuous authentication protocol for the internet of things,” Sensors, vol. 18, 2018, Art. no. 1104.

[24] J. Johnson, A. Menezes, and T. S. Vanstone, “The elliptic curve digital signature algorithm (ECDSA),” IJIS, vol. 1, pp. 36–63, 2001.

[25] Q. Zhang, D. Zhou, and X. Zeng, “HeartID: A multi-faceted solution convolutional neural network for ECG-based biometric human identification in smart health applications,” IEEE Access, vol. 5, pp. 11905–11916, 2017.

[26] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” IEEE Trans. Comput., vol. 8, pp. 138–150, 1990.

[27] S. K. Haftaz Islam, R. Amin, G. P. Biswas, M. Sabzinejad Farash, X. Li, and S. Kumari, “An improved three party authenticated key exchange protocol using hash function and elliptic curve cryptography for mobile-commerce environments,” J. King Saud Univ. – Comput. Inf. Sci., vol. 29, no. 3, pp. 311–324, 2017.

[28] AVISPA, “Automated validation of internet security protocols and applications,” Accessed: Jan. 2019. [Online]. Available: http://www.avispa-project.org/

[29] AVISPA, “The HLPSL tutorial - A beginner’s guide to modeling and analyzing internet security protocols,” Inf. Soc. Technol. Programme, IST-2001-39252, 2006.

BIDI YING received the B.S. degree in communication engineering from Hangzhou Dianzi University, Hangzhou, China, in 2003, and the Ph.D. degree in information and communication engineering from Zhejiang University, Hangzhou, China. She is currently an Associate Professor with the School of Information and Electronic Engineering, Zhejiang Gongshang University. She also is an Associate Researcher with the University of Ottawa, Ottawa, ON, Canada. Her research interests include security or privacy in social networks, vehicular networks, Adhoc network, and wireless sensor network. She was a Member of the program committees of many conferences.

NADA RADWAN MOHSEN received the bachelor of applied science degree in software engineering and master degree in computer science degree from the University of Ottawa, Ottawa, ON, Canada, in 2018 and 2019, respectively. She is currently a Full Stack Software Developer with Cognos Analytics division, IBM. Her thesis defense was exploring a new lightweight and efficient authentication protocol for continuous static and dynamic patient monitoring in wireless medical sensor networks. She has authored or coauthored international publications in highly recognized outlets. Her research interests include authentication, E-health, body sensor network, IoT, Elliptic curve cryptography, and biometrics.

AMIYA NAYAK received the B.Math. degree in computer science and combinatorics and optimization from the University of Waterloo, Waterloo, ON, Canada, in 1981, and the Ph.D. degree in systems and computer engineering from Carleton University, Ottawa, ON, Canada, in 1991. He has more than 17 years of industrial experience in software engineering, avionics and navigation systems, and simulation and system level performance analysis. He is currently a Full Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. He has authored or coauthored more than 300 research articles in international journals and conferences. His research interests include mobile computing, wireless sensor networks, and Internet of Things. He is an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY, Journal of Sensor & Actuator Networks, Future Internet, and International Journal of Distributed Sensor Networks. He was on the Editorial Board of several journals, including the IEEE TRANSACTIONS ON PARALLEL DISTRIBUTED SYSTEMS, INTERNATIONAL JOURNAL OF PARALLEL, EMERGENT AND DISTRIBUTED SYSTEMS, and EURASIP Journal of Wireless Communications and Networking. He has been a Guest Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS (Special Issue on Cloud Computing in Smart Grid Operation and Management), IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY (Special Issue on Federated Learning for 5G Envisioned Internet of Things) and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Special Issue on Networking Challenges in Cloud Computing Systems and Applications). He has coauthored two books: Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication and Hand- book of Applied Algorithms: Solving Scientific, Engineering, and Practical Problems, both published by John Wiley & Sons, New York.