Lightweight & Efficient Authentication for Continuous Static and Dynamic Patient Monitoring in Wireless Body Sensor Networks

By

Nada Ashraf Radwan Mohsen

A thesis submitted to

the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of

the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering & Computer Science
University of Ottawa
Ottawa, Ontario K1N 6N5, Canada

Acknowledgements

First and Foremost, I would like to express all my gratitude to Allah for giving me the spiritual support and strength to accomplish this dissertation successfully.

I would also love to express my deepest and sincere gratitude to my dear thesis supervisor, Dr. Amiya Nayak and to Dr. Bidi Ying for providing me with all necessary support and guidance throughout writing my thesis. Their valuable feedback and constructive criticism to my thoughts were behind the success of this research.

Most importantly, a special thanks to my parents Azza and Ashraf for their prayers, wise council, sympathetic ear and for always supporting and being there for me through the tears and sleepless nights. Their keen interest and encouragement were my motivation to continue every day. My brother Omar and sister Jana were my happy distraction and I could not have made it without them. Finally, I want to thank my fiancé Moustafa for his endless support and encouragement throughout.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Explanation
AES	Advanced Encryption Standard
AVISPA	Automated Validation of Internet Security Protocols and Applications
BAN Logic	Burrows–Abadi–Needham Logic
BSN	Body Sensor Network
BSNS	Trusted Body Sensor Network Server
CCMCDHP	Chaotic Map Computational Diffie–Hellman Problem
CCMDLP	Computational Chaotic Map Discrete Logarithm Problem
CL-AtSe	CL-based Attack Searcher
CPU	Central Processing Unit
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Computational Diffie–Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECG	Electrocardiograph
HIPAA	Health Insurance Portability and Accountability Act
HLPSL	High-Level Protocol Specification Language
IoT	Internet of Things
OFMC	On-the-fly Model-Checker
PC	Personal Computer
PDA	Personal Digital Assistant
RAM	Random Access Memory
SATMC	SAT-based Model-Checker
SDP	Sensor Data Provider
SHA	Secure Hash Algorithm
SPAN	Security Protocol Animator
SRP	Secure Remote Password
TA4SP	Tree Automata-based Protocol Analyzer
TMIS	Telecare Medical Information Systems
WMSN	Wearable Medical Sensor Networks
WSN	Wireless Sensor Network

Abstract

The emergence of the Internet of Things (IoT) brought about the widespread of Body Sensor Networks (BSN) that continuously monitor patients using a collection of tiny-powered and lightweight bio-sensors offering convenience to both physicians and patients in the modern health care environment. Unfortunately, the deployment of bio-sensors in public hacker-prone settings means that they are vulnerable to various security threats exposing the security and privacy of patient information. This thesis presents an authentication scheme for each of two applications of medical sensor networks. The first is an ECC based authentication scheme suitable for a hospital-like setting whereby the patient is hooked up to sensors connected to a medical device such as an ECG monitor while the doctor needs real-time access to continuous sensor readings. The second protocol is a Chebyshev chaotic map-based authentication scheme suitable for deployment on wearable sensors allowing readings from the lightweight sensors connected to patients to be sent and stored on a trusted server while the patient is on the move. We formally and informally proved the security of both schemes. We also simulated both of them on AVISPA to prove their resistance to active and passive attacks. Moreover, we analyzed their performance to show their competitiveness against similar schemes and their suitability for deployment in each of the intended scenarios.

Chapter 1: Introduction

1.1. Background

Information technology and the Internet of Things (IoT) have been used drastically in all fields and health care is no exception. 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) or Body Sensor Networks (BSN) consist of sensors attached to the patient's body such as ECG electrodes, pulse oximeters, temperature or blood pressure sensors that monitor the patient's physiological data periodically and send it over the internet to devices accessible by doctors and caregivers [2]. As a result, remote users can assess the patient's health and monitor his/her heart rate, temperature, blood pressure or blood oxygen level at any time without the need of being within physical proximity [2]. This provides convenience for both parties and improves the quality of care while maintaining the patient's comfort [2].

The nature of patient data is highly sensitive. Therefore, a major issue arises in that this data is being sent through the internet, a wireless public network susceptible to intruder attack. Sensors are deployed in a hacker-prone setting so attackers can intercept them and send incorrect data to the caregivers resulting in an incorrect diagnosis [3]. They can also listen to the sensed data to blackmail the patient [3]. This puts the patient's privacy at risk and is particularly challenging when the patient is monitored in real-time as it means that all his information will be continuously accessible to adversary attackers. In addition, government laws like the Health Insurance Portability and Accountability Act of 1996 (HIPAA) have regulated very strict rules for healthcare providers including that patients' vital signs are only revealed to authorized professionals (i.e., doctors, caregivers and nurses) and authorized family members [2]. In case of failure to abide by HIPAA, the healthcare provider is subject to strict civil and criminal penalties [2].

Therefore, strict precautions must be taken to prevent illegal access to the patient's private data through a strong user authentication mechanism. To prevent unauthorized access, all remote users accessing the sensed data must be authenticated. In addition, the health professional should also be able to authenticate the sensors from which they receive data and agree on a session key to encrypt messages exchanged between them. This creates a secure mutual authentication channel between the remote user and sensor nodes. Design of such protocols in BSNs where resources are very limited is particularly challenging because we require a balance between security, privacy and computational cost [1].

1.2. Problem Statement

Finding the right balance between security, privacy and performance efficiency is challenging particularly in applications handling sensitive data like patient vital signs and involving resource constrained devices such as wearable sensors. To date, there has not been a single complete end-to-end scheme that completely secures the entire communication channel from patient to sensor to server to caregiver. Moreover, there has been no end-to-end scheme that allows continuous patient identity verification as a means of preventing sensor theft. Furthermore, our security and performance analysis of similar schemes or those covering individual sections of our proposed complete scheme in chapters two, three and four shows that to date, no one has reached the perfect combination of meeting the desired security and communication and computational overhead constraints for specific application scenarios of body sensor networks as we have demonstrated in our two scenarios presented in chapters three and four.

State of the art schemes using hash functions have been found to be susceptible to online and off-line password-guessing attacks, denial-of-service attacks, user impersonation and stolen smart card attacks. Other schemes using nonce failed to prevent reflection attacks and provide user anonymity. Some schemes using biometrics failed to protect the biometric template through the use of a bio-hash or fuzzy extraction. Others designed for BSNs do not allow their user the to authenticate a sensor nor to be authenticated by the server resulting in a point of vulnerability.

The majority of schemes utilizing ECC were found to be very computationally heavy especially when targeted for deployment on lightweight sensors making their application unrealistic. This is in addition to the fact that some of them like [4] incorrectly used ECC giving an attacker sniffing the public channel with access to the smart card and biometrics the ability to compute the verification parameter needed for mutual authentication. Examined protocols using chaotic maps also could not support session key security, perfect forward secrecy and smart card revocation. They were also susceptible to replay and man-in-middle attack amongst others.

1.3. Requirements and Objectives

When designing an authentication scheme, one needs to understand the security requirements and performance constraints that the scheme must satisfy. As far as performance constraints are concerned, designing an authentication scheme for body sensor networks puts a huge constraint on computational power because sensors have very limited resources. This is particularly true for sensors embedded in wearable medical devices more than it is true for larger medical devices hooked up to sensors like ECG monitors. Schemes like those introduced in Wang et al. [1] and Park et al. [4] use expensive computations through their utilization of ECC on the sensor side making their scheme impractical in real-world applications of wearable sensors.

Prior literature has already laid out the security requirements that must be met for user authentication schemes designed to be used in body sensor networks. These are outlined below:

1. Mutual Authentication: The bio-sensors, trusted server and any user accessing patient information must undergo mutual authentication to verify each other's legitimacy.
2. User Anonymity: The user's identity (be it the patient, doctor or nurse) should always be kept private meaning that the scheme should prevent an attacker from being able to compute it [1].
3. Session Key Generation: User and sensor nodes must generate a session key that they use to encrypt messages they later exchange [1]. The session key should only be known to participating parties and should be unique per session.

4. Confidential Communication: All secret information should remain secret and private patient data should be securely encrypted before being sent over the public channel.
5. Offline Password Update: The user needs to be able to update his/her password without the involvement of the gateway node [5].
6. Quick Incorrect Password Detection: If a user enters an incorrect password, the scheme should support its rejection before the verification process begins [5].
7. Protection against Stolen-Verifier Attack: The verifier table should not expose any sensitive information needed by an adversary to carry out an impersonation attack or compute session keys [1].
8. Protection against Privileged Insider Attack: An authorized user should not be able to obtain any sensitive information enough to carry out any attack on another user [1].
9. Protection against Offline Password Guessing Attack [1].
10. Protection against Replay Attack and Parallel Session Attack [1].
11. Protection against Impersonation Attack [1].
12. Providing known-key security: If a previous session key has been compromised, an attacker should not be able to compute other session keys [1].

1.4. Contributions

In this thesis, we present an authentication scheme for each of two applications of medical sensor networks. In chapter three, we present an authentication scheme suitable for a hospital-like setting whereby the patient is hooked up to sensors connected to a medical device such as an ECG monitor while the doctor, who is constantly on the move, needs real-time access to continuous sensor readings. Our proposed protocol is an ECC-based scheme which provides [6]:

(1) Lightweight authentication suitable for deployment on medical devices to which lightweight sensors are connected.
(2) For the first time, a complete end-to-end scheme that can be deployed in a real-time environment across the doctor/nurse, trusted server, sensor and patient.

(3) Utilization of biometrics on both patient and caregiver ends to enhance security.
(4) Continuous monitoring for patients by verifying their physiological data and to prevent sensor theft attack.
(5) Biometric privacy protection through the use of fuzzy extraction.
(6) Caregiver and patient anonymity and prevention against traceability through the utilization of a dynamic identity.
(7) Proof of security as outlined in chapter three.
(8) Proof of performance efficiency as outlined in chapter three.

In chapter four, we present an authentication scheme suitable for deployment on wearable sensors allowing readings from the lightweight sensors connected to patients to be sent and stored on a trusted server while the patient is on the move. Our proposed protocol is Chebyshev chaotic map-based which provides [7]:

(1) Lightweight authentication - our scheme does not use any complex operations making it suitable for deployment on lightweight wireless sensors allowing for its use in a more flexible environment where patients are on the move.
(2) Two session keys for enhanced security.
(3) Patient/sensor anonymity and preventing traceability through use of dynamic identity.
(4) Biometric privacy protection through utilization of a bio-hash function.
(5) Proof of security as outlined in chapter four.
(6) Proof of performance efficiency as outlined in chapter four.

1.5. Thesis Organization

The remainder of this thesis is organized as follows:

Chapter two is a literature review of related work including an explanation of the key concepts used throughout the thesis.

Chapter three is a deep dive into our proposed end-to-end BSN authentication scheme based on elliptic curve cryptography. We talk about the target application scenario and scheme contributions and detail the steps in each phase involved in the scheme. We then perform formal security analysis using BAN logic as well as informal security analysis. Next, we describe and present our results from AVISPA, a simulation and formal verification tool which was utilized to prove our scheme is safe against both passive and active attacks. The next section does a performance analysis comparing the overall and phase-specific communication overhead as well as the individual computational overhead for the smart card, sensor and trusted server in our scheme with those in similar schemes.

Chapter four presents our second scheme: BSN authentication based on Chebyshev chaotic maps. Like the previous chapter, we talk about the target application scenario and scheme contributions and detail the steps in each phase involved in the scheme. We then formally analyze the security of our scheme by modelling the adversary's capabilities as queries and showing that it supports session key and known key security by proving that the probability an adversary can break the semantic security of the scheme is small. We also perform an informal security analysis and then proceed to simulation using AVISPA. Section 4.7 does a performance analysis comparing the computational overhead of our authentication phase with that of similar work. This is in addition to the analysis of the communication overhead incurred by the mobile device, server and biosensors.

Finally, chapter five concludes the thesis highlighting key findings and contributions. It also makes suggestions for future work.

1.6. List of publications from Thesis

(1) N. Mohsen, B. Ying and A. Nayak, "Authentication Protocol for Real-time Wearable Medical Sensor Networks using Biometrics and Continuous Monitoring," in *The 12th IEEE International Conference on Internet of Things [iThings 2019]*, Atlanta, 2019.

(2) B. Ying, N. Mohsen and A. Nayak, "Protection for e-health systems using three-factor user authentication," in *IEEE ICC*, Shanghai, China, 2019.

Chapter 2: Literature Review

2.1. Key Concepts

2.1.1. Elliptic Curve Digital Signature Algorithm (ECDSA)

Let p and q be two large prime numbers, F_p be a finite field, E/F_P be an elliptic curve $y^2 = x^3 + ax + b$ over F_p and G be a q-order subgroup of E/F_P. According to ECDSA, we conclude that for α, β ∈ Z_P^* and a point P in G, we can define [1] [4]:

- Elliptic curve discrete logarithm (ECDL) problem: given (P, αP), it is impossible to compute α within polynomial time.
- Elliptic curve computational Diffie–Hellman (ECDH) problem: given (αP, βP), it is impossible to compute αβP within polynomial time.

Because there does not exist a sub exponential-time algorithm to solve ECDL, it provides a very high level of per-key-bit security and is thus a great candidate for use in authentication schemes [8].

2.1.2. Chebyshev Chaotic Maps

Given $x \in (-\infty, +\infty)$, an integer n and a large prime number p, we define the function of Chebyshev polynomial $T_n(x)$: $(-\infty, +\infty) \to [-1, +1]$ to be $T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \bmod p$, where $n \geq 2$, $T_0(x) = 1$ and $T_1(x) = x$. Chebyshev polynomial has the following property: $T_{\alpha\beta}(x) = T_\alpha(T_\beta(x)) = T_\beta(T_\alpha(x))$, where α, β are integers and $x \in (-\infty, +\infty)$. The following two hard problems are used to analyze the security of e-HealthCare systems utilizing chaotic maps: Computational Chaotic Map Discrete Logarithm Problem (CCMDLP) and Computational Chaotic Map Computational Diffie–Hellman Problem (CCMCDHP) [9]:

(1) CCMDLP: Given two elements y and x, it is computationally infeasible to find an integer α such that $T_\alpha(x) = y$ [9].

(2) CMCDHP: Given x, $T_\alpha(x), T_\beta(x)$, it is computationally infeasible to compute $y = T_{\alpha\beta}(x)$ [9].

2.1.3. Bio-hash Function

Since biometric information is highly sensitive data, its use in user identification requires that it be secured and protected. The need for a strong and sophisticated matching technique leads to the introduction of the bio-hashing technique which combines an arbitrary authentication-independent pseudo-random number with the user's biometric template. The biometric template is converted to a feature vector and inserted into a transform function H(.) which combines the template T with the random key K to create a transformed template H(T,K). Furthermore, a bio-hash code is created H(Q,K) where Q is the stored value in the device (biometric query) and is used to check whether the user is registered or not by comparing it to the new value H(T,K) [10].

Identical biometric information creates the same H(Bio) and it is computationally infeasible to compute Bio given H(Bio); therefore, the bio-hash function protects biometric information from exposure to adversaries in addition to reducing the probability of occurrence of the denial of service attack [10].

2.1.4. Fuzzy Extraction

To securely authenticate using biometrics, a cryptographic technique called fuzzy extraction can be used. Fuzzy extraction consists of two randomized operations. The generator function takes in a biometric credential and produces a secret string and a public accessory. The reproduction function takes in a noisy biometric credential and the public accessory to produce the right secret string if and only if the level of noise in the biometric template is less than a set threshold [11].

2.2. State-of-the-Art Two Factor User Authentication and Key Agreement Schemes in WSN

In any authentication scheme, users can be authenticated using what they know (eg. password), what they own (eg. smart card) and what they are (eg. biometrics). The first user authentication and key agreement schemes introduced for use in communication networks utilized passwords only to authenticate users. Because such single-factor schemes were susceptible to a large number of security threats deeming them unsuitable for use in wireless sensor networks, long-term secret keys and smart cards were introduced to supplement passwords, improve security and create two-factor user authentication schemes [12]. However, wireless sensor networks are special in that the sensors are resource-constrained devices usually deployed in a hacker-prone setting making them incapable of securely storing long-term keys. As a result, long-term keys are usually stored into the gateway [13]. 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. [14] designed a two-factor authentication scheme to authenticate user and sensor nodes using a smart card and a password; however, the scheme did not establish a session key at all. To improve on this scheme, Vaidya et al. [15] proposed a two-factor mutual user authentication scheme with key agreement for WSNs. In 2014, Kim et al. [16] showed that [15] is 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.

In 2015, Chang et al. [1] found that [16] is vulnerable to sensor-impersonation, lost smart-card attack and man-in-the-middle attack. In addition, it is susceptible to violation of session key security and fails to protect user privacy. Chang et al. [12] proposed a low computational cost two-factor authentication and key agreement scheme for WSN which uses dynamic identity to ensure user privacy and also uses temporary secret keys to achieve session key security [12]. Although more computationally complex, Chang et al.'s scheme claims to provide mutual authentication, known-key security and resistance to privileged insider attack,

replay attack, parallel session attack, password guessing attack and stolen verifier attack [12]. Unfortunately, Chang et al. [12] was shown to be vulnerable to offline password guessing attack and to lack perfect forward secrecy by Park et al. [4]. 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 [4].

2.3. State-of-the-Art Three Factor User Authentication and Key Agreement Schemes in WSN: Focus on BSN

To address some of the weaknesses of two-factor authentication protocols including their failure to resist offline password guessing attack and inability to update user passwords, authentication schemes introduced biometric keys as a third factor. Biometrics have several advantages including the fact that they cannot be lost or forgotten and are difficult to copy, forge or break [17]. These properties make biometric-based schemes more reliable and more secure than conventional schemes [17].

Awasthi et al. [18] proposed a biometric-based authentication scheme for telecare medical information systems (TMIS) claiming to address weaknesses of previous schemes and to be computationally cheaper through utilizing hash functions and avoiding time consuming exponential computations making it suitable for low cost mobile devices. Unfortunately, Mishra et al. [19] demonstrated that [18] is vulnerable to both online and off-line password-guessing attacks as well as denial-of-service attacks and then proposed another biometric-based scheme using nonce to address the weaknesses in [18]. In 2014, Tan et al. [20] showed that [18] fails to resist reflection attacks in addition to failure to provide user anonymity. Tan et al. [20] then proposed an improved scheme for TMIS maintaining user anonymity. Arshad and Nikooghadam [21] showed that [20] does not resist replay and denial-of-service attacks and presented an improvement.

Yan et al. [22] designed another scheme addressing the above weaknesses; however, [23] found that [22] is insecure against off-line password guessing attack. It also does not provide

user anonymity and its login and password change phases are inefficient. Mishra et al. [23] used hash functions and nonce to propose an enhanced scheme which was later found to be vulnerable to user impersonation attack and identity guessing attacks by Sarvabhatla et al. [24]. Amin et al. [25] also proved that [23] is susceptible to server impersonation attack, session key computation attack and smart card loss attack. Moreover, [26] showed that [23] lacks perfect forward secrecy and is insecure against replay attack and man-in-the-middle attack.

In 2015, A.K. Das et al. [27] 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. [11] found that the scheme is in fact susceptible to stolen smart card attack, [28] proved its susceptibility to impersonation attack, and [29] showed it allowed user-forgery attack and offline-password guessing attacks. In 2016, Choi et al. [30] proposed another biometric based scheme addressing the issues of lack of accuracy of biometric recognition and user verification difficulty. However, Choi's scheme was found to be vulnerable to user impersonation attack and known plain-text attack in addition to not providing user anonymity. Other schemes making use of biometrics include [31] and [32]; however, they both lack off-line password update.

Wood et al. [33] presented a wide-area network (ALARM-NET) integrating environmental and physiological sensors, gateways, PDAs and PCs to allow real-time assisted-living and residential monitoring of medical data without the need to be deployed in a clinical environment. At a higher level, the ALARM-NET architecture spans wearable body networks (wireless sensor devices worn by residents and tailored to each resident's needs to allow sensing of physiological data and activity classification), emplaced wireless sensors (multi-hop wireless network consisting of sensor devices deployed in the living space with the purpose of monitoring environmental conditions such as dust, motion, temperature or light as well as forwarding patient data received through single-hop from the BSNs), IP-network elements, an AlarmGate application to manage system operations (allowing user interfaces to connect, authenticate and interact with the system) and act as an application-level gateway between the sensors and IP components, a back-end system for data storage and analysis and finally user interfaces to allow

doctors, nurses, patient and families to query sensor readings [33]. Kumar et al. [34] showed that ALARM-NET's architecture allows for network and data security. For example, users querying sensor data are authenticated before being provided system access using Secure Remote Password (SRP) protocol. In addition, Link-layer security suites are enabled for the wireless sensor networks and the sensors utilized ([35]) use an Advanced Encryption System (AES) for cryptographic modes [34] where the supported security modes are: none, CBC-MAC authentication only, CTR mode encryption-only and CMM combined with authentication and encryption [33]. However, [34] also showed that ALARM-NET does not embed sufficient security in its architecture. For instance, the sensor built-in cryptosystems do not support AES-based decryption meaning that intermediary nodes cannot access encrypted data. Moreover, Pai et al. [36] showed that ALARM-NET is subject to adversarial confidentiality attacks meaning that the resident's location can be leaked.

In 2016, Gope and Hwang proposed BSN-Care: a secure IoT-based healthcare system using BSN. BSN-Care makes use of a one-way hash-based authentication method that provides validation of users [37]. However, it was found that sensors are not authenticated before the collected data is delivered to the user. Also in 2016, Yeh proposed two authentication mechanisms that use both, a hash function to check if the sensor is legal and a public/private key to check if the user is legal [38]. Unfortunately, both proposed schemes generate heavy computational and communication overhead.

In 2018, Shen et al. [39] proposed a cloud-aided lightweight certificateless authentication scheme for wireless body area networks with claim to support user anonymity such that no one can obtain the user's real identity except the network manager, who can obtain it in the registration phase only. Also in 2018, Wei et al. [40] proposed an anonymous authentication scheme for wireless body area networks based on the use of a low entropy password to address weaknesses of the aforementioned schemes. The scheme claims to be computationally efficient and to provide user anonymity; however, in 2019, Liu et al. [41] found that the scheme presented in [40] is susceptible to personal information disclosure attack in addition to not providing three-factor authentication. Moreover, Wu et al. [42] presented an anonymous authentication scheme

and proved its security under the random oracle model; however, Odelu et al. [43] proved that that the scheme fails to protect the session key and the credentials privacy when the ephemeral secrets are revealed to an attacker. Furthermore, Li et al. [44] proposed a lightweight mutual authentication and key agreement protocol with anonymity for wireless body area networks but [45] demonstrated that a user in Li's scheme does not have the capability to authenticate a sensor nor to be authenticated by the server resulting in a point of vulnerability. To solve this, Chen et al. [45] proposed two authentication mechanisms to mutually authenticate the sensor and user and to mutually authenticate the sensor and server. Unfortunately, the schemes are heavy in terms of communication and computational overhead.

2.3.1. Authentication Schemes using Elliptic Curve Cryptography

In 2016, Park et al. [4] introduced a three-factor user authentication and key agreement protocol for WSN using the elliptic curve cryptosystem and fuzzy extraction. Even though it has a higher computational cost when compared to [1], [4] claims to support user anonymity and protect against user traceability, smart card loss attack, impersonation, man-in-the-middle, stolen verifier, known-key and offline password guessing attack. It also claims to support perfect forward secrecy and mutual authentication [4].

Despite Park et al.'s use of Elliptic Curve Cryptography (ECC), Wang et al. [1] proved that it lacks user anonymity and 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 $M_{U_i,G}$ can be computed using static knowledge readily accessible to an attacker sniffing the public channel with access to the smart card and biometrics. Moreover, its attempt at using dynamic identity made it susceptible to user traceability once an attacker has access to the public channel and the verifier table in the gateway. Moon et al. [46] 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. [11].

The aforementioned three-factor authentication schemes do not protect the biometrics which puts the biometric information at a potential risk of disclosure. For instance, some

schemes store the biometric template on the smart card while others transmit them over the public channel which is subject to intruder attack and eavesdropping. Moreover, some schemes include the biometric template as part of the input to the hash function when registering the user to the server. In reality, it is impractical to verify the hash function upon login unless the biometric templates exactly match which is highly unlikely.

To protect the biometric template, authentication schemes have introduced the concept of a bio-hash function, a secure and sophisticated matching technique for user identification [47]. The bio-hash function is described as a form of "cancellable biometric" (biometric templates that can be cancelled and replaced with an independent authentication factor) which combines biometric features with random vectors that are unique per user [47].

In 2011, Huang et al. [48] proposed a three-factor authentication scheme utilizing fuzzy extractors to generate biometric keys from the biometric templates. Yu et al. [49] claimed that the computational efficiency of [48] can be improved and presented an improvement proving its security using the random oracle model. In 2017, Jung et al. [10] proposed a three-factor user authentication scheme which uses smart card, password and biometrics protected through utilization of the bio-hash function. Wang et al. [1] proved that Jung et al. [10] is not resistant to impersonation attack and offline password guessing attacks in addition to not providing forward secrecy and user anonymity.

In 2017, Wang et al. [1] proposed an enhanced version using biometrics and the elliptic curve cryptosystem to further improve security and efficiency. Unlike Park et al. [4], Wang et al. [1] applies ECC to create a trap door by building a challenge "X" that is required to compute the verification parameter $M_{U_i,G}$ where "X" can only be computed if the dynamic a or the long-term key is known [1]. This made their scheme resistant to offline password guessing attack. Moreover, they used honey words and fuzzy-verifiers to further protect against offline-password guessing attack and provide user anonymity through applying a dynamic identity technique. Wang et al. [1] also claimed to support mutual authentication, user anonymity and forward secrecy as well as resisting privileged insider attack, verifier-stolen attack and replay attack.

Unfortunately, we found that Wang et al. suffers from smart card loss attack. If the adversary has access to the smart card and password and is able to get ID_i through hacking the gateway node database for example, he/she can easily compute $\{DID_i, X_i, M_{U_iG}, T_i\}$ and send an authentication request to the gateway. This is because the biometric helper string P_i is stored in the smart card so the protocol's use of biometrics becomes meaningless if the attacker has access to the card. In addition, the gateway node has no way of tracking whether an X_i has been used in a previous request. This means that an attacker can replay the authentication request using a new timestamp and it will go unnoticed as the gateway node proceeds to authenticate leading to unnecessary computational overhead. Moreover, an adversary can carry out an impersonation attack on the sensor node if he hacks the sensor node to find X_{S_j}. This can be done by simply choosing a b and computing the necessary parameters as a legitimate sensor would in addition to successfully computing the session key. Furthermore, given the computational complexity of ECC [1], it is impractical to execute scalar multiplication $Y_j = \beta P$ on the lightweight sensors in real world applications.

2.3.2. Authentication Schemes using Chaotic Maps

When comparing the computational cost of chaotic-map based authentication schemes to that of ECC-based schemes, we observe that overall, chaotic-map based schemes have a lower computational cost since chaotic map operations require lower power, bandwidth and computational requirements [50] [51] [26] relative to the scalar multiplication operation used in ECC. This makes chaotic maps more suitable for applications where the scheme must be deployed on resource constrained sensors.

In 2016, Li et al. [9] analyzed the security and performance of existing chaotic map based schemes including [52], [53] and [54] to find that they all lack the ability to provision smart card revocation, provide formal security verification and lacked a verification mechanism during the login phase. This is in addition to failing to prevent the time synchronization problem. Li et al.

[9] then proposed a new user authentication and key agreement scheme whose computational overhead is comparable with the aforementioned schemes. They used CCMDLP and CCMDHP to analyze the security of their scheme and proved that the interaction between the user and the server is anonymous and that the proposed system is secure against privileged-insider attack, lost/stolen smart card attack, off-line password guessing attack and impersonation attack. The scheme was also proven to demonstrate session key security, perfect forward secrecy and smart card revocation in addition to efficiently identifying the correctness of users' inputs. In 2019, Kumar et al. [55] found security flaws in [9] and proposed a secure elliptic curve cryptography based mutual authentication protocol for TMIS to address these flaws which include failure of the scheme to provide message authentication, failure to support session key security and failure to secure against impersonation attack during the healthcare center upload phase. Moreover, the protocol presented in [9] fails to provide patient anonymity and fails to protect against patient traceability.

In 2017, Zhang et al. [26] designed a protocol using chaotic map-based cryptography to address weaknesses of the scheme presented by Mishra et al. [23] which included replay attacks, man-in-the-middle attacks and failing to provide perfect forward secrecy. Moreover, another chaotic-map based scheme for telecare medicine information systems was proposed by Lee at al [50] to address weaknesses of previous schemes. However, [56] found that [50] is vulnerable to off-line password guessing attack given that an adversary has access to the information stored in the smart card using a power analysis attack and having the ability to intercept all exchanged messages over the public channel. This is in addition to being susceptible to denial of service attack because an attacker does not need to know the password in order to change the smart card security parameter resulting in a user with the correct password and valid smart card not being able to login. It also has an inefficient password change phase because the smart card carries out the password update successfully without first checking the correctness of the input [56].

2.4. Continuous Authentication Schemes

As discussed above, the vast majority of authentication schemes use one-time (static) authentication. This means that authentication is only invoked at the beginning of a communication session and means that 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 [57]. Because static authentication does not protect against session hijacking, continuous authentication has been introduced as a supplemental means of automatically verifying the legitimacy of a user.

With the aid of machine learning algorithms, some schemes made use of the embedded motion sensors in mobile devices or wearable motion sensors to authenticate users based on prior knowledge of their motion state [58] [59] [60]. For instance, Cola et al. [58] achieved user authentication through the collection of motion data from a wrist band together with an anomaly detection. With 15 volunteers, this technique demonstrated a 2.9% error rate. A few schemes continuously authenticate with every action the user performs. For example, Mondal et al. [59] collected keystroke and mouse usage behavior pattern data to authenticate with every action. Chuang et al. [60] proposed a lightweight static and continuous authentication protocol for sensors and gateway devices. The scheme statically authenticates the sensors every period of time and throughout this period, it uses timestamps and the remaining battery life of the sensor to continuously authenticate with every sensor to gateway node transmission. The protocol claims to support mutual authentication and use only lightweight computation (including XOR and hash) so as to be feasibly supported on the resource-limited sensors.

However, most schemes which implement continuous authentication perform a periodic kind of authentication using behavioral or physiological data including electroencephalogram (EEG), electrocardiogram (ECG) and photoplethysmography (PPG). Wu et al. [57] used a one-class machine learning algorithm to introduce a two-step authentication scheme utilizing an own-built fingertip sensor device capturing both physiological (PPG) and motion data to continuously identify the legitimacy of the wearer automatically and at set intervals. Their scheme achieved a 98.5% accuracy rate [57].

Extensive research [61] [62] [63] on the use of ECG for user identification has been carried out. Through the use of different techniques for feature extraction, Camara et al. [61] achieved at 97% accuracy rate. Under high levels of noise, Kang et al. [62] achieved a false acceptance rate of 5.2% and false rejection rate of 1.9%. This shows a great potential for the use of ECG for user identification especially that Kim et al. [63] talked about the existence of small-size compact ECG sensors. Because we aim to achieve continuous identity verification while achieving a high recognition accuracy, our scheme presented in chapter three makes use of the ECG signal for patient identity recognition.

Chapter 3: End-to-End BSN Authentication based on Elliptic Curve Cryptography

3.1. Introduction & Motivation

This chapter addresses the first of two application scenarios presented in this thesis making use of an IoT-based telemedicine system. This scenario uses BSN to enable doctors, nurses and caregivers to monitor patients dynamically and in real time (see Figure 1 below). In this hospital like setting, patients are monitored in real-time meaning that whenever the doctors log on to the system, he/she has access to the current sensor readings and not static data like in medical information systems. 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 is limited to the caregivers' end in that this scenario assumes the patient is statically hooked up to sensors connected through wires to a medical device such as an ECG monitor which may be wirelessly or through a wire, connected to a trusted server.

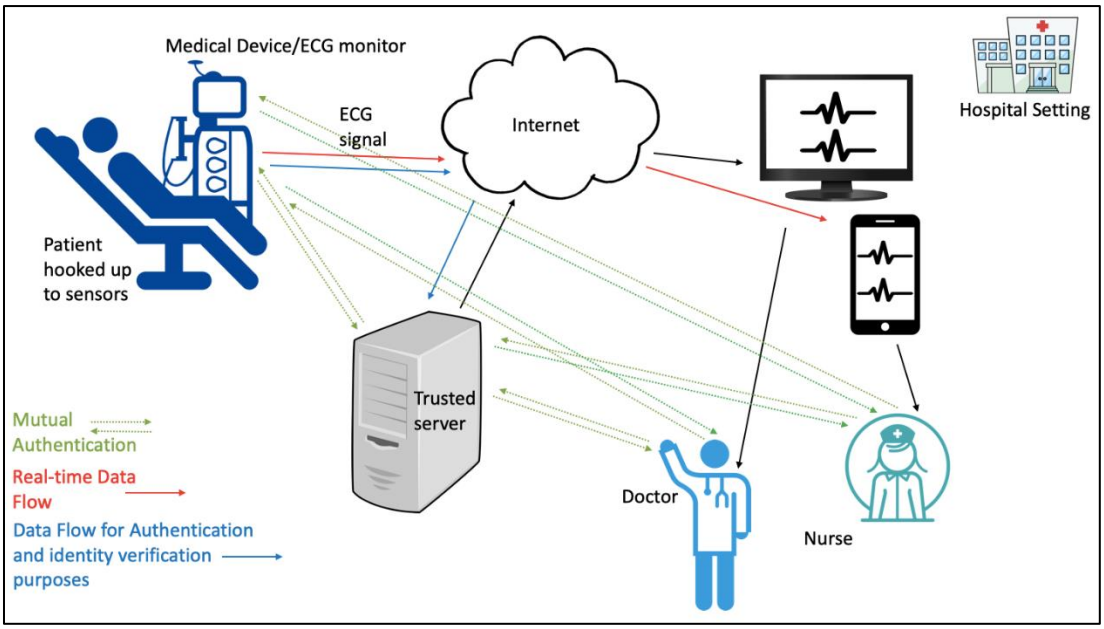


Figure 1 - Applying ECC Based Scheme on BSN

As previously mentioned, the open nature of wireless medical sensor networks in a public untrusted environment makes them vulnerable to various security threats and puts the security and privacy of patient information at risk. This chapter introduces a lightweight ECC-based mutual authentication and key agreement protocol to be used in real-time wireless medical sensor networks between doctors/nurses, trusted servers, sensors and patients. As shown in the above figure, our scheme mutually authenticates the sensor and trusted server as well as the caregiver and trusted server. We claim that our scheme is lightweight despite its use of ECC because deployment will be on the larger more resourceful medical devices connected to the sensors rather than the constrained sensors themselves. Unlike existing schemes, our scheme uses biometrics on both doctor/nurse and patient ends. It allows the doctor/nurse to login to the system using his/her biometrics and verifies patient identity by means of continuous monitoring of physiological data (e.g., ECG signals) in which verification of the patient identity is carried out automatically and at set intervals to detect physical theft of the sensor which may be hooked up to a different patient. Our scheme also uses dynamic identity to provide user (caregiver and patient) anonymity and mitigate against user traceability. After successful key agreement, data from the sensor is encrypted before being sent directly to the caregiver.

3.2. Chapter Contributions

In this chapter, we propose a lightweight biometric based authentication and key-agreement protocol to be used in body sensor networks and which utilizes ECC and dynamic identity. To securely authenticate using biometrics, we use the cryptographic technique called fuzzy extraction which consists of two randomized operations. The generator function takes in a biometric credential and produces a secret string and a public accessory whereas the reproduction function takes in a noisy biometric credential and the public accessory to produce the right secret string if and only if the level of noise in the biometric template is less than a set threshold [11].

To provide continuous monitoring, our scheme first mutually authenticates the doctor/nurse and sensor/patient through three-factor authentication and generates a session key between doctor/nurse and sensor. Similar to [58] [59] [60] which do continuous authentication, we then use the patient's ECG signals to achieve a continuous patient identity verification in the trusted server to ensure that after the initial authentication, the sensor is not stolen and hooked up to a different patient. 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 patient identity 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] [4], our scheme reduces communication /computational overhead.
(2) Continuous monitoring: Our protocol provides continuous monitoring for patients by verifying their physiological data (ECG signals). This helps detect physical sensor theft as hooking the sensor up to a different patient will be detected.
(3) Caregiver and patient anonymity and prevention against traceability: Our scheme protects both caregiver and patient identities. Also, dynamic identities are introduced to provide anonymity and untraceability of mutual authentication as these identities cannot be retrieved by adversaries without knowing secret random numbers. They are also updated in each round of authentication.
(4) Lightweight authentication: Despite its use of ECC, our protocol is lightweight enough to be suitable for deployment on medical devices that have larger resource capacity in comparison to similar ECC based schemes that claim deployment on lightweight sensors which is unrealistic.

(5) Biometric privacy protection through the use of fuzzy extraction: The generator function takes in a biometric credential and produces a secret string and a public accessory. The reproduction function takes in a noisy biometric credential and the public accessory to produce the right secret string if and only if the level of noise in the biometric template is less than a set threshold [11].
(6) Security: We formally validate that our protocol establishes a shared session key and achieves mutual authentication using BAN logic. Simulation results based on AVISPA also prove that our protocol can resist replay attack and man-in-the-middle attack.
(7) Efficiency: We compare the performance of our scheme relative to others and show that our protocol has better performance (e.g., communication overhead, computational overhead).

3.3. Proposed Scheme

Our scheme consists of five phases described in this section. These include doctor/nurse and sensor (ECG monitor) registration, doctor/nurse login, authentication and key agreement, continuous monitoring and the password change phase. The notations used throughout our scheme are presented in Table 1 below. Please note that in the remaining sections of this chapter, we use “sensor” to mean the ECG device to which the lightweight sensors are attached.

Table 1 - Notation Definitions

Parameter	Meanings
U_i, S_j	User *i* (could be a nurse, or a doctor,…) and 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)
χ	Secret key of the trusted server
$Gen(\bullet), Rep(\bullet)$	Fuzzy extraction function
SID_j	S_j's real identity
PID	Patient's real identity
$FeatureSet_{ECG}$	ECG feature set that has been preprocessed and auto-correlated using techniques like [64]
TID_i	User *i*'s dynamic identity
X	A secrect value known by U_i and *TS*
X_{S_i}	A secrect value known by U_i and *TS*
X_{S_j}	A secrect value known by S_j and *TS*
$SK_{i,j}$	A secret session key for data communications between user *i* and the sensor *j*

3.3.1. Doctor/Nurse Registration Phase

Through the registration phase (shown in Figure 2), 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 $(R_i, P_i) = Gen(BIO_i)$ and $HPW_i = h(PW_i \| R_i)$. He/she then sends $\{ID_i, HPW_i, R_i\}$ to the trusted server.

Step 2: The trusted server chooses a random number r_0 and uses its secret key x to compute $X_{S_i} = h(ID_i \| x \| r_0)$, $A_i = X_{S_i} \oplus HPW_i$ 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 $\Upsilon = x\mathrm{P}$. It then chooses a third random number r_2 to compute the dynamic identity $TID_i = h(r_2 \| ID_i)$. The trusted server stores $\{ID_i, h(TID_i \| x), r_0 \oplus x, Honey\_List\}$ 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 $SC = \{A_i, B_i, C_i, \Upsilon, \mathrm{P}, P_i, TID_i\}$.

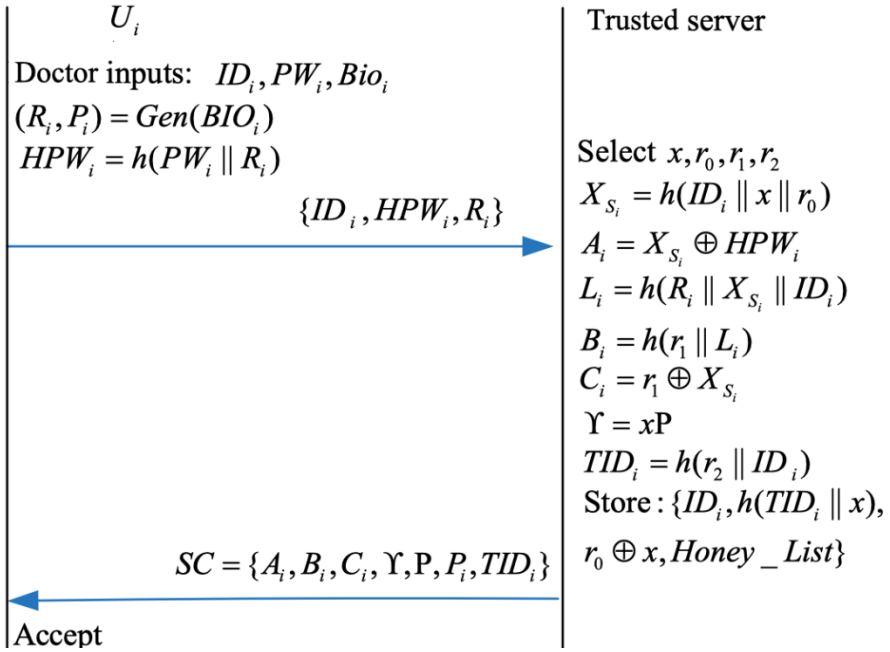


Figure 2 - Doctor/Nurse Registration Phase

3.3.2. Sensor Registration Phase

Similarly, communications between the sensor and the trusted server take place over a secure channel as it is a one-time process (shown in Figure 3). 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 [64] to generate a $FeatureSet_{ECG}$. 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_j to the trusted server.

Step 3: The trusted server uses *PID* and $FeatureSet_{ECG}$ as training data for the convolutional neural network inside the server as is used in [10]. The server also stores $\{h(SID_j \| x), PID \oplus x\}$ in its database and computes $C_0 = h(x)$. It then sends $\{C_0\}$ to the sensor.

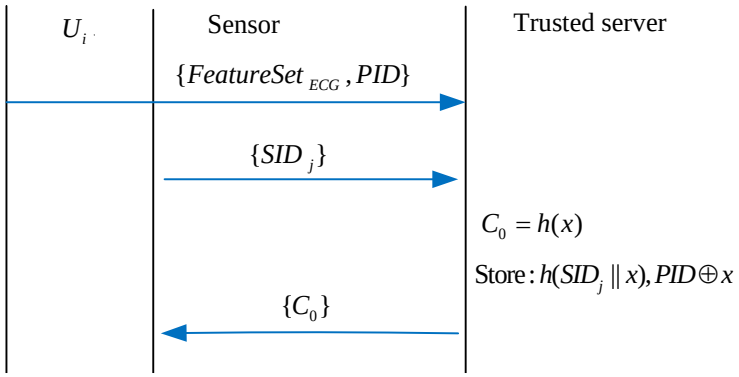


Figure 3 - Sensor Registration Phase

3.3.3. Doctor/Nurse Login Phase

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 BIO_i.

Step 2: The smart card computes $R_i' = Rep(BIO_i, P_i)$, $X_{S_i}' = A_i \oplus h(PW_i \| R_i')$, $r_1' = C_i \oplus h(X_{S_i}')$, $L_i' = h(R_i' \| X_{S_i}' \| ID_i)$ and $B_i' = h(r_1' \| L_i')$. It then checks if $B_i' \stackrel{?}{=} B_i$. If they are equal, it selects a random number $\alpha \in Z_p^*$ and computes $X_i = \alpha P$, $X = \alpha \Upsilon$, $DID_i = TID_i \oplus h(X_i \| X)$ and $M_{U_i,G} = h(X_{S_i}' \| X_i \| X)$. Next, the smart card terminal sends $\{DID_i, X_i, M_{U_i,G}\}$ to the trusted server.

3.3.4. Authentication and Key Agreement Phase

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 (shown in Figure 4), 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 $TID_i' = DID_i \oplus h(X_i \| X')$. It then uses $h(TID_i' \| x)$ to lookup the corresponding r_0, ID_i and *Honey_List* from its database. If *Honey_List* > threshold, the server thinks the smart card has been suspended and rejects the request. If it could not find an entry for $h(TID_i' \| x)$ in its database, it also rejects the login request. Otherwise, it proceeds to compute $X_{S_i}' = h(ID_i \| x \| r_0)$. If $M_{U_i,G} \neq h(X_{S_i}' \| X_i \| X')$, then the server increments the value in the *Honey_List* by 1. Otherwise, it looks up *PID* using $h(SID_j \| x)$ and chooses $\beta \in Z_p^*$

to compute $Y_j = \beta \mathrm{P}$, $X_{S_j} = h(C_0 \| SID_j \| PID)$, $M_{G,S_j} = h(X_{S_j} \| X_i \| SID_j \| T_G)$, and $X_\beta = \beta \oplus X_{S_j}$. The server sends $\{X_i, M_{G,S_j}, X_b, Y_j, T_g\}$ to the sensor, where T_G 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_G and computes $X'_{S_j} = h(C_0 \| PID \| SID_j)$. If $M'_{G,S_j} \neq h(X'_{S_j} \| X_i \| SID_j \| T_G)$, it rejects it. Otherwise, the sensor computes $\beta' = X_\beta \oplus X'_{S_j}$, $k_j = h(X'_{S_j} \| T_j)$, $M_{S_j,G} = h(k_j \| Y_j \| X_{S_j} \| X_i \| T_j \| PID)$, $Z = \beta' X_i$ and $SK_{i,j} = h(Z \| X_i \| Y_j)$. After that, it sends $\{M_{S_j,G}, T_j, E[FeatureSet_{ECG}]_{SK_{i,j}}\}$ 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 $k'_j = h(X_{S_j} \| T_j)$, $Z' = \beta X_i$ and $SK'_{i,j} = h(X_i \| Z' \| Y_j)$. It then decrypts $E[FeatureSet_{ECG}]_{SK'_{i,j}}$ and gets $FeatureSet_{ECG}$. The neural network uses $FeatureSet_{ECG}$ to find PID'. Next, the server checks if $M_{S_j,G} \stackrel{?}{=} h(k'_j \| Y_j \| X_{S_j} \| X_i \| T_j \| PID')$. 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 $M_{G,U_i} = h(X'_{S_i} \| \alpha Y_j \| \alpha \Upsilon \| TID_i)$, $TID_i^{new} = h(r_3 \| ID_i)$ and updates the database entry to $h(TID_i^{new} \| x)$. The server sends $\{Y_j, M_{G,U_i}, r_3\}$ to the user.

Step 4: The user checks if $M_{G,U_i} \stackrel{?}{=} h(X'_{S_i} \| \alpha Y_j \| \alpha \Upsilon \| TID_i)$. If they are not equal, authentication fails. Otherwise, it computes $SK'_{i,j} = h(X_i \| \alpha Y_j \| Y_j)$ and updates $TID_i = h(r_3 \| ID_i)$ in its smart card.

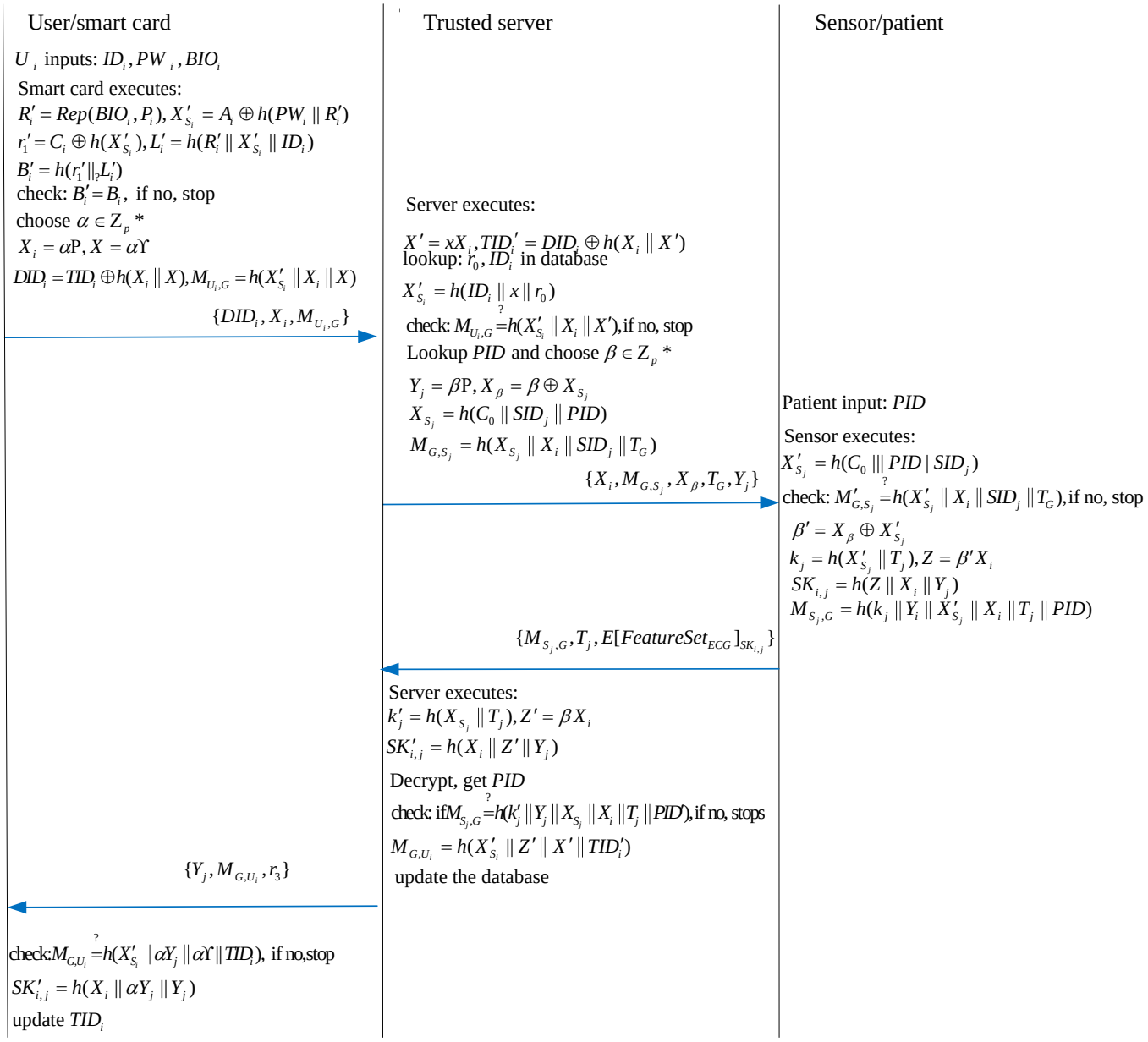


Figure 4 - Login and Authentication Phase

3.3.5. Continuous Real-time Monitoring Phase

Once the one-time authentication has completed successfully, the sensor continuously monitors the patient's heart rate and sends encrypted ECG signals to the doctor in real-time. This data is always encrypted using the established session key $SK_{i,j}$ before being sent to the trusted server or doctor to ensure privacy and secrecy of patient data is maintained. Every 30 minutes, the sensor sends data to the trusted server which had previously cached the patient's *PID* for the current session. The trusted server's neural network uses these signals to compute PID and verify that the computed *PID* matches the cached one. If they match, it means the patient is still actually the correct patient registered and the session continues. If they don't match, the trusted server sends a warning message to the doctor to indicate that the patient might have changed. In response, the doctor may or may not choose to terminate the session.

3.3.6. Password0 Change Phase

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 BIO_i.

Step 2: The smart card computes $R_i^* = Rep(BIO_i, P_i)$, $X_{S_i}^* = A_i \oplus h(PW_i \| R_i^*)$, $r_1^* = C_i \oplus h(X_{S_i}^*)$, $L_i^* = h(R_i^* \| X_{S_i}^* \| ID_i)$ and $B_i^* = h(r_1^* \| L_i^*)$. It then checks if $B_i^* \stackrel{?}{=} B_i$. 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_i^{new}.

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

3.4. Formal Security Analysis – Formal Proof of Authentication and Key Agreement using BAN Logic

The vast majority of authentication schemes, including Wang et al. [1] and Park et al. [4] are formally proven to be secure through the use of the BAN logic. BAN logic [65] is a simple way to analyze the design logic and security of a protocol in an efficient manner. Table 2 below shows the notation used to describe the protocol logic as well as the BAN logic postulates [65].

Table 2 - BAN Logic Notation and Postulates

Notation / Postulate	**Meaning**
$A \mid\equiv B$	*A* believes *B*
$A \overset{K}{\leftrightarrow} B$	Key *K* is a shared key between *A* and *B*
$\#(B)$	Fresh *B* (generated in current session)
$A \lhd B$	*A* sees *B*
$A \mid\Rightarrow B$	*A* has jurisdiction over *B* (*A* can generate or compute *B*)
$A \mid\sim B$	*A* once said *B*
$(A)_K$	*A* is hashed using key *K*
$\{A\}_K$	*A* is encrypted using key *K*
$\frac{A \mid\equiv \#(X), A \mid\equiv B \mid\sim X}{A \mid\equiv B \mid\equiv X}$	Nonce Verification Rule
$\frac{A \mid\equiv B \mid\overset{K}{\leftrightarrow} X, A \lhd \{X\}_K}{A \mid\equiv B \mid\sim X}$	Message Meaning Rule
$\frac{A \mid\equiv B \mid\Rightarrow X, A \mid\equiv B \mid\equiv X}{A \mid\equiv X}$	Jurisdiction Rule
$\frac{A \mid\equiv \#(X)}{A \mid\equiv \#(X,Y)}$	Freshness Conjugation Rule

As such, we used BAN logic to formally prove the security of our scheme. The proposed scheme should satisfy the following security goals for the session key $SK_{i,j}$:

(1) G1: $U_i \mid\equiv S_j \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$

(2) G2: $S_j \mid\equiv U_i \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$

(3) G3: $U_i \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$

(4) G4: $S_j \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$

The following represents the idealized version of our scheme:

M1: $U_i \rightarrow TS : (DID_i, X_i, U_i \overset{X}{\leftrightarrow} TS)_{X_{S_i}}$

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: $TS \rightarrow U_i : (\alpha Y_j, X, TID_i)_{X_{S_i}}$

The following defines the initial assumptions we made about the state of the scheme:

$H1: S_j \mid\equiv \#(T_G)$

$H2: TS \mid\equiv \#(X)$

$H3: TS \mid\equiv \#(T_j)$

$H4: U_i \mid\equiv \#(X)$

$H5: TS \mid\equiv U_i \overset{X_{S_i}}{\leftrightarrow} TS$

$H6: TS \mid\equiv S_j \overset{X_{S_j}}{\leftrightarrow} TS$

$H7: S_j \mid\equiv S_j \overset{X_{S_j}}{\leftrightarrow} TS$

$H8: U_i \mid\equiv U_i \overset{X_{S_i}}{\leftrightarrow} TS$

$H9: U_i \mid\equiv S_j \mid\Rightarrow U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$

$H10: S_j \mid\equiv U_i \mid\Rightarrow U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$

Formal security analysis of the idealized scheme is as follows:

From M1, we get S1: $TS \lhd (DID_i, X_i, U_i \overset{X}{\leftrightarrow} TS)_{X_{S_i}}$. Using *H5* and S1, we can apply the message meaning rule:

$$\frac{TS \mid\equiv U_i \overset{X_{S_i}}{\leftrightarrow} TS \ , \ TS \lhd (DID_i, \ X_i, \ U_i \overset{X}{\leftrightarrow} TS)_{X_{S_i}}}{TS \mid\equiv U_i \mid\sim (DID_i, \ X_i, \ U_i \overset{X}{\leftrightarrow} TS)}$$

Then we can get S2: $TS \mid\equiv U_i \mid\sim (DID_i, \ X_i, \ U_i \overset{X}{\leftrightarrow} TS)$. Using *H2* and M1, we can apply the freshness conjugation rule：

$$\frac{TS \mid\equiv \#(X)}{TS \mid\equiv \#(DID_i, \ X_i, \ U_i \overset{X}{\leftrightarrow} TS)}$$

Thus, we can get S3: $TS \mid\equiv \#(DID_i, \ X_i, \ U_i \overset{X}{\leftrightarrow} TS)$. Using S3 and S2, we can apply the nonce verification rule:

$$\frac{TS \mid\equiv \#(DID_i,\ X_i,\ U_i \overset{X}{\leftrightarrow} TS), TS \mid\equiv U_i \mid\sim (DID_i,\ X_i,\ U_i \overset{X}{\leftrightarrow} TS)}{TS \mid\equiv U_i \mid\equiv (DID_i,\ X_i,\ U_i \overset{X}{\leftrightarrow} TS)}$$

Thus, we have S4: $TS \mid\equiv U_i \mid\equiv (DID_i,\ X_i,\ U_i \overset{X}{\leftrightarrow} TS)$.

From M2, we get S5: $S_j \lhd (X_i, SID_j, T_G)_{X_{S_j}}$ .Using *H7* and S5, we can apply the message meaning rule:

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

Similar, using *H1* and M2, we can apply the freshness conjugation rule:

$$\frac{\mathrm{S}_j \mid\equiv \#(T_G)}{\mathrm{S}_j \mid\equiv \#(X_i, SID_j, T_G)}$$

Then, we apply the nonce verification rule:

$$\frac{\mathrm{S}_j \mid\equiv \#(X_i, SID_j, T_G), S_j \mid\equiv TS \mid\sim (X_i, SID_j, T_G)}{S_j \mid\equiv TS \mid\equiv (X_i, SID_j, T_G)}$$

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

From M3, we get S6: $TS \lhd (k_j, Y_j, T_j, PID)_{X_{S_j}}$ .Using *H6* and S6, we can apply the message meaning rule

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

By using H3, the freshness conjugation rule and the nonce verification rule, we have $TS \mid\equiv S_j \mid\equiv (k_j, Y_j, T_j, PID)$.

Similarly, from M4 and using *H8* and *H4*, the freshness conjugation rule and the nonce verification rule, we have $U_i \mid\equiv TS \mid\equiv (\alpha Y_j, X, TID_i)$.

Since $SK_{i,j} = h(X_i \| \alpha Y_j \| Y_j)$ and given $S_j \mid\equiv TS \mid\equiv (X_i, SID_j, T_G)$ and $TS \mid\equiv S_j \mid\equiv (k_j, Y_j, T_j, PID)$, we have $U_i \mid\equiv S_j \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$. Therefore, we have achieved goal G1. Similarly, we have $S_j \mid\equiv U_i \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$, and that achieves goal G2.

Using H9 and $U_i \mid\equiv S_j \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$ and the jurisdiction rule, we have $U_i \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$. Thus, it achieved goal G3.

Using *H10* and $S_j \mid\equiv U_i \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$ and the jurisdiction rule, we have $S_j \mid\equiv U_i \overset{SK_{i,j}}{\leftrightarrow} S_j$. 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 $SK_{i,j}$.

3.5. Informal Security Analysis

Before analyzing the security of our scheme, we first present the adversary model. We assume that any adversary has the following capabilities when accessing our BSN:

- An adversary can conduct power analysis attack to obtain the information stored in the smart card [66].
- An adversary can intercept all messages transmitted over public channels.
- An adversary can modify, delete and replay all messages transmitted over public channels.
- An adversary can attack and obtain all information stored in the sensor node since sensor nodes are deployed in unprotected environments.
- An adversary can obtain user fingerprint through the use of putty and gelatin or a high-quality scanner [66].

(1) Mutual authentication

The trusted server confirms the user identity and vice versa by comparing the received and computed $M_{U_i,G}$ and M_{G,U_i} respectively. Similarly, the trusted server confirms the identity of the sensor and vice versa by comparing the received and computed $M_{S_j,G}$ and M_{G,S_j} respectively. Therefore, our scheme achieves mutual authentication.

(2) User (doctor/nurse) impersonation attack

An adversary with access to a smart card and its contents (through power analysis attack) cannot compute $M_{U_i,G} = h(X'_{S_i} \| X_i \| X)$ because he/she cannot compute $X'_{S_i} = A_i \oplus h(PW_i \| R'_i)$ since this requires access to the user's R'_i and PW_i. R'_i and PW_i are never stored in the server database, made accessible to the sensor nor sent over the public channel without being hashed $h(PW_i \| R'_i)$. This makes both computationally infeasible to get.

(3) Sensor impersonation attack

An adversary trying to impersonate the sensor will not be able to do so as he/she cannot compute $M_{S_j,G} = h(k_j \| Y_j \| X_{S_j} \| X_i \| T_j \| PID)$ nor can he/she compute the session key. This is because X_{S_j} cannot be computed without knowing both C_0 and SID_j which are never exposed over the public channel.

(4) Sensor theft attack

An attacker with access to C_0 and SID_j in the sensor will still not be able to compute $M_{S_j,G} = h(k_j \| Y_j \| X_{S_j} \| X_i \| T_j \| PID)$ because he/she cannot compute X_{S_j} and Y_j which require PID and β. Similarly, the attacker cannot construct a valid $SK_{i,j}$. Even with access to the trusted server database, the attacker will need *x* to get *PID*. This is an improvement over Wang et al. [1] where once the sensor is compromised, carrying out an impersonation attack and computing the session key is simple.

(5) Patient impersonation attack

The patient's PID is never stored as plain text so even if an attacker is able to access the trusted server database, it is not possible for him/her to get PID without knowing x since the server stores $\{h(SID_j \| x), PID \oplus x\}$. Moreover, PID is never sent over the public channel without being hashed. Furthermore, our use of continuous monitoring prevents this as seen in the sensor theft section.

(6) Physical sensor theft

Our scheme accounts for and has a protection mechanism for cases where an attacker steals the sensor and hooks it up to a different patient. It does this by 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.

(7) Lost/stolen smart card attack

An adversary with access to a smart card and obtaining its contents $\{A_i, B_i, C_i, \Upsilon, \mathrm{P}, P_i, TID_i\}$ cannot construct a valid login message $\{DID_i, X_i, M_{U_i,G}\}$ because it is hard to get X_{S_i} without knowing user's R_i and PW_i or the random numbers r_0 and *x*. All these parameters are never sent over the public channel without being hashed nor are they stored in the smart card as part of the computation of another parameter without being hashed making it computationally infeasible to compute them. Therefore, our scheme resists lost/stolen smart card attack.

(8) Replay attack

As mentioned above, an adversary with access to a smart card and its contents and who is eavesdropping over the public channel cannot construct a valid login message. If he/she tries to replay a valid login message, the trusted server will compute $TID_i' = DID_i \oplus h(X_i \,||\, X')$ and then lookup $h(TID_i' \,||\, x)$ in its database. Since TID_i is a dynamic identity that is recomputed with a new randomly generated number each time the user is successfully authenticated, the database is also updated with the new one so will not be able to find the old one. Therefore, the login request will be rejected. Also, an attacker replaying messages between the trusted server and the sensor will be detected due to the timestamp staleness.

(9) Caregiver and Patient anonymity and protection against traceability

An attacker monitoring the public channel will not be capable of determining which caregiver the messages correspond to. This is accomplished through the use of the randomness of DID_i since X_i and *X* are also randomly generated per login session. Consequently, it is computationally infeasible for an attacker to determine which ID_i the transmitted DID_i belongs to. It must also be noted that all messages transmitted in the public channel satisfy a freshness requirement meaning that it is very difficult for an attacker to know if two messages belong to the same ID_i so the user is truly untraceable.

As for the patient, PID is never exposed at all over the public channel without being hashed so cannot even be computed providing patient anonymity. We also protect against patient traceability because $M_{S_j,G} = h(k_j \| Y_j \| X_{S_j} \| X_i \| T_j \| PID)$ which is exposed over the public channel and includes PID is different every session even for the same patient. This is because α and β are randomly generated in every session and therefore X_i and Y_j are different in every session.

(10) Offline password guessing attack

Suppose that the attacker obtains the information $\{A_i, B_i, C_i, \Upsilon, \mathrm{P}, P_i, TID_i\}$ stored in the smart card of a legal user. The attacker cannot guess the correct password 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.

(11) 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 β. However, it is hard to get β due to the discrete logarithm problem. Even if the attacker tries to compute $\beta = X_\beta \oplus X_{S_j}$, it is still quite difficult as X_{S_j} is protected by a one-way hash function.

(12) Known-key security

The session key in our scheme is computed as $SK_{i,j} = h(Z \| X_i \| Y_j)$. α and β are randomly generated in every session and therefore X_i and Y_j are also different in every session. This makes every component of the session key unique in every session. As a result, our scheme provides known-key security since a compromised session key will not help an attacker find another session key.

(13) Protection of biometric template

Our scheme does not transmit biometric templates over a public channel without being protected. The user's biometric template BIO_i is protected by fuzzy extraction function, and the biometric key R_i is protected by one-way hash function. Therefore, it is infeasible to access the template. In addition, the patient's ECG signals are always encrypted before being sent over the public channel. As previously discussed, it is computationally infeasible for an intruder to compute the session key and therefore, he/she cannot decrypt the encrypted ECG signals.

3.6. Simulation and Formal Security Verification using AVISPA

3.6.1. Introduction to AVISPA Simulation 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 [67] [68] [69] [70]. Protocol schemes and their security properties are defined using a High-Level Protocol Specification Language (HLPSL).

HLPSL is a role-based language. This means that each participant's actions are defined within a module referred to as the basic role. Each role has parameters passed in, an initial state and transitions defining a trigger and action pair that changes the state; for example, sending or receiving messages.

Multiple basic roles are then instantiated in a composed role and interactions between the instances are defined [69]. This composed role is referred to as "session" by convention and it also defines the channels used to send and receive the messages between the participants. We define the intruder model a channel uses by passing an additional parameter "channel (dy)". In this case, "dy" refers to the Dolev-Yao intruder model [70] which gives intruders full control over the channel including impersonating any participant to send messages to other participants as well as the ability to analyze, intercept and modify any message sent over the channel [69].

Figure 5 below depicts the architectural structure of the AVISPA tool which 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 [69].

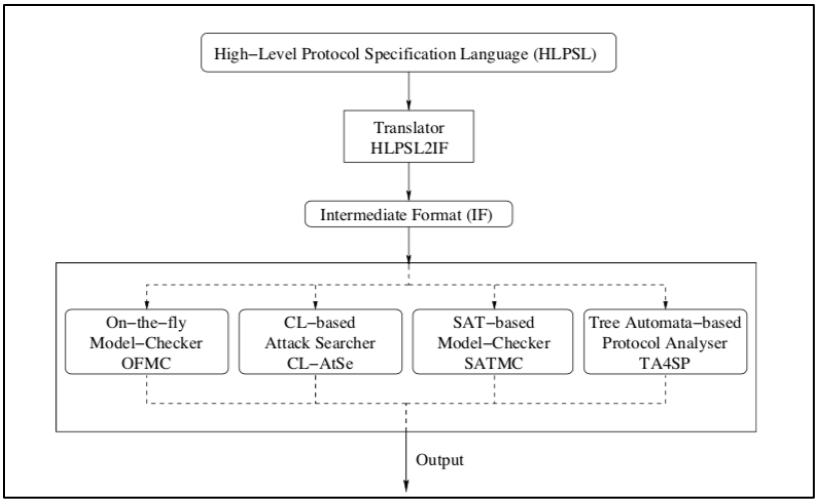


Figure 5 - AVISPA Architectural Structure

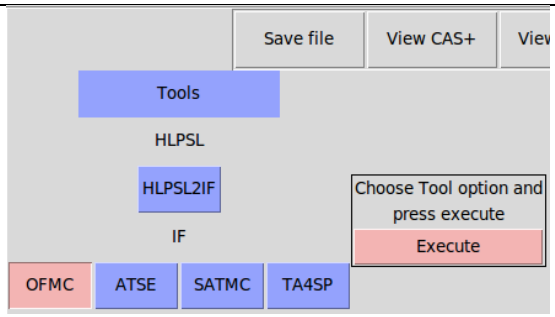
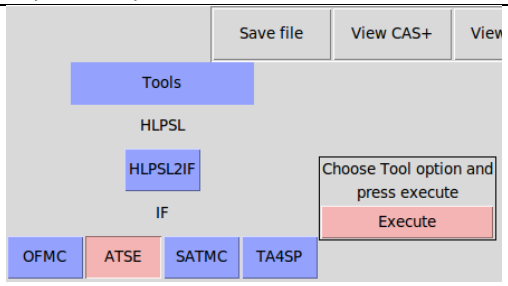
AVISPA is a widely accepted tool and used by many papers including [1] [17]. For the purposes of this work, we used AVISPA with the Security Protocol Animator (SPAN) tool which assists with writing the HLPSL definitions as well as simulating them to generate message sequence charts [68]. We chose to verify the security of our scheme using OFMC and CL-AtSe because they both account for the algebraic properties of the XOR operation which our scheme utilizes [71]. The HLPSL definition we wrote can be seen in the Appendix.

3.6.2. Simulation Results

Running the simulation against the different verification back-ends integrated in AVISPA generates different output results. Each result consists of (1) a summary which determines whether the protocol is safe, unsafe or inconclusive (2) the details section which describes the environment in which the protocol's safeness or lack of it is claimed (3) the name of the protocol under analysis (4) the security goals outlined in the protocol and finally, (5) the backend verifier responsible for these results [11].

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 the table below.

Table 3 - AVISPA Simulation Results

Backend Verifier	Security Analysis Output
OFMC	 % OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/myProtocol_works_auth_final.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 16.13s visitedNodes: 0 nodes depth: 1000000 plies
CL-AtSe	 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/myProtocol_works_auth_final.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 3591 states Reachable : 3591 states Translation: 0.03 seconds Computation: 0.74 seconds

3.7. Performance Analysis

In this section, we compare the performance and computational complexity of our protocol with similar protocols. It must be noted that our protocol is the first end-to-end user to sensor protocol that also does continuous monitoring of the patient so we tried to compare it to protocols that do individual portions. [1], [4] and [10] are user-sensor three-factor authentication schemes whereas [60] is a continuous authentication scheme between a sensor and gateway node.

In Table 4, we compare the security properties of our protocol with aforementioned schemes and observe that none of them satisfy all 13 security requirements. In contrast, our protocol satisfies all 13 security requirements in addition to providing continuous patient monitoring.

Table 4 - Security Comparisons

*: if smart card lost

**: Continuous authentication schemes in literature are applied between the sensor and gateway nodes only. Instead of this increase in overhead, our scheme is the first to propose a complete caregiver to patient scheme with continuous monitoring on the sensor side to verify patient identity.

***: referring to sensor identity

****: access to 2 of the 3 login factors results in this attack

Properties	**Our protocol**	[10]	[4]	[1]	[60]**
Mutual authentication	✓	✓	✓	✓	✓
User impersonation attack	✓	×	✓	×*	N/A
Sensor impersonation attack	✓	×	×	×	×
Physical sensor theft	✓	×	×	×	✓
Lost smart card attack	✓	✓	✓	×****	N/A
Replay attack	✓	✓	✓	×	✓
User anonymity	✓	×	×	✓	×***
User traceability	✓	×	×	✓	×***
Offline password guessing attack	✓	×	×	✓	N/A
Symmetric key protection	✓	×	✓	✓	✓
Known-key security	✓	×	✓	✓	✓
Use biometrics protection	✓	✓	✓	×*	×
Continuous Authentication/Monitoring	✓**	×	×	×	✓

✓: The requirement is satisfied

×: The requirement is not satisfied

3.7.1. Computational Overhead Comparison

Table 5 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. [11], 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 ≈0.5, T_F ≈0.5, T_E ≈50.3 and T_s ≈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 [11]. 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 $22T_h + 8T_E + 2T_F + 2T_S$ in our protocol, $21T_h$ in [10], $25T_h + 4T_E + T_F$ in [4] and $29T_h + 6T_E + T_F$ in [1]. Although [10] 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 - Comparisons of Computational Overhead for Smartcards

Properties	**Login phase**	**Authentication phase**	**Password change phase**
Our protocol	$6T_H + 2T_E + T_F$ ≈104.1ms	$2T_h + 2T_E$ ≈101.6ms	$7T_h + T_F$ ≈4ms
[10]	$7T_h$ ≈3.5ms	$3T_H$ ≈1.5ms	$10T_h$ ≈5ms
[4]	$6T_h + T_E + T_F$ ≈53.8ms	$4T_H + T_E$ ≈52.3ms	$8T_h + T_F$ ≈4.5ms
[1]	$8T_h + T_F + 2T_E$ ≈105.1ms	$2T_H + T_E$ ≈51.3ms	$10T_h + T_F$ 5.5ms
[60]	N/A	N/A	N/A

Table 6 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 [4]. Remember that this protocol is meant to be deployed to the more resourceful ECG monitors rather than the lightweight sensors themselves so this makes it much easier to deploy the protocol in comparison to the more computationally complex schemes that cannot realistically be deployed on sensors as was claimed.

Table 6 - Comparison of Computational Overhead for Sensors

Phase	**Our protocol**	[10]	[4]	[1]	[60]
Static phase	$5T_H + T_E + T_S$ ≈61.5ms	$3T_H$ ≈1.5ms	$4T_H + 2T_E$ ≈102.6ms	$6T_H + 2T_E$ ≈103.6ms	$20T_h$ ≈10ms
Continuous authentication / monitoring phase	T_S ≈8.7ms	N/A	N/A	N/A	$10T_h$ ≈5ms

Table 7 lists the computational overhead incurred by the trusted server. It shows that our protocol has the largest overhead but it must be noted that offloading the sensor greatly enhances the efficiency and practicality of the protocol and this can be done at the price of achieving higher security and lower sensor overhead.

Table 7 - Comparison of Computational Overhead for Trusted Server

Phase	**Our protocol**	[10]	[4]	[1]	[60]
Authentication phase	$9T_H + 3T_E + T_S$ ≈164.1ms	$8T_H$ ≈4ms	$11T_H$ ≈5.5ms	$13T_H + T_E$ ≈56.8ms	$10T_H$ ≈5ms

3.7.2. Communication Overhead Comparison

Table 8 below shows a comparison of the communication overhead incurred by our protocol in comparison to that in [10] [4] [1] [60]. The output of the one-way hash function and bio-hash function, real identity and any random integer is 160bits long. The length of the output of the symmetric encryption/decryption is 256bits. We observe that our protocol is competitively amongst the most efficient especially given that [60] is not a complete end-to-end user to sensor protocol and its communication overhead calculation is only based on the static authentication portion.

Table 8 - Comparison of Communication Overhead

*: Static authentication only			
Properties	**Login phase**	**Authentication phase**	**Total**
Our protocol	320bits	1536bits	1856bits
[10]	640bits	1440bits	2080bits
[4]	480bits	1280bits	1760bits
[1]	480bits	1120bits	1600bits
[60] *	-	1600bits	1600bits

Chapter 4: BSN Authentication based on Chebyshev Chaotic Maps

4.1. Introduction & Motivation

This chapter addresses the second application of medical sensor networks presented in this thesis. This scenario uses BSN to enable readings from lightweight sensors connected to patients to be sent and stored on a trusted server (see Figure 6 below). Unlike the scenario presented in the previous chapter, patients are assumed to be dynamic or in motion meaning that the lightweight limited-resource wearable sensors are attached to their bodies and their vital signs are being tracked as they continue to live actively outside the hospital setting. Sensors collect and send, through Bluetooth, encrypted readings to a mobile device with SDP installed. This SDP then forwards the encrypted readings to a trusted server that can then be accessed by a caregiver but this is outside the scope of this work.

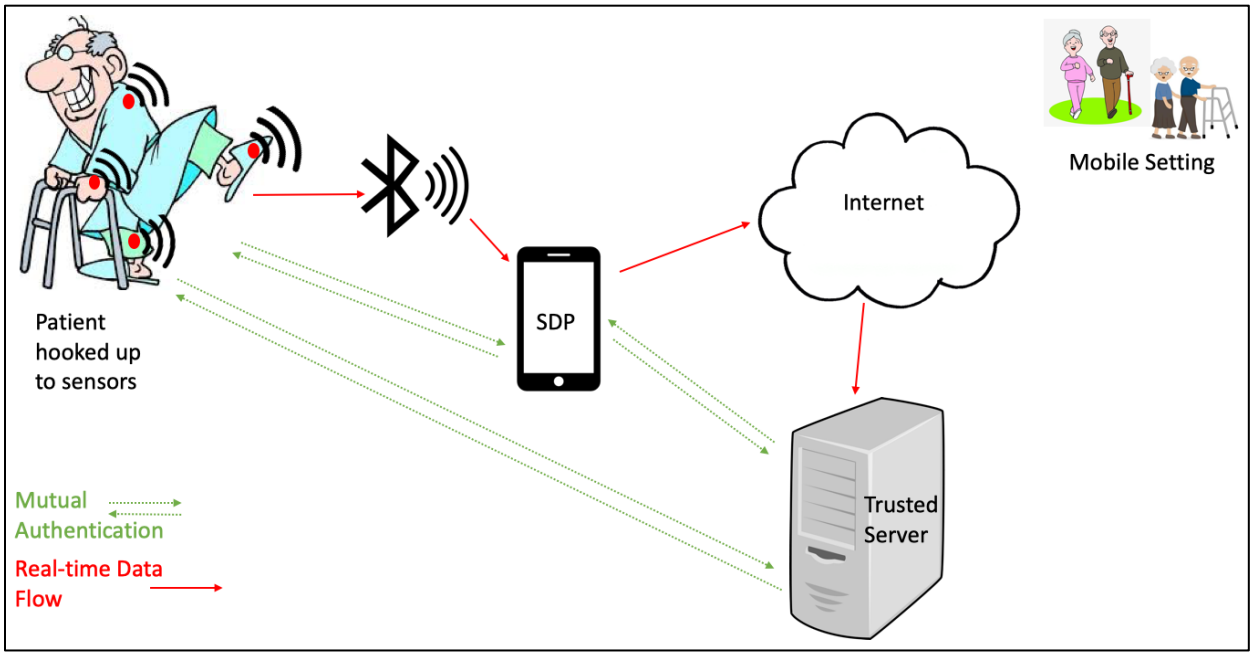


Figure 6 - Applying Chaotic Map Based Scheme on BSN

Again, the public nature of the BSN in which the sensors are deployed means that they are hacker prone so an authentication and key agreement scheme must be established to protect patient security and privacy. The added constraint in this case is the fact that the scheme used must be very lightweight so as to be suitable for deployment on the lightweight sensors themselves. State-of-the-art schemes applying public-key mechanisms (e.g., Elliptic Curve Cryptography (ECC)) to construct a key agreement scheme are not suitable for this application due to the high complexity operations used such as scalar multiplication and the point addition of an elliptic curve. This is because bio-sensors in the BSN system, especially the wearable devices and implanted smart chips, are resource-limited and cannot afford a large amount of computation and communication overhead.

Moreover, a lot of these authentication schemes cannot provide user anonymity nor protect against user traceability resulting in patients having to face privacy disclosure risks when they login to share their information. Furthermore, many schemes do not take any measures to protect the biometric templates which is a huge problem since unlike passwords, biometric templates cannot be changed. In addition to that, other schemes employ the hash function and XOR operations [37] [31]; however, fail to guarantee that communications between bio-sensors and users are secure because bio-sensors are not authenticated before accessing the system. This means that an attacker can impersonate the sensor and send incorrect data to the doctor resulting in a wrong diagnosis.

Furthermore, schemes such as those listed in [40] [44] [39] [42] mentioned earlier cannot be used for this application of BSN because they establish a single session key between a bio-sensor and a server. The mobile device, which needs to know about the sensed data in order to alert the patient wearing the sensor, is not authenticated resulting in a security hole. Schemes such as Yeh et al. [38] which do validate the bio-sensor, user and server incur heavy computational cost so are still not optimal for this application either.

This chapter describes a performance efficient and anonymous authentication protocol utilizing Chebyshev chaotic maps and without residing to the use of complex operations (e.g., scalar multiplication operations, pairing operations) and with the aim of establishing two session keys for secure communication in BSN systems.

4.2. Chapter Contributions

To achieve privacy-preserving in designing an efficient key agreement scheme, we propose a practical key agreement scheme, which can meet the following security requirements and satisfy the computational demands. The main contributions of our work are as follows:

(1) Lightweight authentication - our protocol is very lightweight since it does not use any complex operations making it suitable for deployment on lightweight wireless sensors allowing for its use in a more flexible environment where patients are on the move.
(2) Our protocol establishes two session keys for enhanced security: one between the sensor and SDP and the second between the SDP and trusted server. This eliminates the security hole at the SDP level that mot schemes that establish a single session key between sensor and server encounter.
(3) Patient/sensor anonymity and preventing traceability: Dynamic identities are introduced to provide anonymity and prevent user traceability. These identities cannot be retrieved by adversaries without knowing the secret random numbers which are generated and updated in each round of authentication.
(4) Biometric privacy protection: To protect the patient's biometric information, a bio-hash function is utilized and verified by the server when checking related values. The bio-hash function makes it computationally infeasible for an attacker to extract the biometric template Bio given H(Bio). Moreover, a random integer is combined with the output of a bio-hash function using XOR operation.

(5) Efficiency: Our protocol demonstrates better performance (i.e. communication and computational overhead) in comparison to similar schemes without using any complex operations such as the pairing or scalar multiplication operations during the registration, mutual authentication and password change phases.

(6) Security: Our protocol provides a means of achieving mutual authentication among the bio-sensor, patient and server in addition to providing two session keys for secure data communications. Formal and informal security analysis illustrate that session keys satisfy the requirements of session key security and know-key security. In addition, the protocol provides user anonymity/untraceability and resists sensor theft attack, mobile device theft attack, impersonation attack and privileged insider attacks. Also, simulation and formal security verification using AVISPA proves that our protocol resists replay attack and man-in-middle attack.

4.3. Proposed Scheme

Our protocol requires that all bio-sensors and the SDP register with the BSNS in advance. After registration, security credentials are shared among bio-sensors, SDP and the BSNS. Like all three-factor authentication schemes, the patient enters his/her identity, password and biometric. Then, mutual authentication occurs among the bio-sensors, SDP and BSNS as well in order to ensure that the bio-sensor belongs to the corresponding patient and prevent sensor theft attack and mobile device (SDP) theft attack. Different from [44] that has no session key between the bio-sensor and SDP, our protocol establishes a session key between the bio-sensor and SDP allowing the SDP to provide an immediate alert to the patient according to received data from the bio-sensor. Moreover, unlike existing protocols including those in [72] [22], we utilize dynamic identities and a dynamic master key x_b to provide anonymity and prevent user traceability.

Before we dive into the protocol details, we need to state our assumptions and define some notations in Table 9.

Table 9 - Notation Definitions

Parameter	Meanings
S_{ii}	Sensor *ii* monitoring patient *i*'s physiological signs
SDP_i	Sensor data provider (SDP) *i* that is a software running on a mobile device (e.g., cellphone) belonging to patient *i*.
$BSNS$	Trusted BSN server
IDS_{ii}	Identity of S_{ii}
ID_i , PW_i	Patient *i*'s real identity and real password
MID_i, MPW_i	SDP_i 's masked identity and masked password
x_s	Master key between $BSNS$ and SDP_i
x_b	A temporary master key between *BSNS* and S_{ii}, should be updated for each round of authentication
DID_i	SDP_i's dynamic identity
$DIDS_{ii}$	S_{ii} 's dynamic identity
Z_i	SDP_i's masked biometric information
$h(.)$	A collision free hash function
$h_B(.)$	A secure bio-hash function
C_j	The jth transmitted value in our work
sk_{ss}	A secret session key for communications between SDP_i and Sensor *ii*
sk_{sb}	A secret session key for communications between SDP_i and $BSNS$
q_s	Number of Send queries
q_h	Number of hash queries
q_c	Number of the attacker's guessing attempt to S_{ii}
$Ad_P^{se}(A)$	Advantage of the adversary in breaking the semantic security of a protocol

We assume that: (1) BSNS is trusted; (2) the security parameters delivered during the registration phase are sent through a secure channel; (3) after registration phase, the communication channels are insecure. Note that we use a cryptographically strong pseudo random number generator to generate random numbers, employ SHA hash function, bio-hash function [73] and the AES encryption/decryption cryptographic method in our protocol.

4.3.1. SDP Registration Phase

The registration phase is shown in Figure 7. Communications take place over a secure channel as it is a one-time process. Details are given below:

Step 1: Patient i first selects and inputs his/her identity ID_i, password PW_i, his/her sensor identity IDS_{ii}, and imprints his/her biometric Bio_i into SDP_i. Next, SDP_i generates a random integer N_{u_1}, then computes its masked password MPW_i, masked biometric information Z_i, and masked identity MID_i: $MPW_i = h(PW_i \| h_B(Bio_i) \| ID_i \| N_{u_1})$, $Z_i = h_B(Bio_i) \oplus N_{u_1}$, $MID_i = Z_i \oplus ID_i$. SDP_i submits $\{MID_i, MPW_i, IDS_{ii}, Z_i\}$ to BSNS via a secure channel.

Step 2: BSNS receives the registration request $\{MID_i, MPW_i, IDS_{ii}, Z_i\}$ from SDP_i, computes patient i's identity $ID_i = MID_i \oplus Z_i$ and then selects a high entropy integer x_s as its secret master key, which is only known to the BSNS. Next, BSNS calculates the following: $C_0 = MPW_i \oplus x_s$, $C_1 = h(MID_i \| x_s)$, $C_2 = h(MID_i \| C_1)$. BSNS sends $\{C_0, C_2\}$ to SDP_i through a secure channel.

Step 3: SDP_i stores C_0, C_2, Z_i, $h(\bullet)$, and $h_B(\bullet)$.

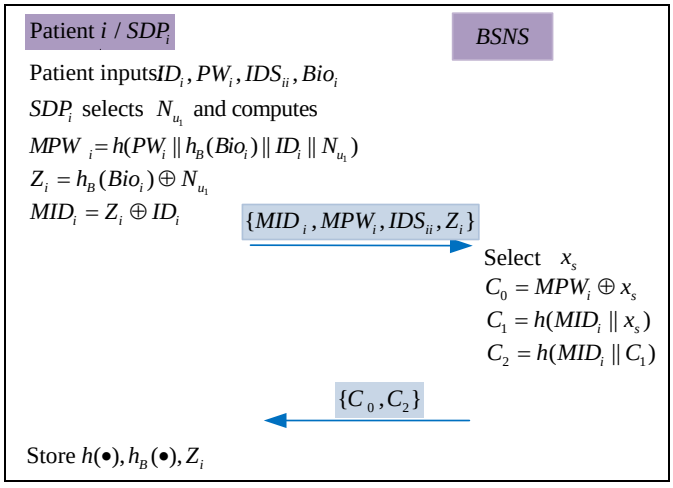


Figure 7 - SDP_i Registration Phase

4.3.2. Sensor Registration Phase

Similarly, communications between the sensor S_{ii} and SDP_i take place over a secure channel. Details are as follows:

Step 1: The bio-sensor S_{ii} its identity IDS_{ii} directly in the secure channel to BSNS.

Step 2: The BSNS stores IDS_{ii} into a database and links IDS_{ii} with ID_i. Then, it generates a random number x_b and sends $\{x_b\}$ to the sensor S_{ii}.

Step 3: The sensor S_{ii} stores x_b and $h(\bullet)$.

4.3.3. Mutual Authentication Phase

In this phase, bio-sensor S_{ii} and SDP_i should be authenticated by BSNS. Then S_{ii} and SDP_i mutually authenticate each other. After a successful mutual authentication (shown in Figure 8), SDP_i and BSNS establish a common secret session key sk_{sb} which is used for future secure communications between them. S_{ii} and SDP_i also establish a session key sk_{ss} for secure communications. Details are as follows:

Step 1: S_{ii} computes $DIDS_{ii} = h(x_b \| IDS_{ii})$, $C_6 = T_\alpha(x_b)$ by selecting a random integer α, then asks for authentication by sending $\{DIDS_{ii}, C_6\}$ to SDP_i.

Step 2: Patient *i* inputs his/her identity ID_i, PW_i, and imprints personal biometric B_i into SDP_i. SDP_i verifies the identity of patient *i* by computing the following: $MID_i^* = Z_i \oplus ID_i$, $MPW_i^* = h(PW_i \| h_B(B_i) \| ID_i \| (Z_i \oplus h_B(B_i)))$, $x_s* = C_0 \oplus MPW_i^*$, $C_1^* = h(MID_i^* \| x_s^*)$, $C_2^* = h(MID_i^* \| C_1^*)$ and then checking if $C_2^* \stackrel{?}{=} C_2$ holds or not. A mismatch between them results in immediate termination of the login phase. Otherwise, it is ensured that the patient has entered correct identity, password and biometric information. Then, SDP_i generates a random integer

N_{u_2} and computes the dynamic identity $DID_i = h(ID_i \| N_{u_2} \| x_s^*)$ and $C_3 = h(C_1^*) \oplus N_{u_2}$. Finally, SDP_i sends the message $\{DID_i, C_3\}$ together with $DIDS_{ii}$ to *BSNS* via a public channel.

Step 3: After receiving the message, *BSNS* computes: $N_{u_2}^* = C_3 \oplus h(C_1)$ and $DID_i^* = h(ID_i \| N_{u_2}^* \| x_s)$. Next, *BSNS* checks if $DID_i^* \stackrel{?}{=} DID_i$ holds. If it does not hold, *BSNS* rejects this phase immediately. Then, *BSNS* checks its database and uses the stored IDS_{ii} to check if $DIDS_{ii} \stackrel{?}{=} h(x_b \| IDS_{ii})$. If it does not hold, *BSNS* stops this phase. If both of $DID_i^* \stackrel{?}{=} DID_i$ and $DIDS_{ii} \stackrel{?}{=} h(x_b \| IDS_{ii})$ are holding, it means SDP_i and sensor S_{ii} are legal for *BSNS*.

Next, *BSNS* generates a random integer N_{S_1} and a random number x_{b1} and then proceeds to compute the following: $C_4 = N_{S_1} \oplus h(C_1 \| N_{u_2}^*)$, $sk_{sb} = h(ID_i^* \| N_{S_1} \| x_s \| N_{u_2}^*)$, $C_5 = h(sk_{sb} \| ID_i^*)$, $C_7 = E(x_b \oplus C_5 \oplus N_{u_2}^*)_{sk_{sb}}$ and $C_8 = E(x_{b1} \oplus IDS_{ii})_{h(x_b \| IDS_{ii})}$. Finally, *BSNS* sends message $\{C_4, C_5, C_7, C_8\}$ to SDP_i via a public channel.

Step 4: When SDP_i receives the message, it computes: $N_{S_1}^* = C_4 \oplus h(C_1^* \| N_{u_2})$, $sk_{sb}^* = h(ID_i \| N_{S_1}^* \| x_s^* \| N_{u_2})$ and $C_5^* = h(sk_{sb}^* \| ID_i)$. Next, SDP_i checks whether C_5^* is equivalent to C_5. If it is not equal, SDP_i stops this session. Otherwise, SDP_i believes that *BSNS* is a legal BSNS server.

Moreover, SDP_i decrypts C_7 and gets x_b, selects a random integer β, and computes: $C_{10} = T_\beta(x_b)$, $sk_{ss} = T_\beta(C_6)$, $C_9 = h(x_b \| sk_{ss})$. Later, SDP_i sends $\{C_8, C_9, C_{10}, MPW_i\}$ to the sensor S_{ii} via a public channel.

Step 5: After receiving the message, bio-sensor S_{ii} decrypts C_8 and computes $sk_{ss}^* = T_\alpha(C_{10})$. It then checks if $C_9^* \stackrel{?}{=} h(x_b \| sk_{ss}^*)$. If it does not hold, the message is rejected. Otherwise, the bio-sensor S_{ii} believes that SDP_i is a legal mobile device.

S_{ii} then replaces the stored x_b with x_{b1}, computes $C_{11} = h(MPW_i \| sk_{ss}^*)$ and sends it to SDP_i.

Step 6: After receiving the message, SDP_i asks patient *i* to input his/her identity ID_i, PW_i, and imprints personal biometric B_i. Then, SDP_i verifies the validation of S_{ii} by checking $C_{11}^* = h(MPW_i^* \| sk_{ss}^*)$, where $MPW_i^* = h(PW_i \| h_B(B_i) \| ID_i \| (Z_i \oplus h_B(B_i)))$. If it holds, it means that S_{ii} is authenticated by SDP_i successfully.

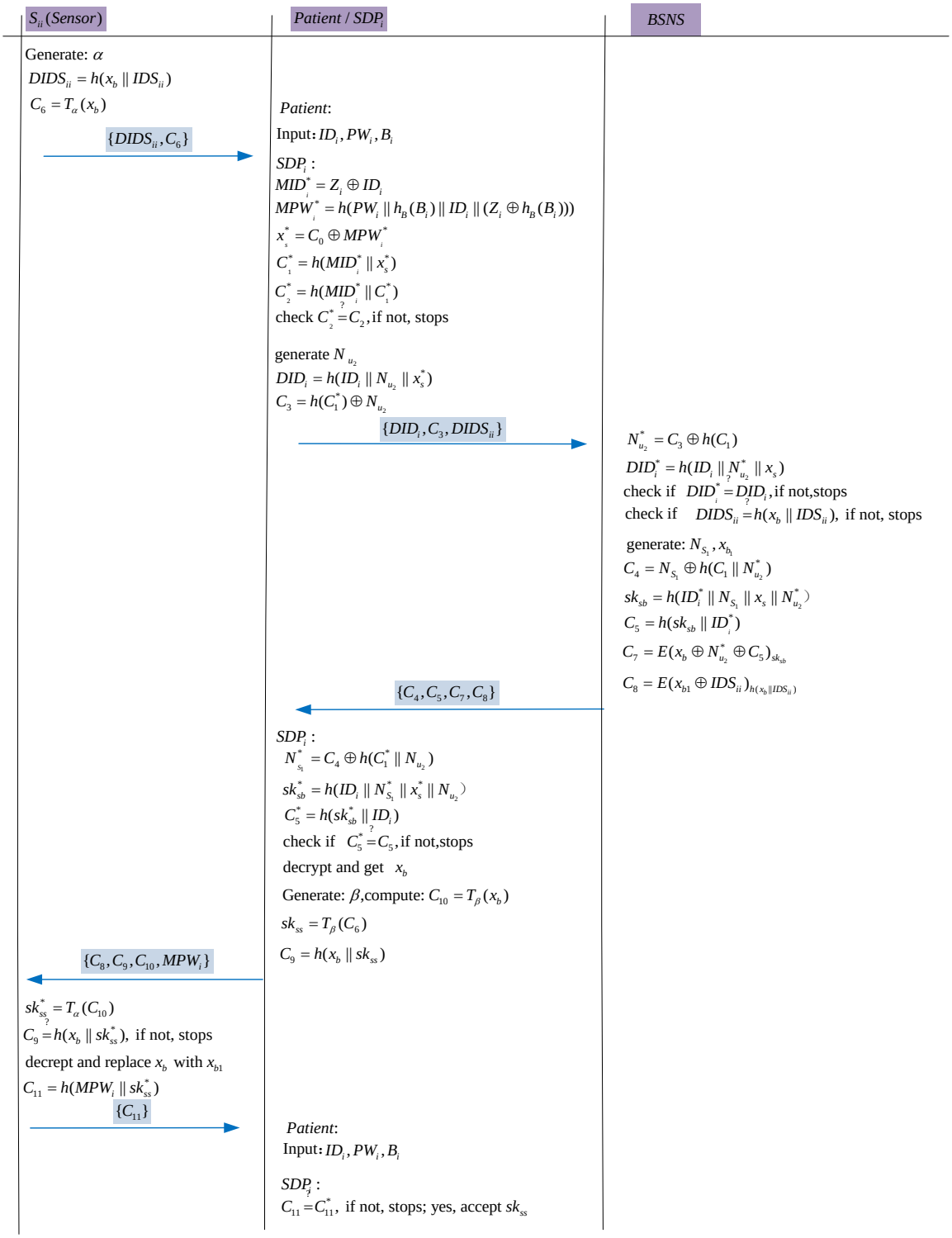


Figure 8 - Authentication Phase

4.3.4. Password Update Phase

This phase enables patient *i* to change his/her password without communication with *BSNS*. Details are seen as follows.

Patient *i* first inputs his/her identity ID_i, PW_i, and imprints personal biometric Bio_i into SDP_i. SDP_i verifies if the equation of $C_2^* \stackrel{?}{=} C_2$ holds by computing $x_s^* = C_0 \oplus h(PW_i \| h_B(Bio_i) \| ID_i \| (Z_i \oplus h_B(Bio_i)))$ and then $C_2{}^* = h((Z_i \oplus ID_i) \| h((Z_i \oplus ID_i) \| x_s^*))$. If it does not hold, SDP_i rejects the request. Otherwise, SDP_i asks patient *i* to input his/her new password.

Patient *i* inputs new password PW_i^{new} into SDP_i and SDP_i computes $MPW_i^{new} = h(PW_i^{new} \| h_B(Bio_i) \| ID_i \| N_{u_1})$ and $C_0^{new} = MPW_i^{new} \oplus x_s^*$. Finally, SDP_i replaces $\{C_0\}$ with $\{C_0^{new}\}$.

Our protocol makes it practical for SDP_i to be authenticated with its new password as SDP_i and *BSNS* do not use the patient *i*'s password in the authentication phase at all. Because no message exchange happens between SDP_i and *BSNS* in the password change phase conducted on the SDP_i by itself, our protocol enhances the security of the password change process.

4.4. Formal Security Analysis

The goal of formal security analysis is to prove that our protocol satisfies session-key security and know-key security. We utilize the formal model for security against attacks used by Bresson et al. [74] where the adversary's capabilities are modeled through queries. In this model, a participator in the authentication process could be a bio-sensor S_{ii}, SDP_i or *BSNS* and all of them have some instance Π. We define $\Pi^{w}_{S_{ii}}$, $\Pi^{u}_{SDP_i}$, Π^{v}_{BSNS} to be the instances *w*, *u*, *v* of the participator S_{ii}, SDP_i and *BSNS* respectively. These instances Π (=$\{\Pi^{w}_{S_{ii}}, \Pi^{u}_{SDP_i}, \Pi^{v}_{BSNS}\}$) are also expressed as the oracles. The following *queries* allow the adversary *A* to interact with the participators:

① Execute ($\Pi^{w}_{S_{ii}}, \Pi^{u}_{SDP_i}$) or ($\Pi^{u}_{SDP_i}, \Pi^{v}_{BSNS}$)

This query occurs under passive attacks which are modeled as eavesdropping attacks, and the output of this query includes exchanged messages among two real participants.

② Send (Π)

This query occurs if an active attacker sends a message to Π and receives a reply generated by Π.

③ CorruptS ($\Pi^{w}_{S_{ii}}$)

This query models that bio-sensor S_{ii} is stolen or compromised by adversary *A*. Thus, adversary *A* knows information stored in S_{ii} and tries to guess a session key.

④ CorruptSDP ($\Pi^{u}_{SDP_i}$)

This query models that the mobile device installing SDP_i is stolen or compromised by adversary *A*. Thus, adversary *A* knows information stored in SDP_i and tries to guess a correct password.

⑤ Reveal (Π)

This query indicates that session keys are leaked to adversary *A*.

⑥ Test (Π)

This query models session key semantic security with a coin *c*. An adversary A can ask the Test query at most once and the query is available to him/her only if the instance Π is fresh (ie. Session key is not obviously known to A). The answer to the query is such that the session key is forwarded if c = 1 and a random number is forwarded if c = 0 [74].

In the authentication protocol, an adversary can ask for a polynomial number of queries. The adversary outputs its guess *c*' for the bit *c* in the Test-query. Assume that *A* is the attacker, *P* is the authentication protocol, and *se* stands for the advantage of the attacker in breaking the semantic security. The advantage of the adversary breaking in the semantic security of our protocol is $Ad_P^{se}(A) = 2\Pr[\mathrm{Suc}(A)] - 1 = 2\Pr[c' = c] - 1$.

Theorem 1. Let adversary *A* run in polynomial time *t* against the authentication protocol in random oracle. The sizes of an identity pool, a password pool, and a biometric pool are $|D_1|, |D_2|, |D_3|$ respectively. The range space of hash oracles ($h(\bullet)$ and $h_B(\bullet)$) is $|H|$ and the range space of random numbers is |*RN*|. *P* is our protocol and *A* is an attacker breaking the semantic security within a time bound *t* with less than q_s Send(Π) queries, q_h hash queries and q_s attempts of attacker's guessing a session key from S_{ii}. Thus, the advantage of the adversary breaking the semantic security of the protocol is defined as follows:

$$Ad_P^{se}(A) \le \frac{q_h^{\,2}}{4|H|} + \frac{Adv_\Omega(l)}{2} + \frac{q_s}{2|H|} + \frac{q_c}{2|RN|} + \frac{q_s}{2|D_1|.|D_2|.|D_3|})$$

where $Adv_\Omega(l)$ is the advantage of the attacker in breaking encryption/decryption cryptographic cipher Ω and *l* is the security parameter.

Proof. We define a series of real games from G_0 to G_5, and define that an event Suc_n happens if the attacker wins the game and guesses correct session keys in the Test-query. We prove Theorem 1 using the following games:

(1) Game G_0

In game G_0 , the attacker *A* guesses the bit *c* at the beginning of the game:

$$Ad_P^{se}(A) = 2\Pr[\mathrm{Suc}_0] - 1 \qquad (1)$$

(2) Game G_1

In this game, the attacker *A* launches eavesdropping attacks by running Execute ($\Pi_{S_{li}}^{w}, \Pi_{SDP_i}^{u}$) or ($\Pi_{SDP_i}^{u}, \Pi_{BSNS}^{v}$) query. Then, the attacker *A* determines whether the output of the Test (Π) query is equal to the exact session keys (sk_{ss}, sk_{sb}). Without knowing the secret parameters (N_{s_1} , $N_{u_2}, x_s,\ x_b, \alpha, \beta$), the probability of breaking the semantic security is not increased while listening to delivered messages. Thus,

$$\Pr[\mathrm{Suc}_0] = \Pr[\mathrm{Suc}_1] \qquad (2)$$

(3) Game G_2

In this game, the attacker runs Send(Π) query to launch active attacks through simulating encryption oracles. Applying encryption under chosen plaintext attack [75], we have

$$|\Pr[\mathrm{Suc}_1]-\Pr[\mathrm{Suc}_2]|\leq Adv_{\Omega}(l) \quad (3)$$

(4) Game G_3

This game is aborted when the adversary has guessed the correct authenticators ($DID_i, DIDS_{ii}, C_5, C_9, C_{11}$) without asking the corresponding hash oracle query. According to [76], we have

$$|\Pr[\mathrm{Suc}_2]-\Pr[\mathrm{Suc}_3]| \leq \frac{q_s}{|H|} \quad (4)$$

(5) Game G_4

This game simulates CorruptSDP($\Pi^u_{SDP_i}$) queries. The attacker *A* tries to guess the correct identity/password/biometric using the dictionary attack after knowing the information stored in SDP_i. The probability of choosing *q* probable parameters is $\frac{q_s}{|D_1|.|D_2|.|D_3|}$, where the sizes of the identity pool, password pool, and biometric pool are $|D_1|,|D_2|,|D_3|$, respectively [76]. Thus, we have

$$|\Pr[\mathrm{Suc}_3]-\Pr[\mathrm{Suc}_4]| \leq \frac{q_s}{|D_1|.|D_2|.|D_3|} \quad (5)$$

(6) Game G_5

In this game, the attacker *A* simulates CorruptS($\Pi_{S_{ii}}^{w}$) queries. The attacker *A* tries to construct a valid session key $sk_{ss}(=T_{\alpha}(T_{\beta}(x_b)))$ by guessing the correct values of α and β. Define |*RN*| to be the range space of the random number; thus, we have

$$|\Pr[\mathrm{Suc}_4]-\Pr[\mathrm{Suc}_5]|\leq\frac{q_c}{|RN|} \tag{6}$$

Since all queries are simulated, the probability of guessing the bit c (*c*=1) by the attacker *A* after the test query is done is $\Pr[\mathrm{Suc}_5]=\frac{1}{2}$.

From Eq. (1) to Eq. (6), we can get

$$Ad_P^{se}(A)\leq\frac{q_h^{\,2}}{4|H|}+\frac{Adv_{\Omega}(l)}{2}+\frac{q_s}{2|H|}+\frac{q_c}{2|RN|}+\frac{q_s}{2|D_1|.|D_2|.|D_3|}) \tag{7}$$

4.5. Informal Security Analysis

We assume that the attacker can compromise messages stored in SDP_i, S_{ii} or record all messages transmitted among S_{ii}, SDP_i and *BSNS* during the authentication phase.

(1) Patient anonymity/ Untraceability

In our protocol, patient *i*'s real identity is included in $DID_i=h(ID_i\|N_{u_2}\|x_s)$ when sending the message {DID_i, C_3, $DIDS_{ii}$}. Due to the usage of the random integer N_{u_2} and x_s, and the collision-resistant hash function $h(\bullet)$, it is computationally infeasible for the attacker to derive ID_i from the message. Similarly, the attacker cannot obtain ID_i from the intercepted message during the authentication phase. Therefore, our protocol provides patient anonymity.

In addition, dynamic identity is adopted to prevent the attacker from linking the same identity. It is hard for the attacker to know whether two messages are sent from the same patient as patient *i* changes its random integer N_{u_2} to generate a new dynamic identity $DID_i = h(ID_i \| N_{u_2} \| x_s)$ in every session. Therefore, our protocol prevents patient traceability.

(2) Sensor identity anonymity/ Untraceability

Similarly, sensor identity is protected by the one-way hash function $DIDS_{ii} = h(x_b \| IDS_{ii})$. Since a random number x_b is generated in each session, we end up with a different dynamic identity per session. Therefore, sensor traceability is prevented.

(3) Patient impersonation attack

Assume that a user U_A is a malicious user who attempts to impersonate patient *i* to establish a session with *BSNS*. Since U_A is a legal user, he/she can pass the login process successfully. U_A computes $DID_A = h(ID_i \| N'_{u_2} \| x'_s)$ by selecting two parameters of (N'_{u_2}, x'_s) and sends the request $\{DID_A, C_3, DIDS_{ii}\}$ to *BSNS*. *BSNS* then checks if $DID^*_A (= h((Z_A \oplus MID_A) \| (C_3 \oplus h(C_2 \| Z_A)) \| x_s))$ is equal to DID_A. This will not hold because of the different parameters and biometric information. Therefore, a legal but malicious user cannot impersonate another legitimate user to have access to the server.

(4) Sensor impersonation attack

Similarly, U_A computes $DIDS_A = h(IDS_{ii} \| x'_b)$ by selecting a random number x'_b. However, *BSNS* will not authenticate it as $DIDS_{ii}$ is computed by one-way hash function with IDS_{ii} and x_b.

(5) Modification attack

Suppose that the attacker modifies C_3 to C_3', and sends the message C_3' to *BSNS* to impersonate patient *i*. *BSNS* can detect this attack because it checks if $DID_i^* \stackrel{?}{=} DID_i$ holds or not. If the attacker wants to pass the server's verification, it needs to construct a valid DID_i and C_3. However, the attacker does not know the value of ID_i, x_s and N_{u_2}. Likewise, the attacker cannot construct a valid C_4' without knowing ID_i, N_{S_1}, N_{u_2}, x_s. Thus, verification of $C_5^* \stackrel{?}{=} C_5$ fails. In the same fashion, verification of $C_9 \stackrel{?}{=} C_9^*$ and $C_{11} \stackrel{?}{=} C_{11}^*$ will fail in the absence of the right value for α and β. Therefore, the proposed scheme can resist modification attack.

(6) Privileged-insider attack

When patient *i* submits $\{MID_i, MPW_i, IDS_{ii}, Z_i\}$ to *BSNS* for registration, it is hard for an inside attacker to verify its correctness as the password ($MPW_i = h(PW_i \| h_B(Bio_i) \| ID_i \| N_{u_1})$) is protected by a one-way hash function with its biometric template Bio_i and a random integer N_{u_1}. As a result, a malicious insider in the server cannot obtain the patient's password during the registration phase.

(7) Replay attack

Since the transmitted messages among S_{ii}, SDP_i and *BSNS* include at least one random number, *BSNS* can detect replay messages by checking the freshness of the random number. Consequently, our protocol can prevent the attacker from launching the replay attack.

(8) Offline password guessing attack

Suppose that the attacker obtains the information $\{C_0, C_2, Z_i, h(\bullet), h_B(\bullet)\}$ stored in the mobile device from a legal patient. He/She will still not be able to guess the correct password because the password is protected by the one-way hash function of $h(PW_i \| h_B(Bio_i) \| ID_i \| N_{u_1})$. It is impossible to guess these four parameters correctly at the same time. Thus, our protocol prevents offline password guessing attack.

(9) Mobile device theft attack

Assume that the attacker knows $\{C_0, C_2, Z_i, h(\bullet), h_B(\bullet)\}$ stored in the SDP_i installed in the mobile device. The attacker will still not be able to construct a valid message $\{DID_i, C_3, DIDS_{ii}\}$. This is because DID_i is protected by a one-way hash function with its real identity ID_i and the secret number x_s. Furthermore, the attacker cannot extract valid information (e.g., real identity, password, biometric) from C_0, C_2 without knowing x_s. When asked to input correct information (identity, password, biometric) to check $C_{11} \stackrel{?}{=} C_{11}^*$, the check will fail. Therefore, our protocol resists mobile device theft attack.

(10) Sensor theft attack

Assume that the attacker knows $\{h(\bullet), x_b\}$. He/She will not be able to construct a valid $DIDS_{ii}$ without knowing the patient's identity since the BSNS confirms sensor identity by fetching the expected $DIDS_{ii}$ associated with the particular patient id from its database. This means that the attacker stealing the sensor must be hooked up to the right legal patient in order for the authentication to go through. Therefore, our scheme protects against sensor theft attack.

(11) Mutual authentication

In our protocol, both the SDP_i and S_{ii} are authenticated by *BSNS* by checking $DID_i \stackrel{?}{=} DID_i^*$ and $DIDS_{ii} \stackrel{?}{=} h(x_b \| IDS_{ii})$ respectively. *BSNS* is also authenticated by SDP_i when checking $C_5 \stackrel{?}{=} C_5^*$. After SDP_i and *BSNS* mutually authenticate each other, SDP_i and S_{ii} mutually authenticate each other by checking $C_9 \stackrel{?}{=} C_9^*$ and $C_{11} \stackrel{?}{=} C_{11}^*$. Therefore, our protocol provides mutual authentications.

(12) Man-in-the-middle attack

In man-in-the-middle attack, the attacker secretly relays and may modify the sent messages between two parties who he/she believes are directly communicating with each other. The attack is successful only when the attacker is capable of impersonating each endpoint to their satisfaction as expected from the legitimate ends; hence, by-passing mutual authentication. In our protocol, the mutual authentication occurs among sensor, patient and server. As previously argued, our protocol can resist sensor and patient impersonation attack, privileged-insider attack and modification attack. Hence, our protocol resists the man-in-the-middle attack.

(13) Session key security

Because the random integers N_{S_1} and N_{U_2} generated by *BSNS* and SDP_i are unique per session, only SDP_i and *BSNS* can compute their session key ($sk_{sb} = h(ID_i \| N_{S_1} \| x_s \| N_{u_2})$). Moreover, even if the attacker uses a valid patient's identity, it is still infeasible to guess the correct session key without knowing the parameters shown above since the session key is protected by a secure one-way hash function. This is in addition to the fact that x_s is only known to *BSNS*.

Similarly, the session key $sk_{ss} = T_\alpha(T_\beta(x_b)) = T_\beta(T_\alpha(x_b))$ is protected by the Chebyshev chaotic map where the α and β generated by the sensor and SDP_i are different in every session. As a result, if the attacker knows C_6 and C_{10} from the public channel and compromises x_b, it will still be hard for him/her to calculate the session key due to the hardness of CMDLP.

Hence, our protocol achieves session key security.

(14) Known-key security

Since N_{s_1} and N_{u_2} are generated randomly and independently by SDP_i and *BSNS* respectively, the session key sk_{sb} is unique in each run of the authentication process. Also, α, β, x_b are different in each session. Consequently, sk_{ss} is not the same in each session. Therefore, compromising the current key will not have any impact on other session keys. Thus, our protocol supports know-key security.

(15) Biometric protection

In our protocol, the biometric information is masked by a random integer and the output of a bio-hash function using XOR operation. In addition, *BSNS* verifies the biometric information through checking related values. Thus, our protocol can completely protect the patient's biometric information.

4.6. Simulation and Formal Security Verification using AVISPA

Similar to our work in chapter three, we used the AVISPA [68] simulation platform which is widely accepted by many researchers [67] to simulate and perform the formal security verification of our protocol. Simulation in AVISPA helps verify the secrecy of parameters belonging to *BSNS*, or *SDP* or bio-sensor and to confirm that mutual authentication happens among them. We created the basic roles including the roles for *BSNS*, *SDP*, and bio-sensor in

addition to defining the environment section which contains the global constant and a composition of sessions.

In our study, we executed OFMC and CL-AtSe [68], two verification back-ends embedded in AVISPA, to check for resistance to replay attack and man-in-the-middle attack. To verify resistance to the above attacks, OFMC and CL-AtSe check whether a passive /active intruder who has knowledge of legal sessions between the above roles can perform an attack. Simulation results for our scheme are presented in Figure 9 and Figure 10. We observe that our protocol is SAFE under both backends: OFMC and CL-AtSe. Therefore, we claim that our protocol resists reply attack and man-in-the-middle attack.

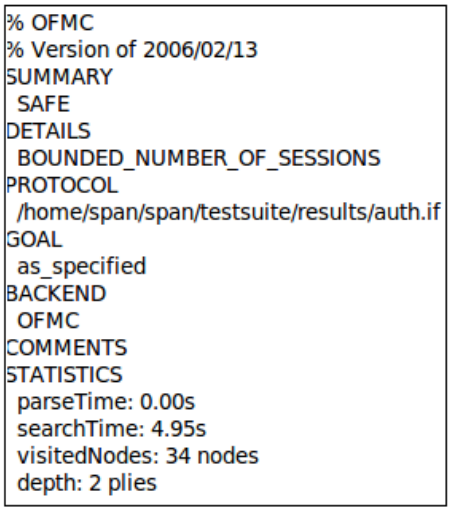


Figure 9 - Simulation Results for OFMC

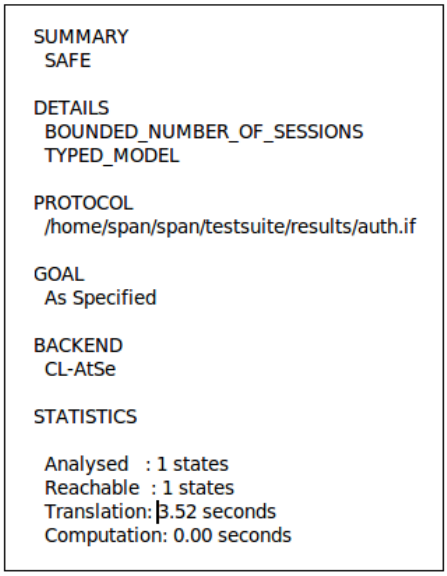


Figure 10 - Simulation Results for CL-AtSe

4.7. Performance Analysis

In this section, we compare the performance of our protocol with those in [37], [31], [38], [72], [22] and [23], where the protocols in [31], [72], [22] and [23] use three factors (identity, smart card and biometric) to authenticate the patient's identity. Before examining the communication and computational overhead, Table 10 lists the security properties of our protocol in comparison with those in [37], [31], [38], [72], [22] and [23]. According to Table 10, none of

the protocols listed in [37], [31], [38], [72], [22] and [23] satisfy all 17 security requirements; however, our protocol meets all of them. The protocol in [38] and our protocol provide mutual authentication to verify the bio-sensor and patient; however, [38], cannot resist sensor theft attack and SDP theft attack. Although Mishra et al. [23] fixed the limitations of Yan et al. [22], it was found that the protocol in [23] suffers from modification attack, smart card loss attack, and so on.

Table 10 - Security Comparisons

Properties	**Our Protocol**	[37]	[31]	[38]	[72]	[22]	[23]
Patient Anonymity	✓	✓	✓	✓	×	×	×
Patient Untraceability	✓	✓	✓	✓	×	×	×
Sensor Anonymity	✓	-	-	✓	-	-	-
Sensor Untraceability	✓	-	-	✓	-	-	-
Modification attack	✓	✓	✓	✓	×	×	×
Privileged-inside attack	✓	-	✓	-	×	×	✓
Patient impersonation attack	✓	✓	✓	×	×	×	✓
Sensor impersonation attack	✓	-	-	×	-	-	-
Replay attack	✓	✓	✓	✓	×	✓	×
Offline password guessing attack	✓	-	✓	-	×	×	✓
Smart card loss (or SDP theft) attack	✓	×	✓	×	×	×	×
Sensor theft attack		-	-	×	-	-	-
Mutual authentication	✓	✓	✓	✓	✓	✓	×
Man-in-the-middle attack	✓	×	×	×	×	×	×
Session key security	✓	×	✓	✓	×	✓	×
Know-key security	✓	✓	✓	✓	×	✓	✓
Biometric protection	✓	-	✓	-	×	×	✓

✓: The requirement is satisfied

×: The requirement is not satisfied

-: None. It has no property

4.7.1. Computational Overhead Comparison

For the computational overhead comparison during the authentication phase, let T_h, T_c, T_B, T_s, T_{sm-ecc}, T_{pa-ecc} be the time complexity of the one-way hash function operation, the Chebyshev chaotic map operation, the bio-hash function operation, the symmetric key encryption /decryption operations (for example, AES-128), a scalar multiplication operation of an elliptic curve and a point addition operation related to an elliptic curve respectively. Note that the computation for an XOR operation is very minimal compared to other operations and so it is not accounted for in the computational overhead comparisons. In our calculations, we make use of the experiment values reported in [77] and [78],where $T_h \approx 0.0001ms$, $T_c \approx T_h$, $T_B \approx T_h$, $T_s \approx 0.0002ms$, $T_{sm-ecc} \approx 0.442ms$, $T_{pa-ecc} = 0.0018ms$.

Table 11 compares the computational overhead during the authentication phase. From Table 11, we can see that Yan et al. [22] has the lowest computational overhead for the smart card from the user. However, in the scheme presented by Yan et al. [22], the smart card keeps executing the login process even if the wrong identity and password are input which results in extra unnecessary communication and computational overhead in the login/ authentication phase. Moreover, this makes the scheme vulnerable to offline password guessing attack, modification attack, and so on. In addition to that, the scheme exposes the biometric information to the public resulting in more serious damages.

We observe that our protocol has a slightly higher computational overhead in comparison to Gope [37], Zhang [31], Zan [72] and Mishra [23]. However, our protocol provides more security properties (e.g., authenticating the sensor, resistance to sensor theft attack) than those in the aforementioned papers. Both Yeh [38] and our protocol authenticate the sensor, user and server; however, Yeh [38] has a higher computational overhead in the mobile device and the server because of their use of elliptic curve cryptography. Therefore, our protocol achieves a delicate balance between security and computational overhead.

Table 11 - Comparison of Computational Overhead in Authentication Phase

Protocol	**Authentication phase**		
	Bio-sensor	**Mobile device from patient (or user)**	**Server**
Our Protocol	$3T_H + T_s + 2T_C$ ≈ 0.0007ms	$11T_H + T_s + 2T_B + 2T_C$ ≈ 0.0017ms	$7T_H + 2T_s$ ≈ 0.0011ms
[37]	-	$7T_h$ ≈ 0.0007ms	$7T_h$ ≈ 0.0007ms
[31]	-	$9T_H + 4T_B$ ≈ 0.0013ms	$9T_H + 3T_B$ 0.0012ms
[38]	$4T_h$ ≈ 0.0004ms	$6T_h + 4T_{sm-ecc} + T_{pa-ecc}$ ≈ 1.77ms	$10T_h + 4T_{sm-ecc} + T_{pa-ecc}$ ≈ 1.774ms
[72]	-	$7T_H + T_S$ ≈ 0.0009ms	$4T_h\ 4T_H + T_s$ ≈ 0.0006ms
[22]	-	$5T_h$ ≈ 0.0005ms	$5T_h$ ≈ 0.0005ms
[23]	-	$7T_H + T_B + T_S$ ≈ 0.001ms	$6T_H + T_s$ ≈ 0.0008ms
-: Property non-existent			

4.7.2. Communication Overhead Comparison

Table 12 depicts the communication overhead incurred by our protocol in comparison to other related schemes. Assume that the size of an element in elliptic curve cryptography is 160bits, the length of the output of the hash function is 160bits, the length of the output of the symmetric encryption/decryption is 128bits, the output of the bio-hash function is 160bits and the length of the output of real identity, the length of Chebyshev chaotic map, and random integer is 160bits respectively.

We observe that in our protocol, the size of messages $\{DIDS_{ii}, C_6, C_{11}\}$ sent out by the bio-sensor is about 480bits, the size of messages $\{DID_i, C_3, DIDS_{ii}, C_8, C_9, C_{10}, MPW_i\}$ sent out by the SDP (or user) is about 1088bits and the size of messages $\{C_4, C_5, C_7, C_8\}$ sent out by a server is about 576bits. In [38], sizes of messages sent out by a bio-sensor, a user and a server are 800bits, 1600bits, and 800bits respectively. In comparison to [38], communication overhead of bio-sensor, SDP and server in our protocol are only 60% of [38], 68% of [38], and 72% of [38] respectively. Therefore, our protocol reduces communication overhead particularly in the bio-sensor which is very important because wearable sensors are very resource constrained.

The protocols in [37], [31], [72], [22] and [23] have less communication overhead compared to our protocol, but we should point out that they do not validate the bio-sensor and only provide a session key between the user and the server. Security analysis of these protocols also shows that compromising the bio-sensor results in leaking the patient's sensitive information. Therefore, our protocol achieves a delicate balance between security and communication overhead.

Table 12 - Comparison of Communication Overhead in Authentication Phase

Protocol	Authentication phase		
	Bio-sensor	Mobile device from patient (or user)	Server
Our Protocol	480bits	1088bits	576bits
[37]	-	800bits	480bits
[31]	-	640bits	480bits
[38]	800bits	1600bits	800bits
[72]	-	448bits	320bits
[22]	-	640bits	320bits
[23]	-	608bits	320bits

Chapter 5: Conclusions and Future Work

In today's world, we observe a rapid development in information technology, emergence of a wide range of devices and ease of user communication through Wi-Fi or cellular networks. This emerging trend has brought us a new application, e-health, which describes the application of information and communications technologies in the health sector. One of the most prominent technologies in e-health is the use of Body Sensor Network (BSN) that continuously monitor patients using a collection of tiny-powered and lightweight bio-sensors. This offers convenience to both physicians and patients in the modern health care environment. However, since bio-sensors are deployed in a hacker-prone setting, they are vulnerable to various security threats exposing the security and privacy of patient information. In this thesis, we presented an authentication scheme for each of two applications of medical sensor networks.

In chapter three, we presented an ECC based authentication scheme suitable for a hospital-like setting whereby the patient is hooked up to sensors connected to a medical device such as an ECG monitor while the doctor, who is constantly on the move, needs real-time access to continuous sensor readings. This scheme is the first complete end-to-end scheme that can be deployed in a real-time environment across the doctor/nurse, trusted server, sensor and patient. It is lightweight making it suitable for deployment on medical devices such as ECG monitors and continuously monitors the patient allowing for continuous patient identity verification. Fuzzy extraction was used to protect the biometric template. Formal and informal security analysis showed that our protocol is resistant to user and sensor impersonation attacks, physical sensor theft and more. Simulation using AVISPA proved our scheme is resistant to passive and active attacks. Moreover, our scheme also makes the smartcard and the medical device connected to the sensor incur a low computational overhead. Its sensor computational and overall communication overhead is competitively amongst the most efficient in comparison to similar schemes.

In chapter four, we presented a Chebyshev chaotic map-based authentication scheme suitable for deployment on wearable sensors allowing readings from the lightweight sensors connected to patients to be sent and stored on a trusted server while the patient is on the move. This scheme does not use any complex operations making it suitable for deployment on lightweight wireless sensors while providing patient/sensor anonymity and preventing against traceability through the use of dynamic identity. The bio-hash function was used to protect the biometric template. Formal and informal security analysis showed that our protocol satisfies session-key security and know-key security in addition to preventing replay attack, man-in-the-middle attack and sensor theft attack amongst others. Simulation using AVISPA proved our scheme is resistant to passive and active attacks. In addition to that, the communication overhead incurred by the bio-sensor, SDP and server in our scheme was shown to be significantly lower than similar schemes especially for the bio-sensor.

Our proposed scheme in chapter three, to the best of our knowledge, is the first end-to-end doctor to patient biometric authentication schemes ensuring a secure communication channel. Both our protocols have been proven to address security and performance weaknesses of similar schemes; however, there is a big opportunity to improve on the performance of our protocols to make them even more lightweight. This particularly the case for the scheme applying ECC which we claimed can be supported on larger medical devices as opposed to resource constrained wearable sensors.

Moreover, both our protocols were designed to be more secure and efficient relevant to similar schemes and were simulated using AVISPA, a simulation and formal verification tool. We did not get the opportunity to really test their deployment with the real stakeholders: hospitals making use of BSNs. Therefore, whether they will hold against the security risks and performance constraints of a real-life setting remains an open question.

Another aspect to further investigate is the continuous identity verification component of our scheme proposed in chapter three. We assumed the use of state-of-the-art neural networks trained to identify patients using ECG signals. The accuracy rate of different types of neural networks can be analyzed to examine and identify whether this approach is sound enough to recognize patients and whether fluctuating ECG signals of a patient can affect the number of false positives resulting in a low identity recognition rate.

Finally, the proposed schemes in this thesis were targeting BSNs in a medical setting utilizing medical equipment and very resource constrained wearable sensors. Our schemes are however not limited to these application only so future researchers are invited to look into other applications that may require lightweight authentication and key agreement schemes.

Bibliography

[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, p. 2946, 2017.

[2] P. Kumar, S. Lee and H. 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, p. 2919, 2017.

[4] 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, p. 2123, 2016.

[5] J. Jung, J. Kim, Y. Choi and D. Won, "An Anonymous User Authentication and Key Agreement Scheme Based on Symmetric Cryprosystem in Wireless Sensor Networks," *Sensors*, vol. 16, p. 1299, 2016.

[6] N. Mohsen, B. Ying and A. Nayak, "Authentication Protocol for Real-time Wearable Medical Sensor Networks using Biometrics and Continuous Monitoring," in *The 12th IEEE International Conference on Internet of Things [iThings 2019]*, Atlanta, 2019.

[7] B. Ying, N. Mohsen and A. Nayak, "Protection for e-health systems using three-factor user authentication," in *IEEE International Conference on Communications*, Shanghai, China, 2019.

[8] T. S. Messerges, E. A. Dabbish and R. H. Sloan, "Examining smart-card security under the threat of power analysis attack," *IEEE Trans. Comput*, vol. 51, pp. 541-552, 2002.

[9] C. Li, C. Lee, C. Weng and S. Chen, "A secure dynamic identity and chaotic maps based user authentication and key agreement scheme for e-health care systems," *Int. J. Commun.*

Syst., vol. 40, no. 11, 2016.

[10] 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.

[11] 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, p. 136, 2017.

[12] 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. 12, pp. 29841-29854, 2015.

[13] A. S. Pathan, H. W. Lee and C. S. Hong, "Security in Wireless Sensor Networks: Issues and Challenges," in *In Proceedings of the 8th International Conference Advanced Communication Technology (ICACT)*, Phoenix Park, Korea, 2006.

[14] M. L. Das, "Two-factor User Authentication Scheme in Wireless Sensor Networks," *IEEE Trans. Wirel. Commun.,* vol. 8, pp. 1086-1090, 2009.

[15] 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.

[16] J. Kim, D. Lee, W. 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, pp. 6443-6462, 2014.

[17] 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.

[18] A. Awasthi and K. Srivastava, "A biometric authentication scheme for telecare medicine information systems with nonce," *J. Med. Syst.,* vol. 37, no. 5, pp. 1-7, 2013.

[19] D. Mishra, S. Mukhopadhyay, S. Kumari, M. Khan and A. Chaturvedi, "Security enhancement of a biometrics based authentication scheme for telecare medicine information systems with nonce," *J. Med. Syst.,* vol. 38, no. 5, pp. 1-11, 2014.

[20] Z. Tan, "A User Anonymity Preserving Three-Factor Authentication Scheme for Telecare Medicine Information Systems," *J. Med. Syst.,* vol. 38, no. 3, pp. 1-9, 2014.

[21] H. Arshad and M. Nikooghadam, "Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems," *J. Med. Syst.,* vol. 38, no. 3, pp. 1-9, 2014.

[22] X. Yan, W. Li, P. Li, J. Wang, X. Hao and P. Gong, "A secure biometrics based authentication scheme for telecare medicine information systems," *J. Med. Syst.,* vol. 37, no. 5, pp. 1-6, 2013.

[23] D. Mishra, S. Mukhopadhyay, A. Chaturvedi, S. Kumari and M. Khan, "Cryptanalysis and improvement of Yan et al.'s biometric-based authentication scheme for telecare medicine information systems," vol. 38, no. 6, pp. 1-12, 2014.

[24] M. Sarvabhatla, M. Giri and C. S. Vorugunti, "Cryptanalysis of cryptanalysis and improvement of Yan et al. biometric-based authentication scheme for TMIS," vol. 7, no. 1, pp. 42-45, 2014.

[25] R. Amin and G. P. Biswas, "A secure three-factor user authentication and key agreement protocol for TMIS with user anonymity," *J. Med. Syst.,* vol. 39, no. 8, pp. 1-19, 2015.

[26] L. Zhang, S. Zhu and S. Tang, "Privacy protection for telecare medicine information systems using a chaotic map-based three-factor authenticated key agreement scheme," *IEEE J. Biomed. Health Inf.,* vol. 21, no. 2, pp. 465-475, 2017.

[27] 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, 2015.

[28] J. Ryu, H. Lee, H. Kim and D. Won, "Secure and Efficient Three-Factor Protocol for Wireless Sensor Networks," *Sensors,* vol. 18, no. 12, p. 4481, 2018.

[29] 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.

[30] 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, pp. 1-16, 2016.

[31] 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 Transactions on Industrial Electronics,* vol. 1, p. 99, 2017.

[32] 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.

[33] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo and Y. Wu, "ALARM-NET: wireless sensor networks for assisted-living and residential monitoring," *Dept. Comput. Sci., Univ. Virginia, Charlottesville,* vol. 26, p. 17, 2006.

[34] P. Kumar and H. J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: A survey," *Sensors (Basel),* vol. 12, no. 1, p. 62, 2012.

[35] "MizaZ Datasheet," 12 August 2019. [Online]. Available: http://www.openautomation.net/uploadsproductos/ micaz_datasheet.pdf .

[36] S. Pai, M. Meingast, T. Roosta, S. Bermudez, S. B. Wicker, D. K. Mulligan and S. Sastry, "Transactional Confidentiality in Sensor Networks," *IEEE Security & Privacy,* vol. 6, no. 4, pp. 28-35, 2008.

[37] P. Gope and T. Hwang, "BSN-care: a secure IoT-based modern healthcare system using body sensor network," *IEEE Sensors J.,* vol. 16, no. 5, pp. 1368-1376, 2016.

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

[39] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *J. Netw. Comput. Appl.,* vol. 106, pp. 117-123, 2018.

[40] F. Wei, P. Vijayakumar, J. Shen, R. Zhang and L. Li, "A provably secure password-based anonymous authentication scheme for wireless body area networks," *Computers and Electrical Engineering,* vol. 65, pp. 322-331, 2018.

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

[42] L. Wu, Y. Zhang, L. Li and J. Shen, "Effecient and anonymous authentication scheme for wireless body area networks," *J. Med. Syst.,* vol. 40, p. 134, 2016.

[43] V. Odelu, S. Saha, R. Prasath, L. Sadineni, M. Conti and M. Jo, "Efficient privacy preserving device authentication in WBANs for industrial e-health applications," *Computers and Security,* vol. 83, pp. 300-312, 2019.

[44] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta and K. K. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Comput. Netw,,* vol. 129, pp. 429-443, 2017.

[45] C. Chen, B. Xiang, T. Wu and K. Wang, "An anonymous mutual authenticated key agreement scheme for wearable sensors in wireless body area networks," *MDPI,* pp. 1-15, 2018.

[46] J. Moon, D. Lee, Y. Lee and D. Won, "Improving Biometric-Based Authentication Schemes with Smart Card Revocation/Reissue for Wireless Sensor Networks," *Sensors,* vol. 17, p. 940, 2017.

[47] A. B. Teoh, Y. W. Kuan and S. Lee, "Cancellable biometrics and annottions on BioHash," *Pattern Recognit.,* vol. 41, pp. 2034-2044, 2008.

[48] X. Y. Huang, Y. Xiang, A. Chonka, J. Y. Zhou and R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Trans. Parallel Distrib. Syst.,* vol. 22, no. 8, pp. 1390-1397, 2011.

[49] J. S. Yu, G. L. Wang, Y. Mu and W. Gao, "An effecient generic framwork for three-factor authentication with provable secure instantiation," *IEEE Trans. Inf. Forensics Secur.,* vol. 9, no. 12, pp. 2302-2313, 2014.

[50] T. F. Lee, "An efficient chaotic maps-based authentication and key agreement scheme using smartcards for telecaremedicine information systems," *J. Med. Syst.,* vol. 37, no. 6, pp. 1-9, 2013.

[51] D. Mishra, J. Srinivas and S. Mukhopadhyay, "A secure and efficient chaotic map-based

authenticated key agreement scheme for telecare medicine information systems," *J. Med. Syst.,* vol. 38, no. 10, pp. 1-10, 2014.

[52] M. S. Farash and M. A. Attari, "Cryptanalysis and improvement of a chaotic map-based key agreement protocol using Chebyshev sequence membership testing," *Nonlinear Dyn.,* vol. 76, no. 2, pp. 1203-1213, 2014.

[53] X. Hao, J. Wang, Q. Yang, X. Yan and P. Li, "A chaotic map-based authentication scheme for telecare medicine information systems," *J.Med. Syst.,* vol. 37, no. 2, p. 9919, 2013.

[54] X. Y. Wang and D. P. Luan, "A secure key agreement protocol based on chaotic maps," *Chin. Phys. B,* vol. 22, no. 11, p. 110503, 2013.

[55] V. Kumar, M. Ahmed and A. Kumari, "A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS," *Telematics and Informatics,* vol. 38, pp. 100-117, 2019.

[56] D. Mishra, "Understanding Security Failures of Two Authentication and Key Agreement Schemes for Telecare Medicine Information Systems," *Journal of Medical Systems,* vol. 39, no. 19, 2015.

[57] G. Wu, J. Wang, Y. Zhang and S. Jiang, "A Continuous Identity Authentication Scheme Based on Physiological and Behavioral Characteristics," *Sensors,* vol. 18, p. 179, 2018.

[58] G. Cola, M. Avvenuti, F. Musso and A. Vecchio, "Gait-based Authentication Using a Wrist-worn Device," in *In Proceedings of the Thirteenth International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, Hiroshima, Japan, 2016.

[59] S. Mondal and P. Bours, "Continuous Authentication in a Real World Settings," in *In Proceedings of the 2015 Eighth International Conference on Advances in Pattern Recognition (ICARP)*, Kolkota, India, 2015.

[60] Y. Chuang, N. Lo, C. Yang and S. Tang, "A Lightweight Continuous Authentication Protocol for the Internet of Things," *Sensors,* vol. 18, p. 1104, 2018.

[61] C. Camara, P. Peris-Lopez and J. E. Tapiador, "Human Identification Using Compressed ECG Signals," *J. Med. Syst,* vol. 39, p. 149, 2015.

[62] S. J. Kang, S. Y. Lee, H. I. Cho and H. Park, "ECG Authentication System Design Based on Signal Analysis in Mobile and Wearable Devices," *IEEE Signal Process. Lett*, vol. 23, pp. 805-808, 2016.

[63] H. Kim, R. F. Yazicioglu, S. Kim and N. V. Helleputte, "A Configurable and Low-power Mixed Signal SoC for Portable ECG Monitoring Applications," in *In Proceedings of the symposium on VLSI Circuits*, Honolulu,HI, USA.

[64] Q. Zhang, D. Zhou and X. Zeng, "HeartID: A multiresolution convolutional neural network for ECG-based biometric human identification in smart health applications," *IEEE Access*, vol. 5, pp. 11805-11816, 2017.

[65] M. Burrows, M. Abadi and R. Needham, "A Logic of Authentication," *IEEE Trans. Comput.*, vol. 8, pp. 18-36, 1990.

[66] D. Johnson, A. Menezes and T. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *IJIS*, vol. 1, pp. 36-63, 2001.

[67] S. HafizullIslam, R. Amin, G. Biswas, M. 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,," *Journal King Saud. Univ. SCI.*, vol. 29, no. 3, pp. 311-324, 2017.

[68] AVISPA, "Automated validation of internet security protocols and applications," September 2019. [Online]. Available: http://www.avispa-project.org/.

[69] AVISPA, "The HLPSL Tutorial - A Beginner's Guide to Modeling and Analyzing Internet Security Protocols," *Information Society Technologies Programme*, 2006.

[70] D. Dolev and A. Yao, "On the Security of Public-Key Protocols," *IEEE Transactions on Information Theory*, vol. 2, no. 29, 1983.

[71] S. Qiu, G. Xu, H. Ahmad and Y. Guo, "An Enhanced Password Authentication Scheme for Session Initiation Protocol with Perfect Fprward Secrecy," *PLOS ONE*, vol. 13, no. 3, 2018.

[72] Z. Tan, "An efficient biometrics-based authentication scheme for telecare medicine information systems," *Przeglad Elektrotechniczy*, vol. 89, no. 5, pp. 200-204, 2013.

[73] A. Lumini and L. Nanni, "An improved BioHashing for human authentication," *Pattern Recognition,* vol. 40, pp. 1057-1065, 2007.

[74] E. Bresson, O. Chevassut and D. Pointcheval, "Security proofs for an efficient password-based key exchange," *ACM CCS 2003,* pp. 241-250, 2003.

[75] S. Wu and K. Chen, "An effecient key-management scheme for hierarchical access control in E-medicine system," *J. Med. Syst.,* vol. 36, no. 4, pp. 2325-2337, 2012.

[76] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular network," *IEEE Trans. Veh. Technol.,* vol. 66, no. 12, pp. 10626-10636, 2017.

[77] D. He, N. Kumar, J. H. Lee and R. S. Sherrat, "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Trans. Consum. Electron.,* vol. 60, no. 1, pp. 30-37, 2014.

[78] D. He, S. Zeadally, B. Xu and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular adhoc networks," *IEEE Trans. Inf. Forensics Security,* vol. 10, no. 12, pp. 2681-2691, 2015.

Appendix:

HLPSL Definition of the protocol in chapter 3:

```
role doctor( U,TS,S:agent, H, EccMul, Rep: hash_func, SND,RCV:channel(dy),
ID,PW,BIO,Pi,Ai,Ci,P,Y,TID:text)

played_by U
def=
        local State:nat,
        Alpha :text,
        Ri, HPW, Xsi, R1, Li, Bi, Xi, X, HX, DID, MUiG, AY, Yj, AYJ, MGuI : message
        const sec_1, sec_2, sec_3 : protocol_id

        init State := 0

        transition
        1. State=0 /\ RCV(start) =|>
             State':=1 /\ Ri':=Rep(BIO.Pi)
        /\ HPW':=H(PW.Ri')
        /\ Xsi':=xor(Ai, HPW')
        /\ R1':=xor(Ci, H(Xsi'))
        /\ Li':=H(Ri'.Xsi'.ID)
```

/\ Bi':=H(R1'.Li')

/\ Alpha':=new()

/\ Xi':= EccMul(Alpha'.P)

/\ X':= EccMul(Alpha'.Y)

/\ HX':=H(Xi'.X')

/\ DID':=xor(TID, HX')

/\ MUiG':=H(Xsi'.Xi'.X')

/\ SND(DID'.Xi'.MUiG')

/\ secret({X'},sec_1,{U,TS})

/\ secret({Alpha'}, sec_2, {U})

/\ witness (U, TS, doctor_server_alpha, Alpha')

2. State=1 /\ RCV(Yj'.MGuI') =|>

State':=2 /\ AYJ':= EccMul(Alpha.Yj')

/\ AY':= EccMul(Alpha.Y)

/\ MGuI':=H(Xsi.AYJ'.AY'.TID)

/\ secret({Xsi}, sec_3, {U,TS})

end role

role trusted_server(U,TS,S:agent, H, EccMul: hash_func, SND,RCV:channel(dy),

R0, ID, XX, PID, P, C0, IDS : text)

played_by TS

def=

local State : nat,

Beta, Tg, Tj :text,

X, HX, DID, Xi, MUiG, Xsi, Yj, Xbeta, Xsj, MGsJ, MSjG, Kj, Z, MGuI, TID1 :message

const sec_4, sec_5: protocol_id

init State := 0

transition

1. State=0 /\ RCV(DID'.Xi'.MUiG') =|>

State':=1 /\ X':=EccMul(XX.Xi')

/\ HX':=H(Xi'.X')

/\ TID1':=xor(DID',HX')

/\ Xsi':=H(ID.XX.R0)

/\ MUiG':=H(Xsi'.Xi'.X')

/\Beta':=new()

/\ Yj':=EccMul(Beta'.P)

/\ Xsj':=H(C0.IDS.PID)

/\ Xbeta':=xor(Beta',Xsj')

```
        /\Tg':=new()
        /\ MGsJ':=H(Xsj'.Xi'.IDS.Tg')
        /\SND(Xi'.MGsJ'.Xbeta'.Tg')

        /\secret({Beta'}, sec_4, {TS,S})
        /\secret({Xsj'}, sec_5, {TS,S})
        /\ witness (TS, U, server_doctor_beta, Beta')

        2. State=1 /\ RCV(MSjG'.Tj') =|>
        State':=2 /\  Kj':=H(Xsj.Tj')
        /\ Z':= EccMul(Beta.Xi)
        /\ MSjG':=H(Kj'.Yj.Xsj.Xi.Tj'.PID)
        /\ MGuI':=H(Xsi.Z'.X.TID1)
        /\SND(Yj.MGuI')
end role

role sensor(U,TS,S: agent, H,EccMul: hash_func, SND,RCV:channel(dy),
PID, C0, IDS, P :text)

played_by S
def=
        local State:nat,
```

```
        Tg, Tj: text,
        Xi, Xbeta, Xsj, MGsJ, Yj, Kj, Z, MSjG , Beta1:message
        const sec_6 :protocol_id

        init State := 0
        transition
        1. State=0 /\ RCV(Xi'.MGsJ'.Xbeta'.Tg') =|>
              State':=1 /\ Xsj':=H(C0.PID.IDS)
        /\  MGsJ':= H(Xsj'.Xi'.IDS.Tg')
        /\ Beta1':=xor(Xbeta',Xsj')
        /\ Yj':= EccMul(Beta1'.P)
        /\Tj':=new()
        /\ Kj':=H(Xsj'.Tj')
        /\ Z':=EccMul(Beta1'.Xi')
        /\ MSjG':= H(Kj'.Yj'.Xsj'.Xi'.Tj'.PID)

        /\ SND(MSjG'.Tj')
        /\ secret(Z', sec_6, {S,TS})
end role

role session(U,TS,S:agent, H,EccMul, Rep: hash_func)
def=
```

```
        local
                SND1,RCV1,SND2,RCV2,SND3,RCV3:channel(dy),
                ID, PW, BIO, Pi, Ai, Ci, P, Y, R0, XX, PID, C0, IDS, TID:text
        composition

           doctor(U, TS, S, H, EccMul, Rep, SND1, RCV1, ID, PW, BIO, Pi, Ai, Ci, P, Y, TID)
        /\ trusted_server(U, TS, S, H, EccMul, SND2, RCV2, R0, ID, XX, PID, P, C0, IDS)
           /\ sensor(U, TS, S, H, EccMul, SND3, RCV3, PID, C0, IDS, P)
end role

role environment()
def=
        const u, ts, s: agent,
        eccMul, rep, hsh: hash_func,
           sec_1,sec_2, sec_3, sec_4, sec_5, sec_6, sec_7, doctor_server_alpha,
server_doctor_beta, doctor_server_did, doctor_server_xi,
        server_sensor_c0, server_sensor_pid, server_sensor_ids:protocol_id,
        id, bio, pi, ai, ci, p, y, r0, pid, c0, ids, tid:text
          intruder_knowledge = {u, ts, s, id, pi, ai, ci, p, y, r0, bio, pid, c0, ids, tid }

        composition
        session(u,ts, s, hsh, eccMul, rep)
        /\ session(i,ts, s, hsh, eccMul, rep)
```

```
        %/\ session(u,ts, i, hsh, eccMul, rep)
end role

goal
        secrecy_of sec_1, sec_2, sec_3, sec_4, sec_5, sec_6
        authentication_on  doctor_server_alpha
        authentication_on server_doctor_beta
end goal
environment
```