Journal of Reliable Intelligent Environments (2018) 4:141–160
https://doi.org/10.1007/s40860-018-0062-5

ORIGINAL ARTICLE

Multi-factor user authentication scheme for IoT-based healthcare services

Parwinder Kaur Dhillon[1] · Sheetal Kalra[1]

Received: 21 November 2017 / Accepted: 8 May 2018 / Published online: 22 May 2018

Abstract

Due to the tremendous rise of the cloud computing and the Internet of Things (IoT) paradigms, the possibility of remote monitoring of the patients in real time by a remote Medical Professional (MP) has become feasible and patients can enjoy healthcare services at home. To achieve this, the patient's medical data will need to be stored on the Cloud server. However, patient's medical data stored on server are highly sensitive and, hence, the Cloud-IoT network becomes open to many attacks. For that reason, it must ensure that patients' medical data do not get exposed to malicious users. This makes strong user authentication a prerequisite for the successful global deployment of centralized healthcare systems. In this paper, we present an efficient, strong authentication protocol, for the MP to access patient data for healthcare applications based on Cloud-IoT network. The proposed protocol includes: (1) three-factor MP authentication (i.e. password, biometrics and smartcard); (2) mutual authentication between MP and the cloud server; (3) establishes a secure shared session key; and (4) maintains key freshness. Furthermore, the proposed protocol uses only two message exchanges between MP and cloud server, and attains efficiency (i.e. low computation and communication costs). Through the formal analysis using AVISPA web tool, security analysis and performance analysis, we conclude that the proposed protocol is more secure against potential attacks and obtains a trade-off between security and performance cost for healthcare application using Cloud-IoT networks.

Keywords Authentication · Biometrics · Big data · Cloud · ECC · Healthcare · IoT

1 Introduction

Internet of Things (IoT) and next-level big data analytics tools are promising Information Communication Technology (ICT) paradigms possessing potential to transform healthcare services. This is due to the increased pervasive existence of smart devices embedded with Radio Frequency IDentification (RFID) tags, sensors, and actuators nodes, having unique IP addresses. Using the unique address, these objects can communicate together and use data gathered, for producing interpretations or predicting some results [1]. Big data analytics tools can aid the physicians use complex predictive analysis for early prediction of certain diseases from the patient's Electronic Health Records (EMRs). This will allow prevention of chronic ailments, reduce treatment costs, personalized and better healthcare facilities. Also, physicians can use big data analytics tools for checking out alternative treatment options for a particular patient based on factors such as personal history, prior health issues, and hereditary data.

Furthermore, the innovative advancements in Wireless Body Area Networks (WBANs) have allowed several wearable sensors and devices deployed on to patient body. This allows for ubiquitous monitoring and tracking of physiological data and health-related information. Integrating WBANs with IoT, cloud and big data technologies will allow for real-time monitoring of patients anytime and anywhere. This led to the development of Real-time Health Systems (RTHS). These systems will be vital for healthcare in IoT, because Big Data Analytics tools and processes will be applied to estimate both dynamic and static data for predictive analysis. Since these systems will operate in heterogeneous wireless environments which are insecure, they require secure communication of patient's health information along with

✉ Parwinder Kaur Dhillon
parwindhillon@gmail.com

Sheetal Kalra
sheetal.kalra@gmail.com

[1] Department of Computer Science and Engineering, Guru Nanak Dev University, Regional Campus, Jalandhar, Punjab 144001, India

a guarantee of data integrity and confidentiality for reliable healthcare architecture. Major challenges of RTHS are (1) key management for secure communication, (2) secure data forwarding and (3) patient-centric access control to the stored EMRs.

In this paper, we have proposed a new lightweight key management and authentication protocol for healthcare services based on Cloud-IoT and big data environment. The protocol establishes a secure communication channel between a physician and a remote entity (i.e. cloud server). Using this secured channel physician can access patients EMRs stored on cloud server while ensuring confidentiality and authentication. Our protocol considers a network model consisting of a centralized healthcare authority to which several hospitals are connected.

Cloud-IoT-based healthcare service architecture used to discuss the protocol proposed in this paper is shown in Fig. 1. The main entities of the system are IoT network (containing patients and sensors), cloud server, centralized Healthcare Authority (HA), hospitals and the MPs. The IoT network will consist of nodes of two types. *First* are the sensor nodes that continuously or on-demand measure or observe patient's health-related data and report it to the smart device such as Smartphone, e.g. rate of heart beat, and body temperature. *Second* are actuators; these nodes receive commands from the physicians, nurses or other medical staff to execute actions necessary to deal probable health issues, e.g. breathing pumps in the case of asthma or insulin pumps in diabetes.

Every patient consists of tiny nodes embedded onto his body which gather vital static and dynamic statistics about his health, e.g. blood glucose sugar level, heart beat sensors, etc. The data get collected and stored on the cloud server. This cloud server is controlled and managed by the centralized HA, to which many hospitals are connected. Setting up such a centralized system to share information of a patient between the different hospitals allows improving the quality of health care for patients. From the cloud server, any MP registered at the Healthcare authority can access its patient's sensor data in real time and remotely monitor and suggest diagnosis. This centralized system allows the patient to be registered with the centralized HA once and then all the medical records are controlled by it.

From the proposed scenario, we have identified vital requirements for the centralised remote patient remote health-monitoring system. *First*, the system should allow a patient/MP to monitor the health anytime anywhere through the smart devices and sensors. *Second*, the system must be independent of the geographical location of both patient and MP. *Third*, the system must also be scalable to handle many patients and healthcare professionals such as MP and nurses along with different health devices and data formats. *Fourth*, the system must be generic so that it is able to monitor different health scenarios. *Fifth*, system must maintain patient's privacy. *Lastly*, the system should be user-friendly and simple enough for the MPs to use.

1.1 Motivations

Authentication and key establishment play a significant role in heterogeneous environments and this has led to the development of several schemes for providing secure communication. These schemes have their own advantages and disadvantages. To the best of our knowledge, there is no proposed key establishment and authentication scheme specific for a centralized Cloud-IoT and big data-based healthcare applications until now. In fact, in a centralized Cloud-IoT environment, we need a key establishment scheme, which allows a physician or medical professional to securely access EHRs from the cloud server. This can be provided using a multi-factor authentication scheme. The aim of this paper is to present a key establishment scheme and authentication scheme for RTHS, which satisfies essential security and efficiency requirements and maintains low communication and computation overhead.

1.2 Our contributions

- We present a scenario for Cloud-IoT-based Healthcare system along with various threats and security model.
- We propose a secure authentication protocol based on ECC for remote patient monitoring in Cloud-IoT environments. The protocol is a multi-factor protocol because it uses three different factors for preserving user identity: password, smart card and biometrics. The use of biometrics increases the security of the protocol because biometrics is difficult to forge or steal or forget.
- We prove our scheme secure using a formal proof and analysis.
- We simulate our scheme using AVISPA tool for the formal security analysis and demonstrate that the proposed scheme is secure against active and passive attacks.
- We perform a comparative evaluation of our scheme with some latest schemes in terms of communication and computational overheads.

1.3 Paper organization

The rest of the paper is organized as follows: Sect. 2 discusses the work related to user authentication in healthcare applications. Next, we demonstrate the attack model and different security requirements in Sect. 3. We then discuss some preliminaries of ECC, one way hash functions in Sect. 4. The proposed authentication protocol is discussed in Sect. 5. Section 6 discusses the security and performance analysis of the proposed protocol. Section 7 gives a detailed formal veri-

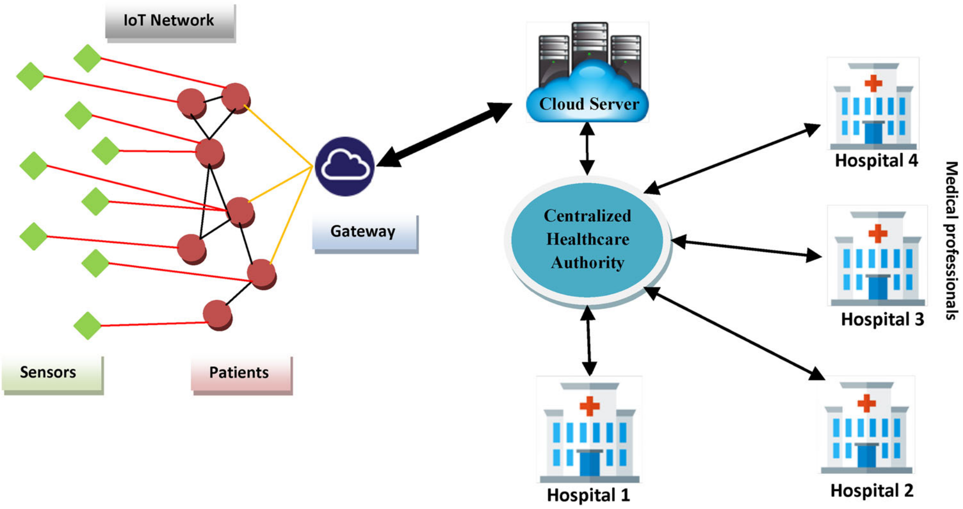


Fig. 1 Cloud-IoT healthcare service architecture

fication of the proposed protocol using AVISPA. Section 8 presents the comparison of the protocol with other related schemes proposed in literature. Finally, Sect. 9 concludes.

2 Related work

Cloud computing and IoT promise an innovative paradigm shift which will allow interconnecting several sensors, smart devices to gather and share data for observation and interpretation. This evolving merger offers a wide range of potential applications that can improve the quality of people's life. The most promising and upcoming potential application is real time remote patient healthcare monitoring and tracking, where a remote patient's health related data gets accumulated with the help sensors, which is then delivered through internet and can be accessed by healthcare professionals for analysis and evaluation of patient's health. In this section, we discuss the existing studies carried out by researchers for providing authentication in remote patient healthcare monitoring and extract out the limitations of the existing work.

In 2004, Watro et al. [2] developed a public-key-based protocol for authentication. The protocol allowed exchange of data between a sensor network and a third party as well as between two sensor networks.

In 2005, Benenson et al. [3] developed a user authentication protocol based on elliptic curves which is robust to several attacks and handles the sensor node capture attack properly. However, the protocol fails to offer mutual authentication, data confidentiality, integrity, and is also vulnerable to DoS attacks, node compromise attacks.

In 2006, Wong et al. [4] proposed a user authentication protocol for WSNs. The protocol is lightweight and is composed of registration, login and authentication phases. It provides security from replay and impersonation attacks; however, it does not provide mutual authentication, data confidentiality, secrecy and scalability and doesn't provide resistance against attacks such as stolen verifier attacks, and sensor node compromise attacks.

In 2007, Tseng et al. [5] developed a user authentication protocol having password change phase based on Wong et al.'s [4] protocol. They claimed that Wong et al.'s protocol cannot resist replay attack, forgery attack, stolen-verifier attack, sensor node revealing and exposing the password to the other node and has password change phase. Still their protocol fails against several potential attacks such as stolen verifier attack and DoS attack and does not provide mutual authentication. Hu et al. [6] created a real-time healthcare monitoring system for cardiac patients. In their architecture, patient's ECG signals gets collected automatically to a server from where the professional can access the data for further analysis and generating reports. The proposed architecture offers data confidentiality and integrity; however, strong user authentication is lacking.

In 2009, Das [7] gave an authentication protocol for healthcare based on wireless sensor networks. The protocol is defenceless against several attacks such as node bypass, user impersonation, and insider attack and also does not provide message confidentiality, and mutual authentication. Huang et al. [8] designed a secure architecture for sensor-based healthcare monitoring. Their hierarchical system, however, lacks strong user authentication, which is critical for remote healthcare services. Malasri et al. [9] presented an ECC-based key agreement protocol for mote-based medical sensor network-based healthcare services. In this protocol, two-tier architecture is employed to authenticate access to patient data. The scheme provides sufficient security to patient's data but it does not provide strong authentication for health professional that can access patient's data which can open to backdoor to attackers. Sriram [10] designed a security framework for securing remote health monitoring systems based on sensor networks. Their proposed architecture provides secure exchange of patient's data across the sensor network. The protocol uses patient's ECG pattern to uniquely identify him/her. However, the protocol lacks strong security check and fails to provide mutual authentication.

In 2010, Sarier et al. [11] proposed a multi-factor protocol for allowing secure communication between two entities and establishing key agreement with server. Venkatasubramanian et al. [12] presented a protocol for physiological signal-based key agreement for authenticated communication between neighbouring nodes in an MSN. A multi-factor user authentication protocol was proposed by Yuan et al. [13] for WSNs. The protocol was resistant to replay attacks, denial attacks, and forgery attack but was susceptible to insider attack, DoS attacks and impersonation attacks. Also it did not provide mutual authentication, data integrity, password change and key agreement.

In 2011, Chen et al. [14] proposed an authentication protocol based on ECC suitable for applications having strong security needs. However, the protocol fails to provide stronger security for low power devices. Le et al. [15] proposed an access control protocol based on ECC that allowed mutually authenticated professionals to access patient's data. The protocol is defensive against replay and denial-of-service attacks. However, it is vulnerable to leakage attacks and, hence, can pose a serious risk to patient's privacy, therefore not suitable healthcare. Yeh et al. [16] found that Chen et al.'s [17] protocol did not allow the user to update password and was susceptible to insider attack and other attacks. They then proposed a new improved authentication protocol for WSNs based on elliptic curve cryptography (ECC). Yoon et al. [18] proposed an improved user authentication for WSNs. They also reviewed Yuan et al.'s [13] protocol and found that it did not provide data integrity. But their protocol also had no key agreement and was susceptible to compromise attacks.

In 2012, Drira et al. [19] developed a scheme for MSNs for providing hybrid authentication and shared session key agreement based on symmetric cryptography. The protocol is defenceless to several potential threats such as password guessing and denial of service. He et al. [20] gave a lightweight protocol with better performance and prevent malicious behaviours. However, the protocol does not provide forward secrecy, scalability and is also susceptible to forgery attacks, denial attacks and password guessing attacks. Kumar et al. [21] proposed an authentication protocol for WMSNs in healthcare applications. It is suited for use in hospitals, home cares, and clinic applications which are based on WMSNs. The protocol offers user authentication, data confidentiality and allows generating shared session key; however, their scheme cannot withstand offline password guessing attacks and insider attacks. Also, it lacks user anonymity and scalability. Zhang et al. [22] gave a biometric-based lightweight protocol for authentication and the protocol offers efficient key generation with lesser overheads. However, their protocol fails to provide forward secrecy and message confidentiality.

In 2013, also, Ohood et al. [23] cryptanalysed the authentication protocol proposed by Yoon et al.'s [18] scheme and found that the protocol has no key agreement, does not provide message confidentiality, and vulnerable to DoS attacks and compromised node attacks. They presented an improved user authentication protocol based on biometrics for wireless sensor networks to improve Yoon et al.'s [18] scheme. Barua et al. [24] developed a patient centric scheme for information sharing based on cloud paradigms. The scheme is defensive to several potential threats but lacks forward secrecy, confidentiality and scalability. Divi et al. [25] gave medical sensor network and cloud-based model to observe and analyse range of bio-medical situations by also fulfilling the proposed security objectives for providing secure medical services. Li et al. proposed an authentication protocol for secure key exchange and management in MSNs [26]. The protocol was based on Group Device Pairing (GDP) which successfully builds up initial trust among devices in MSN devices. But the protocol proved susceptible to impersonation attack and denial of service attack. A three-party authentication protocol was proposed by Lv et al. [27] for allowing secure communication between two entities and establishing key agreement with server. However, the protocol lacks stronger authentication and is susceptible to server impersonation attack and provides no user anonymity. Shi et al. [28] also developed a authentication protocol which used different received signal strength variations. For the authentication purpose, the variations are recorded between an on-body communication and an off-body channel, which is difficult to forge by adversaries. Xue et al. [29] also proposed a temporal credentials-based authentication protocol for WSNs. However, it suffers from offline password guessing attack, user impersonation attack, sensor node impersonation attack and modification attack; also, it fails to provide user anonymity. Wenbo et al. [30] proposed a user authentication protocol based on ECC for WSNs to remove the weaknesses of Yeh et al.'s [16] protocol and found it susceptible to replay attacks, user impersonation attacks, and gateway impersonation attacks. However, the improved protocol still lacks mutual authentication and is not resistant to various attacks such as insider attack, forgery attack and DoS (denial of service) attack.

In 2014, Almashaqbeh et al. [31] laid down a cloud-based framework for remote patient health monitoring by integrating WBANs with the Cloud. Han et al. [32] studied Yeh et al.'s [16] protocol and found that it failed to provide basic performance requirements of a security protocol. They presented a protocol for handling data confidentiality issues and providing secure communication between the cloud and MSNs; however, the protocol lacks strong authentication and does not provide data integrity. Mishra et al. [33] proposed an authentication protocol for session initiation protocol. The improved scheme was then proposed which provided high degree of security to several potential threats with lesser computational overheads. However, the protocol did not provide forward secrecy and scalability and was also defenceless to guessing attacks. Tan et al. [34] also developed a multi-factor protocol for telecare medical information system. The protocol has mutual authentication and offers security from several threats. However, the protocol is exposed to security threats such as node replication and denial of service. A protocol for secure remote patient monitoring based on clouds was proposed by Thilakanathan et al. [35]. In this the authors elaborated a secure framework sensor data sharing over Cloud suitable for mobile healthcare applications. The protocol uses Elgamal encryption and is resistant to several attacks; however, it does not offer mutual authentication. Xu et al. [36] presented an IoT-based system for accessing data ubiquitously for emergency medical scenarios. In the model, the data can be stored securely and in order to access the data a resource-based data access method is provided. Zhao [37] proposed ECC-based authentication protocol for WBANs that uses identity-based authentication. However, the protocol lacks confidentiality, forward secrecy and user anonymity. A hybrid MAC protocol for medical sensor networks is proposed by Ullah et al. [38]. The proposed protocol is highly secure and uses security key set to avoid any malicious network access. The protocol is also vulnerable to several potential threats.

In 2015, Yang et al. [39] presented an adaptive key evolving authentication scheme for remote healthcare applications. The scheme establishes a shared key and also provides mutual authentication. However, the protocol could not provide stronger identity check and is vulnerable to user impersonation attack and DoS attacks. Shankar et al. [40] proposed an ECC-based secure key distribution and data exchange protocol for healthcare WBANs. The protocol is secure to replay attack as it uses timestamps. Their scheme, however, lacks data confidentiality and user anonymity. Quan et al. [41] reviewed Wenbo et al.'s [30] protocol and identified security weaknesses. Furthermore, they proposed an enhanced protocol using identity-based cryptography for user authentication. However, their improved protocol fails to provide user anonymity and is defenceless against several potential threats such as insider attack, and gateway node impersonation attack. Lu et al. [42] reviewed Arshad et al.'s [43] multi-factor protocol for remote medical systems and claimed it to be non-resistant to offline password guessing attack and developed an enhanced protocol to provide resistance to attacks and advanced security. Hossain et al. [44] gave an ECG health monitoring service based on IoT in the cloud. They have presented a HealthIIoT framework for securely transmitting patient data from sensor nodes to the cloud in unattended wireless environment. Chen et al. [17] proposed a protocol based on a two-factor authentication to provide mutual authentication, data integrity, forward secrecy and is resistant to several attacks such as replay attack, impersonation attack, and password guessing attacks. However, the protocol failed to provide data confidentiality, forward secrecy and password updating phase. Amin et al. [45] studied Mishra et al.'s [33] and Xu et al.'s [46] protocols for authentication in TMIS (Telecare Medical Information Systems) to uncover several flaws. The authors also developed an enhanced authentication protocol but their protocol was susceptible to stolen verifier attack, insider attack and server spoofing attacks.

In 2016, Liu and Chung [47] proposed a user authentication scheme to allow medical personnel to continuously monitor the patients, and provide them with medical care. The protocol is based on smart cards and passwords. Moosavi et al. [48] propose an end-to-end security scheme for mobility enabled healthcare Internet of Things (IoT).

In 2017, Wu et al. [49] proposed a novel hash-based lightweight authentication scheme based on cloud for ehealthcare applications. In comparison to previous schemes, their scheme seems more suitable in ehealthcare applications. Dhillon and Kalra [50] proposed a lightweight biometric-based remote user authentication and key agreement scheme for secure access to IoT services. The protocol uses hash operations and XOR operation. However, Li et al. [51] crypt-analysed Liu–Chung's [47] scheme and found it vulnerable to several potential threats such as Sense Data Disclosure Attacks, and Sense Data Forgery Attacks. The authors then proposed an improved authentication and data encryption scheme for the IoT-based medical care system. The authors proved the security of their scheme using random oracle model under ECDHP. Dhillon and Kalra [52] proposed a multi-factor remote user authentication and key agreement scheme for IoT environments. Using this protocol, any authorized user can access and gather real-time sensor data from the IoT nodes. Table 1 summarizes the reviewed literature.

2.1 Limitations of previous studies

- Most of the protocols do not address the issue of MP accessing patient's data from the cloud server. They have considered authentication between the sensor node and gateway node.
- Most of the protocols suffer from security weaknesses, thereby making them impractical for healthcare applications.
- The protocols are mostly based on two factors such as passwords and smart cards, where passwords are easy to forget and smart cards are prone to thefts.
- Most of the schemes lack formal verification using any tool like AVISPA, Proverif, etc.
- Most of the protocols fail to provide mutual authentication and user anonymity.

Table 1 Literature survey summary

Author	Limitations
Benenson et al. [3]	Doesn't provide mutual authentication
	Lacks data confidentiality
	Lacks integrity
	Susceptible to denial of service attack
	Susceptible to node compromise attacks
Wong et al. [4]	Doesn't provide mutual authentication
	Lacks data confidentiality
	Lacks forward secrecy
	Lacks scalability
	Susceptible to stolen verifier attack
	Susceptible to sensor node compromise attack
Tseng et al. [5]	Lacks mutual authentication
	Susceptible to stolen verifier attack
	Susceptible to Denial of service attack
Hu et al. [6]	Lacks strong user authentication
	Susceptible to user impersonation attack
	Susceptible to insider attack
Das [7]	Susceptible to node bypass attack
	Susceptible to user impersonation attack
	Susceptible to insider attack
	Lacks message confidentiality
	Lacks mutual authentication
Huang et al. [8]	Lacks strong user authentication
	Susceptible to denial of service attack
	Susceptible to user impersonation attack
Malasri et al. [9]	Lacks strong user authentication
	Susceptible to privileged insider attack
Sriram [10]	Lacks stronger security check
	Provides no mutual authentication
Yuan et al. [13]	Lacks mutual authentication
	Susceptible to insider attack
	Susceptible to denial of service attack
	Susceptible to user impersonation attack
	No key agreement phase
	No password change phase
Chen et al. [14]	No password change phase
	Susceptible to insider attack
	Susceptible to denial of service attack
	Susceptible to user impersonation attack
Le et al. [15]	Susceptible to leakage attacks
	Doesn't protect patient's privacy
Yeh et al. [16]	Susceptible to replay attacks
	Susceptible to user impersonation attacks
	Susceptible to gateway impersonation attacks
Chen et al. [17]	Lacks data confidentiality
	Lacks forward secrecy
	Lacks password update phase

Table 1 continued

Author	Limitations
Yoon et al. [18]	No key agreement No message confidentiality Susceptible to DoS attacks Susceptible to Compromised node attacks
Drira et al. [19]	Susceptible to Password guessing attacks Provides no mutual authentication
Kumar et al. [21]	Susceptible to offline password guessing attacks Susceptible to insider attacks Lacks user anonymity Lacks scalability
Zhang et al. [22]	Provides no forward secrecy Provides no message confidentiality
Barua et al. [24]	Lacks forward secrecy Lacks confidentiality Lacks scalability
Li et al. [26]	Susceptible to user impersonation attacks Susceptible to Denial of service attacks
Lu et al. [27]	Susceptible to server impersonation attacks
Xue et al. [29]	Susceptible to offline password guessing attacks Susceptible to user impersonation attacks Susceptible to modification attacks Lacks user anonymity
Wenbo et al. [30]	Lacks mutual authentication Susceptible to denial of service attacks Susceptible to Password guessing attacks
Han et al. [32]	Lacks strong user authentication Lacks data integrity
Mishra et al. [33]	No forward secrecy Lacks scalability Susceptible to password guessing attacks
Tan et al. [34]	Node replication attacks Susceptible to denial of service attacks Susceptible to Password guessing attacks Lacks data confidentiality
Thilakanathan et al. [35]	No mutual authentication Susceptible to denial of service attack, Susceptible to user impersonation attack
Zhao et al. [39]	Lacks stronger identity check Susceptible to user impersonation attacks Susceptible to Denial of service attacks
Shankar et al. [40]	Lacks data confidentiality Lacks user anonymity
Quan et al. [41]	Lacks user anonymity Susceptible to insider attacks Susceptible to gateway node impersonation attacks Susceptible to user impersonation attacks

Table 1 continued

Author	Limitations
Amin et al. [45]	Susceptible to stolen verifier attacks
	Susceptible to insider attacks
	Server spoofing attacks
Liu and Chung [47]	Susceptible to stolen smart card attacks
	Susceptible to password guessing attacks
	Susceptible to sensor data disclosure attacks
	Susceptible to sensor data forgery attacks

3 Attack model and security issues

Like any system, public cloud-IoT system must be secured against the common adversaries such as spammers, hackers, and malware. An adversary refers to any malicious entity which intrudes the system with the aim to prevent the legitimate users from achieving their goals of privacy, integrity, and availability of data. He might attempt to access secret data, manipulate the data in the system, and spoof the identity of a legal sender or receiver, and many more. We presume that adversary can step in each and every communication paths and is, therefore, capable of altering or copying messages, replaying them, or injecting false data or messages. This section will summarize the possible threats to the patient's data stored on cloud server and critical security requirements.

3.1 Attack model

Cloud-IoT-based environments face the same set of threats similar to any conventional network. However, due to the huge amount of data that is being stored on the cloud servers, the cloud service providers become an easy and attractive target for the attackers. These threats/attacks may originate from different entities with their adversary models.

(a) *Eavesdropping attack* This attack refers to illegal interception of a communication between two entities. Such attacks can occur when the cloud service provider accesses the data stored on the server out of curiosity. These attacks are menacing since they are difficult to identify and the users unknowingly storing sensitive data such as passwords, on the server.

(b) *Integrity attack* A data integrity attack occurs when an attacker tries to corrupt or manipulate data without permissions of the owner. The attack is usually carried out via malware program that deletes or modifies contents of a smart device.

(c) *Denial attack* In this attack, one of the communicating parties denies either all or some part of the transmission tasks.

(d) *Denial of service attack* This attack happens when a cloud server is flooded by large number of service requests which it cannot handle. It can cause the server to crash and legitimate users are denied from service.

(e) *Cloud server compromise attack* This attack occurs when an attacker gains control of the server after network deployment. An attacker can connect to a server and can completely control it for fetching the information or controlling that server and its further communication.

(f) *Replay attack* This attack takes place when the malicious entity spies the ongoing communication that takes place between the two parties. The malicious entity collects the authenticated information, e.g. shared session key and then tries to contact the receiver later on with that key. The attacker simply replays the eavesdropped message.

(g) *Impersonation attack* In this attack, the attacker tries to impersonate a legal entity and tries to communicate with the other entity as a legitimate entity.

(h) *Stolen verifier attack* In such attacks, the attacker is successful in stealing vital information from server either from the present or previously successful sessions. The attacker can use the stolen information to gain access to the data stored on the server.

(i) *Insider attack* Such attacks occur when the attacker is a trusted entity having authorized admittance to the system and also has all understanding of the underlying architecture. Such attacks are carried with an intention to do a fraud, theft of secret information or of intellectual property.

(j) *Man-in-the-middle attack* Such attacks occur when the attacker is able to secretly transmit and also change the communication taking place between two entities who think they are communicating with each other.

3.2 Security requirements

To augment the inherent security for remotely monitoring of patients for being suitable to various applications and services, we have identified several security requirements to be taken care of while building a secure authentication protocol. These requirements are defined as follows:

(a) *Mutual authentication* This requirement states that before the patients' data are accessed by an MP from the cloud server, authentication should occur between cloud server and the MP. The two-way authentication is a process in which the communicating parties authenticate simultaneously.

(b) *Confidentiality* This requirement states that the secret information must be transmitted in a secure manner over the communications. For that reason, the data from the sensors in an IoT network, e.g. health data collected from patients, must be transmitted in an encrypted form onto the cloud so that only the recognized data consumers can recognize it. Also, when the data consumers try to fetch the patient's data stored on cloud, data are accessed in an encrypted manner.

(c) *Anonymity* This requirement states that the adversary must not be able trace any sensor data using interactions with it. In case the exchanged sensor data do not satisfy anonymity, the attacker having same provider will easily track owner of a specific sensor or be able to discover the location of the data owner.

(d) *Availability* Authentication process must be executed every time whenever the data consumer tries to access sensitive information stored on the cloud about the data owner. Availability ensures that the data consumer must be able to access all the time from the cloud service provider.

(e) *Forward security* This requirement states that the information transmitted previously is untraceable using the currently transmitted information. If the previously exchanged messages are easy to be traced using the intercepted information, it can result in serious privacy risks.

(f) *Scalability* Scalability enables a system to handle growing amounts of work in a graceful manner. The Cloud-IoT system must provide opportunity for the IoT networks to scale their computing resources whenever they deem it necessary. Hence, the computational workload must be sustained by the cloud with the increase in the sensors in the IoT networks.

4 Preliminaries

4.1 Notations

The notations used in the scheme are listed in Table 2.

Table 2 Notations used

Symbol	Description
p	Large prime number
$\mathrm{Z_n}$	Finite field
E	Elliptic curve
G	Generator point on elliptic curve E having order q
MP	Medical professional
CS	Cloud server
$\mathrm{ID_{MP}}$	Identity of MP
PW	MP's password
$\mathrm{B_{MP}}$	MP's biometric imprint
$\mathrm{BIO_{MP}}$	Perceptually hashed biometric
$\mathrm{E_K(m)}$	Encryption operation using K
$\mathrm{D_K(c)}$	Decryption operation using K
$\oplus$	XOR operation
$\|$	Concatenation operation
$h(.)$	Perceptual hash function
$H(.)$	One-way hash function
$T_1, T_2, \mathrm{T_{curr}}$	Timestamps
a, u, y_{MP}	Random numbers

4.2 Elliptic curve cryptography

Elliptic curve cryptography (ECC) is a type of public key cryptography which can be used to achieve a high degree of security in constrained devices. Compared to other existing asymmetric techniques, ECC needs small keys and guarantees high security. The security levels are more significant with bigger key sizes, for instance, a symmetric key of 256-bit needs to be protected using more than 15,000-bit RSA, whereas the similar degree of security can be provided using a 512 bits asymmetric ECC. ECC with smaller key size allows for saving cost in terms of memory and processing power needed. This makes ECC highly recommended for designing compact and faster implementations of the cryptographic operations that can perform well on constrained tiny chips. Due to this lesser amount of heat is produced and lesser amount of processing power is consumed which makes it highly suitable for implementation in resource-constrained devices.

4.2.1 Domain parameters of ECC

Elliptic curve is a plane curve defined over a finite field having points that satisfies the equation $y^2 = x^3+ax+b$ over $F(q)$. It also has a distinguished point at infinity, denoted ∞. Before beginning any communication, the communicating parties must decide on all the domain parameters of the scheme, which define the elliptic curve [53].

- $F(q)$: represents the finite field defined over q a prime number and represents the size of finite field.
- h : Cofactor, $h = \#F(q)/n$.
- (a, b) : The parameters of E elliptic curve $F(q)$.
- $G(x_G, y_G)$: Represents a generator point which is an element of the curve but $G \neq 0$.
- n : generator point G order.

4.3 One-way hash function

One way hash function converts messages or texts of variable long length into a fixed sized string of digits. They are impossible to invert, i.e. it is difficult to recover the original text from the hash value. They are mostly used to generate digital signatures, which are used to identify and authenticate the sender. Even a slight change in the input value leads to a different hash value. The hash functions produce hash values of 128 bits and higher.

4.4 Perceptual hashing

When using biometrics for user authentication schemes, the standard encryption or hashing algorithms cannot be used to encrypt the biometric template. This is because biometric data, e.g. fingerprint, voice, etc. changes with time and environment. To resolve this issue, researchers have suggested using Perceptual Hashing (P-Hash) [54]. In this approach, a hash value is computed for a multimedia data and it remains more or less the same if the content is not modified significantly. The benefit of using P-Hash is that it can tolerate minor variation in quality and format of the input. The size of the hash value generated by perceptual hashing varies from 64 to 128 bits. The process of perceptual hashing is shown in Fig. 2.

5 Proposed protocol for sensor data

The proposed cloud-IoT-based healthcare scenario of Fig. 1 consists of seven entities, particularly in the authentication protocol. These are patients, medical professionals, sensor nodes and cloud server.

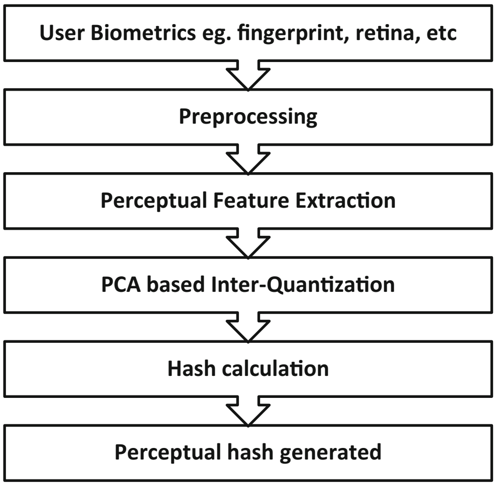

User Biometrics eg. fingerprint, retina, etc
Preprocessing
Perceptual Feature Extraction
PCA based Inter-Quantization
Hash calculation
Perceptual hash generated

Fig. 2 Perceptual hashing process [54]

(a) *Patient* is a passive entity receiving a particular treatment and registered with the healthcare authority to receive medical supervision.
(b) *Sensor nodes* are active entities and are small tiny sensors which are deployed onto patient's body for observing health related figures like BP, temperature, heart beat rate, etc.
(c) *A wireless sensor network (WSN)* also referred to as wireless sensor and actuator networks (WSAN) is active entities and consists of sensors which are spatially scattered and independent and which continuously or on demand monitor and track the environment conditions and pass the observed data to cloud server.
(d) *Patient gateway* refers to a node providing access to another network using different protocols and allows transmitted data to use its routing paths.
(e) *Medical professional* (*MP*) can be either doctor, surgeons, nurses, etc. who can access patient's information through Cloud-IoT framework.
(f) *Healthcare Authority* (*HA*) is the primary entity which provides quality healthcare services, assures collecting, analysis and disseminating health related information to its registered patients through its registered set of MPs.
(g) *Cloud server* the main server playing the major role of storing the healthcare data of the patient, the MP can access the data by logging into the cloud server and once the server authenticates he/she can access it.

The authentication protocol proposed in this paper will allow the MPs to securely gain access to the patients' health data stored on the cloud server. The notations used in the paper are shown in Table 2. The proposed protocol is composed of four phases:

Phase 1: patient registration phase,
Phase 2: MP registration phase,
Phase 3: pre-computation and Login phase,
Phase 4: authentication phase.

5.1 Phase 1: patient registration phase

The proposed protocol requires the patient to register at the healthcare authority which is the registration center at the hospital. To successfully register, patient sends a registration request message along with his name and medical diagnosis to the healthcare center. The healthcare authority selects the required sensor kit as per the diagnosis of patient's condition and allocates suitable MPs. The healthcare authority also generates a unique identity for the patient and supplies the medical kit along with the unique identity to the patient. A technician from the hospital then deploys the sensors onto the patient's body.

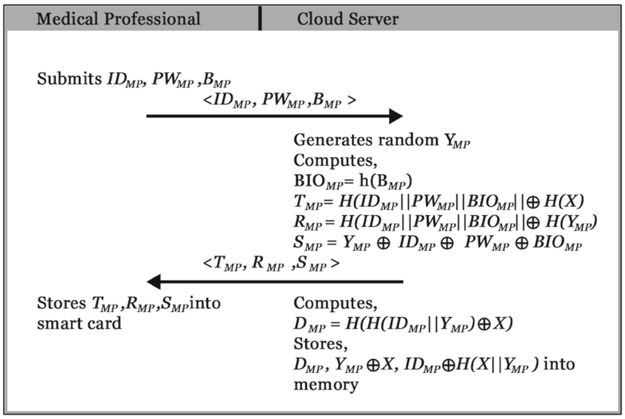

Medical Professional
Cloud Server
Submits ID_{MP}, PW_{MP} ,B_{MP}
<ID_{MP}, PW_{MP} ,B_{MP} >
Generates random Y_{MP}
Computes,
BIO_{MP}= h(B_{MP})
T_{MP}= $H(ID_{MP}||PW_{MP}||BIO_{MP}|| \oplus H(X)$
R_{MP} = $H(ID_{MP}||PW_{MP}||BIO_{MP}|| \oplus H(Y_{MP})$
$S_{MP} = Y_{MP} \oplus ID_{MP} \oplus PW_{MP} \oplus BIO_{MP}$
<T_{MP}, R_{MP} ,S_{MP} >
Stores T_{MP},R_{MP},S_{MP}into smart card
Computes,
$D_{MP} = H(H(ID_{MP}||Y_{MP}) \oplus X)$
Stores,
D_{MP}, $Y_{MP} \oplus X$, $ID_{MP} \oplus H(X||Y_{MP})$ into memory

Fig. 3 Registration phase for medical professional

5.2 Phase 2: medical professional registration phase

The MP who is the active user in the proposed protocol needs to get himself/herself registered with the Healthcare Authority (HA). The HA will generate a suitable security key information for the MP. The process is shown in Fig. 3.

(a) Over the secure channel the MP will submit his identity $\mathrm{ID}_{\mathrm{MP}}$, password $\mathrm{PW}_{\mathrm{MP}}$ and biometric information B_{MP} to cloud server.
(b) Next, the cloud server will compute perceptual hash of input biometric B_{MP} as $\mathrm{BIO}_{\mathrm{MP}} = h\left(B_{\mathrm{MP}}\right)$. It also generates a random number y_{MP}. It then calculates $T_{\mathrm{MP}} = H(\mathrm{ID}_{\mathrm{MP}}\|\mathrm{PW}_{\mathrm{MP}}\|\mathrm{BIO}_{\mathrm{MP}}) \oplus H\left(X\right)$ and $R_{\mathrm{MP}} = H(\mathrm{ID}_{\mathrm{MP}}\|\mathrm{PW}_{\mathrm{MP}}\|\mathrm{BIO}_{\mathrm{MP}}) \oplus H\left(y_{\mathrm{MP}}\right)$. Also, the cloud server calculates $S_{\mathrm{MP}} = y_{\mathrm{MP}} \oplus \mathrm{ID}_{\mathrm{MP}} \oplus \mathrm{PW}_{\mathrm{MP}} \oplus \mathrm{BIO}_{\mathrm{MP}}$. It then through a secure channel sends $\langle T_{\mathrm{MP}}, R_{\mathrm{MP}}, S_{\mathrm{MP}}\rangle$ to MP.
(c) MP will receive a smart card with $\langle T_{\mathrm{MP}}, R_{\mathrm{MP}}, S_{\mathrm{MP}}\rangle$ stored into it.
(d) Next, the cloud server computes $D_{\mathrm{MP}} = H(H(\mathrm{ID}_{\mathrm{MP}}\| y_{\mathrm{MP}}) \oplus X)$ and stores D_{MP}, $y_{\mathrm{MP}} \oplus X$, $\mathrm{ID}_{\mathrm{MP}} \oplus H(X||y_{\mathrm{MP}})$ into its memory.

5.3 Phase 3: pre-computation and medical professional login phase

For accessing patient's healthcare data stored on cloud server, MP, must login into the cloud server and get authenticated first. The MP performs following steps to login:

(a) The medical professional MP uses smart card and submits the login credentials viz. identity, secret password and his/her personal biometric information, i.e. $\mathrm{ID}'_{\mathrm{MP}}$, $\mathrm{PW}'_{\mathrm{MP}}$ and B'_{MP}. It then the perceptual hash computes of the entered biometric as $\mathrm{BIO}'_{\mathrm{MP}} = h\left(B'_{\mathrm{MP}}\right)$.

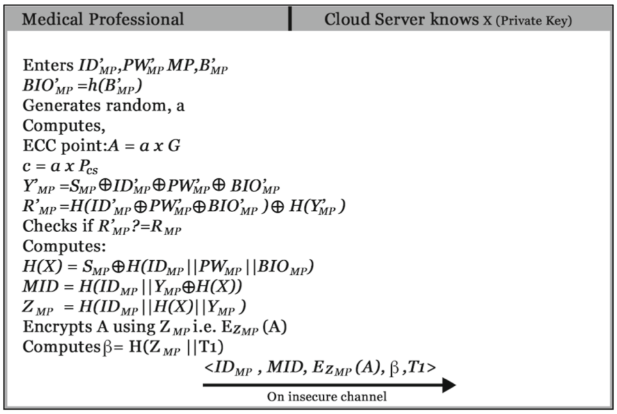

Medical Professional
Cloud Server knows X (Private Key)
Enters ID'_{MP},PW'_{MP} MP,B'_{MP}
BIO'_{MP} =$h(B'_{MP})$
Generates random, a
Computes,
ECC point:$A = a$ x G
$c = a$ x P_{cs}
$Y'_{MP} = S_{MP} \oplus ID'_{MP} \oplus PW'_{MP} \oplus BIO'_{MP}$
$R'_{MP} = H(ID'_{MP} \oplus PW'_{MP} \oplus BIO'_{MP}) \oplus H(Y'_{MP})$
Checks if R'_{MP}?=R_{MP}
Computes:
$H(X) = S_{MP} \oplus H(ID_{MP}||PW_{MP}||BIO_{MP})$
$MID = H(ID_{MP}||Y_{MP} \oplus H(X))$
$Z_{MP} = H(ID_{MP}||H(X)||Y_{MP})$
Encrypts A using Z_{MP} i.e. $E_{Z_{MP}}$ (A)
Computes β= H(Z_{MP} ||T1)
<ID_{MP} , MID, $E_{Z_{MP}}$(A), β ,T1>
On insecure channel

Fig. 4 Login request by medical professional to cloud server

(b) Next, the MP generates random number, a and calculates the ECC point A as $A = a \times G$ and $c = a \times P_{\mathrm{CS}}$.
(c) The MP then calculates $y'_{\mathrm{MP}} = S_{\mathrm{MP}} \oplus \mathrm{ID}'_{\mathrm{MP}} \oplus \mathrm{PW}'_{\mathrm{MP}} \oplus \mathrm{BIO}'_{\mathrm{MP}}$ and $R'_{\mathrm{MP}} = H\left(\mathrm{ID}'_{\mathrm{MP}} \oplus \mathrm{PW}'_{\mathrm{MP}} \oplus \mathrm{BIO}'_{\mathrm{MP}}\right) \oplus H\left(y'_{\mathrm{MP}}\right)$.
(d) The MP then checks if $R'_{\mathrm{MP}}? = R_{\mathrm{MP}}$. If the condition holds, the information entered is correct and it continues further, otherwise the login process gets terminated because some illegitimate user is trying to access the server.
(e) Next, it computes $H\left(X\right) = S_{\mathrm{MP}} \oplus \mathrm{H}\left(\mathrm{ID}_{\mathrm{MP}} \|\mathrm{PW}_{\mathrm{MP}}\| \mathrm{BIO}_{\mathrm{MP}}\right)$ and $\mathrm{MID} = H(\mathrm{ID}_{\mathrm{MP}}\| y_{\mathrm{MP}} \oplus H\left(X\right))$ and also, $Z_{\mathrm{MP}} = H(\mathrm{ID}_{\mathrm{MP}}\| H\left(X\right) \| y_{\mathrm{MP}})$.
(f) Next, it encrypts A using Z_{MP} i.e. $E_{Z_{\mathrm{MP}}}\left(A\right)$ and also computes $\beta = H(Z_{\mathrm{MP}}\|T_1)$. It then forwards the login request $\langle \mathrm{MID}, E_{Z_{\mathrm{MP}}}\left(A\right), \beta, T_1\rangle$ message to the cloud server.

The workflow of the steps is shown in Fig. 4.

5.4 Phase 4: authentication phase

During this phase, the cloud server authenticates the MP. This phase is required so that only a legitimate MP can get access to the sensitive patient data stored on the cloud server. The steps of the process are:

(a) The cloud server will receive login request message and will check if $(T_1 - T_{\mathrm{curr}}) \leq \Delta T$? If the check doesn't hold, the login process gets terminated. Otherwise, cloud server computes $D'_{\mathrm{MP}} = H\left(\mathrm{MID} \oplus H\left(X\right) \oplus \mathrm{X}\right)$. This check allows handling the message replay attacks.
(b) Next, it checks for the condition if $D'_{\mathrm{MP}} = D_{\mathrm{MP}}$? If the condition fails, the server terminates the process, else the cloud server calculates $Z'_{\mathrm{MP}} = H\left(\mathrm{ID}_{\mathrm{MP}} \| H\left(X\right)\| y_{\mathrm{MP}}\right)$ and $\beta' = H(Z'_{\mathrm{MP}}\|T_1)$ to verify whether the message has been send by a legal MP.

(c) Next, the cloud server checks for the condition if $\beta' = \beta$? This check again allows taking care of the message replay attacks, since if the value of the timestamp got modified the condition will fail to hold and the cloud server will cancel the login request by rejecting the login message. If not, the cloud server will decrypt A using Z'_{MP}, i.e. $D_{Z'_{\mathrm{MP}}}\left(E_{Z_{\mathrm{MP}}}(A)\right)$ to extract A.

(d) The cloud server will compute $c = A \times X_{CS}$ and $L = H(A||T_2)$. It next generates a random number u to compute $\gamma_{\mathrm{CS}} = H(c\|\mathrm{u}\|Z'_{\mathrm{MP}}\|T_2)$. It then transmits the message to the MP's smart device, i.e. $\langle\gamma_{\mathrm{CS}}, u, L, T_2\rangle$ and computes session key i.e. $S_K = H(H(X)\,||Z'_{\mathrm{MP}}\,\|c\|\,u)$.

(e) The MP will receive the message $\langle\gamma_{\mathrm{CS}}, u, L, T_2\rangle$. The smart device will then check for the condition if $(T_2 - T_{\mathrm{curr}}) \le \Delta T$ satisfies or not. In case the condition doesn't satisfies, the request message is rejected, since it's a previously intercepted message replayed again by the illegitimate user. Otherwise, the smart device computes $L' = H(A||T_2)$.

(f) Next, the smart device checks for the condition if $L' = L$? if the condition holds, the message is sent by a legitimate cloud server otherwise the process terminates indicating that the message has been intercepted and modified during transit.

(g) If successful, the smart device computes $\gamma'_{\mathrm{CS}} = H(c\|\mathrm{u}\|Z_{\mathrm{MP}}\|T_2)$ and checks if the condition holds or not, i.e. $\gamma'_{\mathrm{CS}} = \gamma_{\mathrm{CS}}$? If the condition fails, the message is rejected, otherwise, the device calculates shared session key as $S_K = H(H(X)\,||Z_{\mathrm{MP}}\,\|c\|\,u)$.

Figure 5 shows the authentication process.

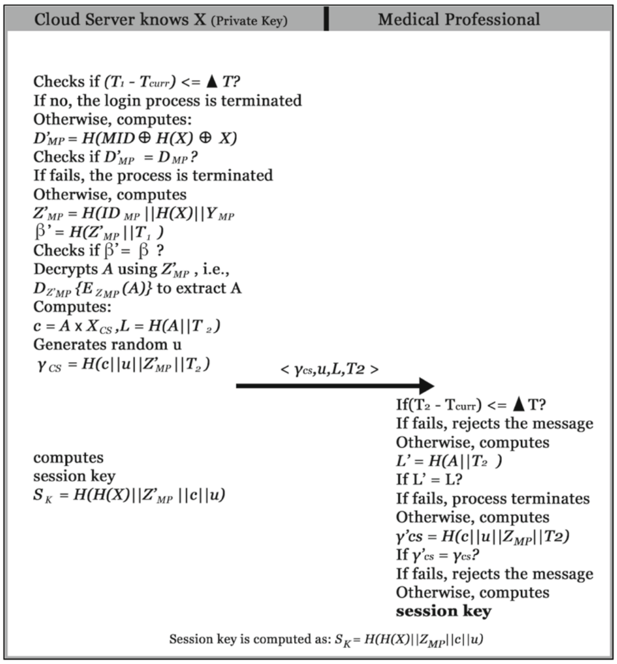
6 Performance and security analysis

This section discusses the informal protocol verification against the security and performance requirements specified in the Sect. 3. Although achieving security of authentication protocols is extremely important, however, it is also difficult to accomplish. The proposed authentication scheme establishes a shared session key between the cloud server and the MP also offers mutual authentication between them and is defensive to all the attacks discussed in Sect. 3.1.

6.1 Informal security analysis

6.1.1 Provides mutual authentication

The proposed protocol guarantees mutual authentication between the cloud server and MP. In the proposed protocol, the MP before accessing sensitive IoT sensor node data about a patient mutually authenticates with the Cloud server so as to verify the authenticity of the server. The Cloud server authenticates the MP when the condition $D'_{\mathrm{MP}} = D_{\mathrm{MP}}$ holds. Similarly, the MP authenticates the cloud server if $\gamma'_{\mathrm{CS}} = \gamma_{\mathrm{CS}}$ and $L' = L$ holds.



Checks if $(T_1 - T_{curr}) <= \blacktriangle T$?
If no, the login process is terminated
Otherwise, computes:
$D'_{MP} = H(MID \oplus H(X) \oplus X)$
Checks if $D'_{MP} = D_{MP}$?
If fails, the process is terminated
Otherwise, computes
$Z'_{MP} = H(ID_{MP}||H(X)||Y_{MP}$
$\beta' = H(Z'_{MP}||T_1)$
Checks if $\beta' = \beta$?
Decrypts A using Z'_{MP} , i.e.,
$D_{Z'MP}\{E_{ZMP}(A)\}$ to extract A
Computes:
$c = A \times X_{CS}, L = H(A||T_2)$
Generates random u
$\gamma_{CS} = H(c||u||Z'_{MP}||T_2)$

$< \gamma_{cs}, u, L, T2 >$

If$(T_2 - T_{curr}) <= \blacktriangle T$?
If fails, rejects the message
Otherwise, computes
$L' = H(A||T_2)$
If L' = L?
If fails, process terminates
Otherwise, computes
$\gamma' cs = H(c||u||Z_{MP}||T2)$
If $\gamma'_{cs} = \gamma_{cs}$?
If fails, rejects the message
Otherwise, computes
session key

computes
session key
$S_K = H(H(X)||Z'_{MP}||c||u)$

Session key is computed as: $S_K = H(H(X)||Z_{MP}||c||u)$

Fig. 5 Authentication and key agreement phase

6.1.2 Provides confidentiality

The proposed protocol provides confidentiality by transmitting sensitive data in an encrypted form to the cloud server. Transmitting sensitive data without encoding over insecure channel will allow attacker easily observe the ongoing communication. The session key is generated independently by the cloud server and MP. Message confidentiality protects against eavesdropping attacks.

6.1.3 Provides anonymity

The MP communicates with the cloud server in network via open insecure wireless channel. The proposed protocol provides user anonymity by employing multi-factors i.e. biometric information B_{MP} as unique identification of the MP which is impossible to forge, along with the password $\mathrm{PW}_{\mathrm{MP}}$, thereby securing disclosure of any private information even if any illegitimate users eavesdrop the communication.

6.1.4 Provides forward security

Ensuring forward security requires that even if legitimate user's secret key is leaked out it will not compromise the session key generated. In the proposed protocol, if the MP MP's key is compromised; any intruder is still unable to generate the session since, to generate the session key $S_K = H(H(X) \| Z_{\mathrm{MP}} \| c \| u)$, the adversary needs both $\mathrm{ID}_{\mathrm{MP}}$ MP's identity, private key X, parameter c and u. Also, the adversary needs to resolve ECDLP which is a computationally hard problem and he cannot predict c and L. This proves that the proposed protocol provides forward secrecy.

6.1.5 Provides scalability

The proposed protocol provides scalability. This property allows the system to expand. Any number of patients can be added to the system without affecting the system. Since the cloud server does not store or maintain any verifier table or any database of passwords. Hence, the proposed protocol offers scalability.

6.1.6 Efficient login phase

In proposed protocol, during MP login phase, the smart device of a legitimate medical processional verifies the correctness of inputs $\mathrm{ID}_{\mathrm{MP}}$, $\mathrm{PW}_{\mathrm{MP}}$ and B'_{MP}using the condition if $R'_{\mathrm{MP}}? = R_{\mathrm{MP}}$. If the condition satisfies, the smart device executes and the further else terminates the login process. This proves that the proposed protocol effectively finds the validity of data provided by the MP.

6.1.7 Known key secrecy

In proposed protocol, even if the session key $S_K = H(H(X) || Z_{\mathrm{MP}} \| c \| u)$ of previous communications gets leaked to an attacker, he still cannot use it to predict the information of other session keys because each session key is generated using one-way hash functions. Hence, no intelligence gets extracted from the session key.

6.1.8 Key freshness

In the proposed protocol, each established session key $S_K = H(H(X) || Z_{\mathrm{MP}} || c || u)$ includes a fresh random number u. The use of fresh random numbers allows achieving the freshness of the key for every communication session. This allows that an exclusive key is generated every time. Therefore, the exclusiveness ensures the key freshness.

6.2 Resistance to potential attacks

This section explains how the proposed protocol resists against the attacks presented in the threat model in Sect. 3.1.

6.2.1 Resistance to integrity attacks

Data integrity attacks comprise modifying or alteration and insertion of data. Maintaining this property requires that the exchanged data does not gets modified by illegitimate user. In the proposed protocol, the cloud server can verify whether the login request message $\langle \mathrm{MID}, E_{Z_{\mathrm{MP}}}(A), \beta, T_1 \rangle$ got intruded on the way by checking $(T_1 - T_{\mathrm{curr}}) \leq \Delta T$? If the condition does not hold, the login process is terminated. This check allows handling the message replay attacks. Also, the cloud server checks for the condition if $\beta' = \beta$? This check again allows taking care of the message replay attacks, since if the value of the timestamp got modified the condition will fail to pass the check and the cloud server cancels the login request message. Similarly, the MP can verify the authentication response message sent by the cloud server by checking for the condition $(T_2 - T_{\mathrm{curr}}) \leq \Delta T$. If the condition fails to pass the check, the request message gets ignored, since it is a previously intercepted message replayed again by the illegitimate user. Also, the smart device checks for the condition $L' = L$? if the condition holds, the message is sent by a legitimate cloud server otherwise the process terminates indicating that the message has been intercepted and modified during transit.

6.2.2 Resistance to denial attacks

The proposed protocol protects from denial attacks In this attack one of the communicating parties denies either all or some part of the transmission tasks. In the proposed protocol, cloud server is assumed as a trusted third entity which constructs a unique private key for an MP. Although cloud server is not maintaining any table for the private key storage, it can still track the activities using the public key of the party. Hence, any party in communication cannot refuse a transmission.

6.2.3 Resistance to denial of service (DoS) attacks

The proposed protocol is resistant to DoS attacks The DoS attacks occur when an illegitimate user transmits a huge number of messages either to MP or cloud server during the login or authentication phase. The proposed protocol associates timestamps with each transmitted message, i.e. T_1, T_2, etc. If the entity passes the initial check of timestamps, only then the authentication or login process is continued else it is terminated. Any message that is timeout gets rejected. Hence, the proposed protocol can withstand DoS attack successfully.

6.2.4 Resistance to cloud server compromise attacks

The proposed protocol is resistant to cloud server compromise attacks Since the proposed Cloud-IoT-based system operates in an open environment, it becomes an easy target for the attackers. In case the attacker gains access to sensitive information by capturing the cloud server, he/she can attack the Cloud-IoT-based sensor network. If the authenticating MP is allowed direct access to IoT sensor node data without the cloud server, this attack becomes very high. However, in the proposed protocol, MP and cloud server mutually authenticates each other before information exchange. All the information exchanged is encrypted using the secretly generated session key.

6.2.5 Resistance to replay attacks

The proposed protocol protects from replay attacks. The replay attack becomes invalid only if the previously transmitted information cannot be reused again. In the proposed protocol, every transmitted message is validated using timestamps. For instance, if the attacker is able to extract the login message $\langle$MID, $E_{Z_{\mathrm{MP}}}(A)$, $T_{1,}$ β$\rangle$, it can replay the message for logging into cloud server, he/she will fail the verification of the login message since $(T_1 - T_0) > \Delta$T, where T_0 represents the time when cloud server gets the replayed message. Also, every transmitted message has a timestamp associated with it, which help it avoiding the replay attacks.

6.2.6 Resistance to impersonation attacks

The proposed protocol protects from impersonation attacks since all the information is transmitted in an encrypted manner. Also, each entity validates every message it receives by checking for timestamps. The attacker needs to solve computationally hard ECDLP to calculate the private key, which is impossible to extract A. Moreover, for an attacker it is very difficult to access sensitive information since it is stored in an encrypted way in the smart device and it can be accessed only if he knows the MP's password and biometric information. Therefore, it makes it impossible that the attacker is able to generate the session key. Therefore, impersonation is difficult to achieve.

6.2.7 Resistance to stolen verifier attacks

The proposed protocol is resistant to stolen verifier attack. An attacker who has stolen MP's secret key information from the smart device using any intruding methods will not be successful to obtain any meaningful information. This is because the first, the sensitive secret key information is stored in an encrypted form. Second, the attacker needs to impersonate the biometric of the MP, which is impossible to forge.

6.2.8 Resistance to stolen smart device attacks

The proposed protocol is resistant to attacks of smart device theft. Even if the smart device gets stolen the attacker is unable to impersonate a legal MP to gain right to use the data stored on cloud server since attacker does not know the password and biometric information, and the smart device will not legalize the request to login and will reject any login request of the illegitimate user.

6.2.9 Resistance to insider attacks

The proposed protocol prevents insider attacks. This occurs when an authorized entity intentionally misuses its authorization. Any insider of cloud environment cannot gain MP's biometric imprint and also the cloud server does not store any other sensitive information regarding MP or any table. Furthermore, the cloud manager cannot obtain any useful information of MP from the smart device since the sensitive information is stored in encrypted manner. Hence, the protocol is defensive against privileged insider attacks.

6.2.10 Resistance to man-in-the-middle attacks

The proposed protocol protects against man-in-the-middle attack. This attack arises when an attacker is able intercept the communication between a legal MP and cloud server and he/she able to successfully masquerade as legal user to other entities. In the proposed protocol, each of the entity (MP and cloud server) mutually authenticates each other which let the proposed protocol to successfully prevent the attack.

7 Automatic formal verification using AVISPA

The developed protocol is simulated in AVISPA (Automated Validation of Internet Security Protocols and Applications) tool. It is a web-based push button tool using which the formal security verification of the security protocols is carried out. To simulate the protocols the user uses HLPSL (High Level Protocols Specification Language) to write the code for different roles in the protocol. AVISPA consists of a *translator tool* called HLPSL2IF and four back-ends. The *translator tool* is used to convert a protocol written in HLPSL into Intermediate Format (IF). This IF is a general language understood by all the back-ends and is used by different back-ends to test and analyse different properties specified in the protocol. These back-ends are [55]:

- Constraint-Logic-based Attack Searcher (CL-AtSe),
- Onthe-fly Model-Checker (OFMC),
- Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP),
- SAT-based Model-Checker (SATMC).

The structure of the AVISPA tool is shown in Fig. 6.

These back-ends produce the Output Format (OF), having the following parts:

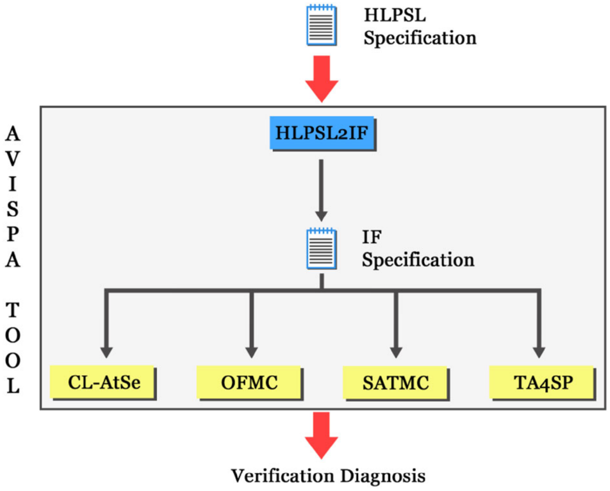


Fig. 6 Structure of AVISPA [55]

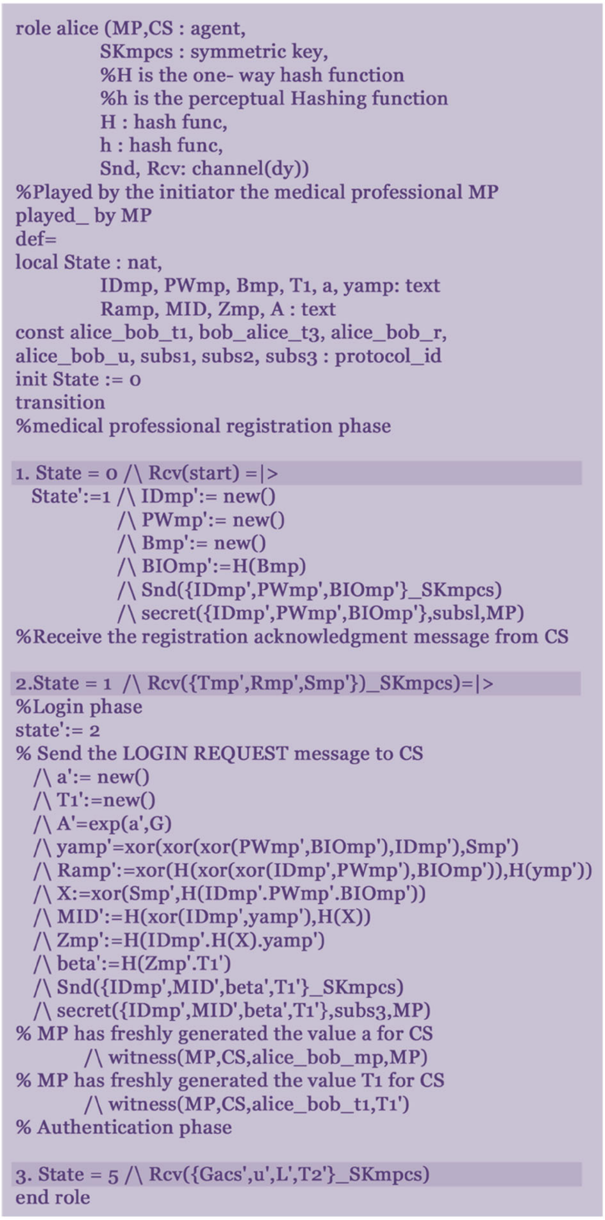
- *Summary* This section tells whether the proposed protocol is safe, unsafe, or whether the analysis is also inconclusive.
- *Details* This section tells under what conditions the proposed protocol was concluded safe, or an attack is found, or why the result was inconclusive.
- *Protocol*, *goal and backend* These sections specify the protocol name, goal of the analysis and the back-end used, respectively.

Several basic types are supported by HLPSL; the commonly used are [55]:

- *agent* principal name. The intruder has always the special identifier *i*.
- *public_key* represents agents' public keys.
- *symmetric_key* represents a symmetric-key.
- *text* represents nonces.
- *nat* represents natural numbers used non-message contexts.
- *const* represents constants.
- *hash_func* represents a cryptographic collision-resistant one-way hash function.

7.1 Specifying our scheme in HLPSL

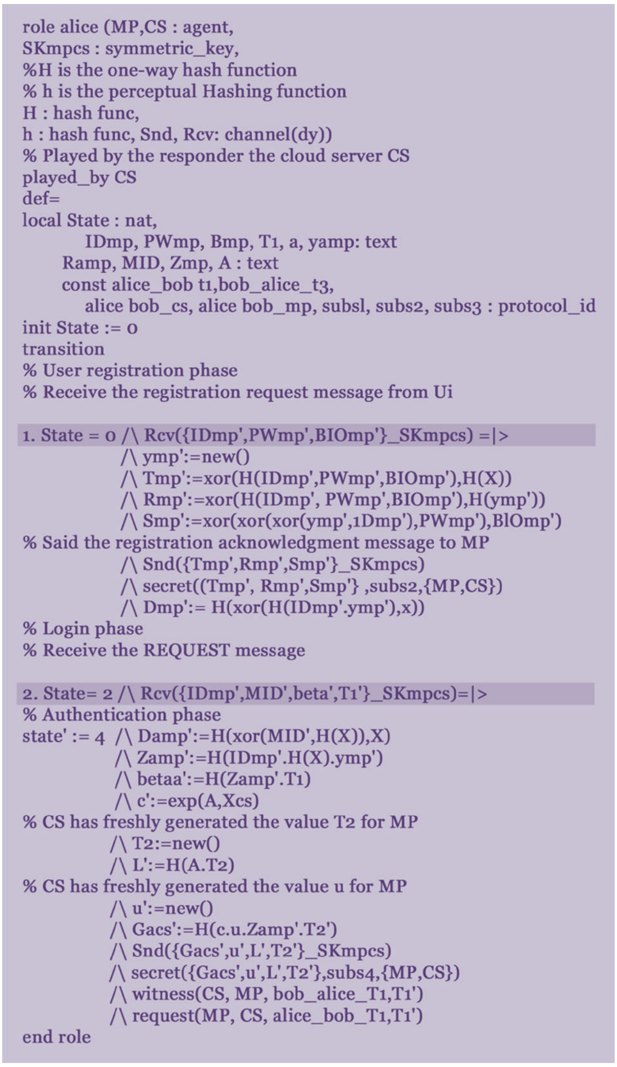
In the proposed protocol, there are two basic roles: MP MP and cloud server CS. Both the roles have been specified in HLPSL. Apart from these basic roles, we have three other roles: the session, environment and goal.



```
role alice (MP,CS : agent,
            SKmpcs : symmetric key,
            %H is the one- way hash function
            %h is the perceptual Hashing function
            H : hash func,
            h : hash func,
            Snd, Rcv: channel(dy))
%Played by the initiator the medical professional MP
played_ by MP
def=
local State : nat,
            IDmp, PWmp, Bmp, T1, a, yamp: text
            Ramp, MID, Zmp, A : text
const alice_bob_t1, bob_alice_t3, alice_bob_r,
alice_bob_u, subs1, subs2, subs3 : protocol_id
init State := 0
transition
%medical professional registration phase

1. State = 0 /\ Rcv(start) =|>
  State':=1 /\ IDmp':= new()
              /\ PWmp':= new()
              /\ Bmp':= new()
              /\ BIOmp':=H(Bmp)
              /\ Snd({IDmp',PWmp',BIOmp'}_SKmpcs)
              /\ secret({IDmp',PWmp',BIOmp'},subsl,MP)
%Receive the registration acknowledgment message from CS

2.State = 1  /\ Rcv({Tmp',Rmp',Smp'})_SKmpcs)=|>
%Login phase
state':= 2
% Send the LOGIN REQUEST message to CS
  /\ a':= new()
  /\ T1':=new()
  /\ A'=exp(a',G)
  /\ yamp'=xor(xor(xor(PWmp',BIOmp'),IDmp'),Smp')
  /\ Ramp':=xor(H(xor(xor(IDmp',PWmp'),BIOmp')),H(ymp'))
  /\ X:=xor(Smp',H(IDmp'.PWmp'.BIOmp'))
  /\ MID':=H(xor(IDmp',yamp'),H(X))
  /\ Zmp':=H(IDmp'.H(X).yamp')
  /\ beta':=H(Zmp'.T1')
  /\ Snd({IDmp',MID',beta',T1'}_SKmpcs)
  /\ secret({IDmp',MID',beta',T1'},subs3,MP)
% MP has freshly generated the value a for CS
        /\ witness(MP,CS,alice_bob_mp,MP)
% MP has freshly generated the value T1 for CS
        /\ witness(MP,CS,alice_bob_t1,T1')
% Authentication phase

3. State = 5 /\ Rcv({Gacs',u',L',T2'}_SKmpcs)
end role
```

Fig. 7 Role specification for the user MP

Figure 7 specifies the role of the MP represented by MP. On receiving the start signal MP will change its state from 0 to 1, and send registration request message ⟨IDmp,PWmp,Bmp⟩ via secure channel to CS. To send the message to CS, it uses the Snd() operation. CS will then perform specified computations and through a secure channel sends the smart card issued for MP. During the login phase, MP will send the login message ⟨IDmp,MID,beta,T1⟩ to CS on an insecure public channel, and then waits for an authentication message ⟨Gacs,u,L,t2⟩ from CS using Rcv() operation. witness(MP, CS, alice_bob_t1, T1) declares that MP has freshly generated the timestamp T1 for CS. witness(MP, CS, alice_bob_a, a') declares that MP has freshly generated the nonce a for CS. request(CS, MP, bob_alice_t1, T1') indicates that MP's acceptance of the timestamp T1 generated by MP by CS in which MP authenticates CS. request(CS, MP, bob_alice_a, a') indicates that MP's acceptance of the



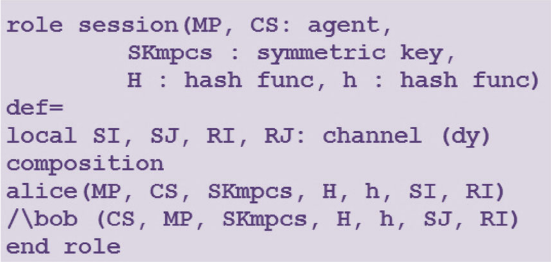
```
role alice (MP,CS : agent,
SKmpcs : symmetric_key,
%H is the one-way hash function
% h is the perceptual Hashing function
H : hash func,
h : hash func, Snd, Rcv: channel(dy))
% Played by the responder the cloud server CS
played_by CS
def=
local State : nat,
         IDmp, PWmp, Bmp, T1, a, yamp: text
      Ramp, MID, Zmp, A : text
      const alice_bob t1,bob_alice_t3,
         alice bob_cs, alice bob_mp, subsl, subs2, subs3 : protocol_id
init State := 0
transition
% User registration phase
% Receive the registration request message from Ui

1. State = 0 /\ Rcv({IDmp',PWmp',BIOmp'}_SKmpcs) =|>
               /\ ymp':=new()
               /\ Tmp':=xor(H(IDmp',PWmp',BIOmp'),H(X))
               /\ Rmp':=xor(H(IDmp', PWmp',BIOmp'),H(ymp'))
               /\ Smp':=xor(xor(xor(ymp',1Dmp'),PWmp'),BlOmp')
% Said the registration acknowledgment message to MP
               /\ Snd({Tmp',Rmp',Smp'}_SKmpcs)
               /\ secret((Tmp', Rmp',Smp'} ,subs2,{MP,CS})
               /\ Dmp':= H(xor(H(IDmp'.ymp'),x))
% Login phase
% Receive the REQUEST message

2. State= 2 /\ Rcv({IDmp',MID',beta',T1'}_SKmpcs)=|>
% Authentication phase
state' := 4  /\ Damp':=H(xor(MID',H(X)),X)
              /\ Zamp':=H(IDmp'.H(X).ymp')
              /\ betaa':=H(Zamp'.T1)
              /\ c':=exp(A,Xcs)
% CS has freshly generated the value T2 for MP
             /\ T2:=new()
             /\ L':=H(A.T2)
% CS has freshly generated the value u for MP
             /\ u':=new()
             /\ Gacs':=H(c.u.Zamp'.T2')
             /\ Snd({Gacs',u',L',T2'}_SKmpcs)
             /\ secret({Gacs',u',L',T2'},subs4,{MP,CS})
             /\ witness(CS, MP, bob_alice_T1,T1')
             /\ request(MP, CS, alice_bob_T1,T1')
end role
```

Fig. 8 Role specification for the CS

nonce generated by MP for CS in which MP authenticates CS.

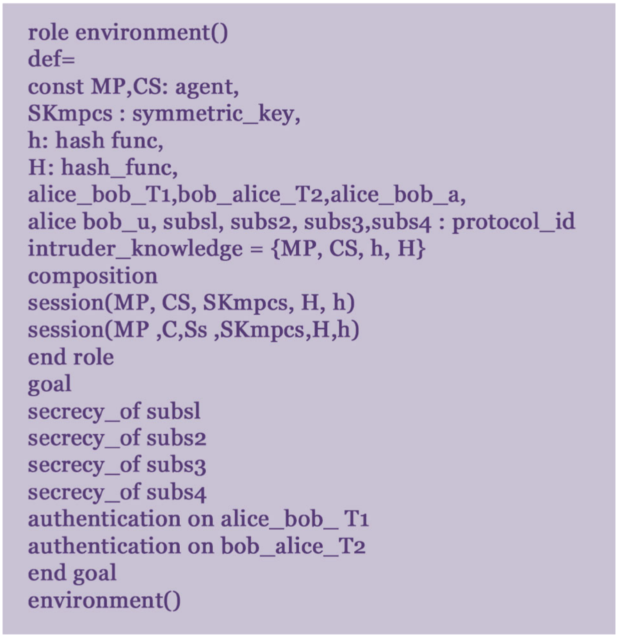
Figure 8 shows the implementation of the proposed protocol for the cloud server CS. During the registration phase, after receiving the message ⟨IDmp,PWmp,Bmp⟩ on secure channel from MP, CS carries out required computations and then sends the smart card through secure channel to MP. CS will then receive the login request message ⟨IDmp,MID,beta,T1⟩. Subsequently, CS will send the authentication message ⟨Gacs,u,L,T2⟩ on an insecure public channel to MP as a reply to the received login message from MP.

Figures 9 and 10 show the role specification for session, and for the goal and environment, respectively. In the session role, all basic roles including the roles for MP and CS are considered as the instances with concrete arguments. The environment role contains the global constants and a composition of one or more sessions. It is assumed that an intruder ‘I’ may also play some roles as the legitimate users. The intruder thus participates in the execution of a protocol as a concrete session. In our implementation, four secrecy goals and two authentications are verified, which are shown in Fig. 10.



```
role session(MP, CS: agent,
         SKmpcs : symmetric key,
         H : hash func, h : hash func)
def=
local SI, SJ, RI, RJ: channel (dy)
composition
alice(MP, CS, SKmpcs, H, h, SI, RI)
/\bob (CS, MP, SKmpcs, H, h, SJ, RI)
end role
```

Fig. 9 Role specification for the session

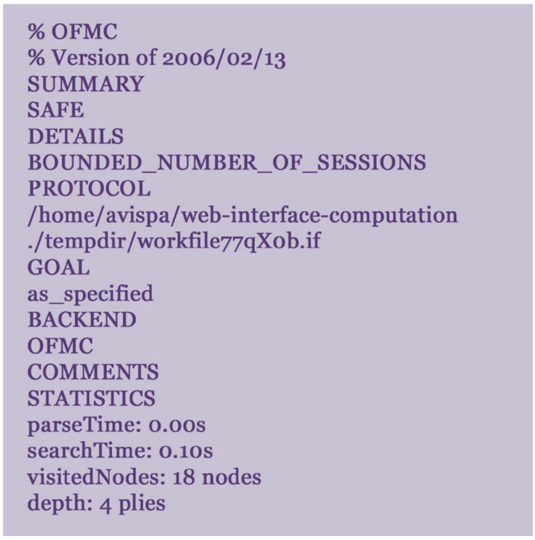


```
role environment()
def=
const MP,CS: agent,
SKmpcs : symmetric_key,
h: hash func,
H: hash_func,
alice_bob_T1,bob_alice_T2,alice_bob_a,
alice bob_u, subsl, subs2, subs3,subs4 : protocol_id
intruder_knowledge = {MP, CS, h, H}
composition
session(MP, CS, SKmpcs, H, h)
session(MP ,C,Ss ,SKmpcs,H,h)
end role
goal
secrecy_of subsl
secrecy_of subs2
secrecy_of subs3
secrecy_of subs4
authentication on alice_bob_ T1
authentication on bob_alice_T2
end goal
environment()
```

Fig. 10 Role specification for the goal and environment

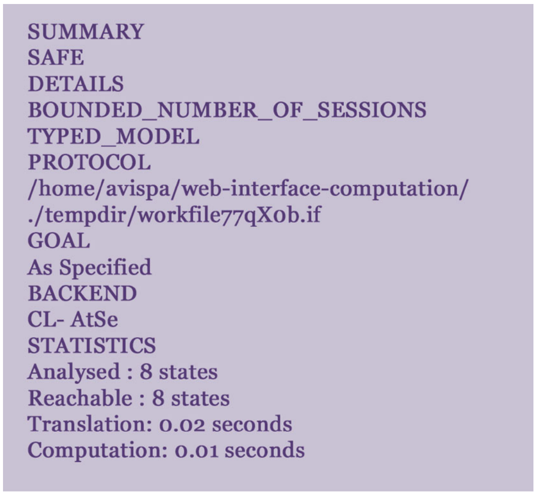
7.2 Simulation results

Two widely used back-ends OFMC and CL-AtSe are used to execute and test the protocol. To check for replay attack, both the analysers test whether the legal agents can execute the specified protocol by performing a search of a passive intruder. The back-ends then provide the intruder with knowledge of some normal sessions among the legitimate agents. For the Dolev–Yao model checking, these back-ends also verify whether there is any possible man-in-the-middle attack by the intruder (attacker). Figures 11 and 12 show the outputs of the two back-ends. The simulation results clearly demon-



```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/avispa/web-interface-computation
./tempdir/workfile77qXob.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.10s
visitedNodes: 18 nodes
depth: 4 plies
```

Fig. 11 Simulation results of the analysis using OFMC



```
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL
PROTOCOL
/home/avispa/web-interface-computation/
./tempdir/workfile77qXob.if
GOAL
As Specified
BACKEND
CL- AtSe
STATISTICS
Analysed : 8 states
Reachable : 8 states
Translation: 0.02 seconds
Computation: 0.01 seconds
```

Fig. 12 Simulation results of the analysis using CL-AtSe

strate that our protocol is safe and is secure against the replay and man-in-the-middle attacks.

8 Performance comparison

Performance analysis and comparison of the proposed protocol with the related protocols have been carried out. The analysis is done on the basis of total computation cost for different phases, and security features. The comparative analysis proves that the protocol is highly efficient as compared to the other protocols.

Table 3 Security comparison

Authors	M1	M2	M3	M4	M5	M6	M7	M8
Watro et al. [2]	✓	✓	×	×	×	×	×	×
Benson et al. [3]	×	×	×	×	×	×	×	×
Wong et al. [4]	×	×	×	✓	✓	✓	×	×
Das et al. [7]	×	×	×	✓	×	×	×	✓
Yuan et al. [13]	×	×	×	×	×	×	×	×
Yeh et al. [16]	×	×	✓	×	×	×	×	×
Chen et al. [17]	✓	×	×	✓	✓	×	✓	×
Yoon et al. [18]	✓	×	×	×	×	×	×	×
Ohood et al. [23]	✓	✓	×	✓	✓	×	×	✓
Wenbo et al. [30]	✓	✓	✓	✓	×	×	×	×
Quan et al. [41]	✓	✓	×	✓	×	✓	✓	×
Proposed protocol	✓	✓	✓	✓	✓	✓	✓	✓

M1 mutual authentication, *M2* key agreement, *M3* password change, *M4* data integrity, *M5* data confidentiality, *M6* availability, *M7* forward security, *M8* scalability

Table 3 summarizes security analysis of the proposed protocol with similar protocols in terms of several security requirements such as mutual authentication, user anonymity, forward secrecy, data confidentiality, etc. from the comparison, it is clear that the proposed protocol adheres to all the security requirements, and provide mutual authentication which most of the protocols failed to satisfy. Also, the proposed protocol offers scalability since it is implemented on cloud. Also, because the protocol is based on ECC, solving ECDLP is computationally infeasible, this allows for maintaining integrity and confidentiality.

Table 4 represents the comparison of the proposed authentication protocol in terms of resistance against the attacks discussed in Sect. 3.1. The comparison shows that the proposed protocol is defensive to all the attacks as compared to the other related schemes. Since most of the protocols are password based they are unable to protect against stolen verifier attacks. The proposed protocol being biometric based prevents from such attacks since biometrics are hard to forge.

For comparing computational cost performance of proposed protocol for the different phases, the notations for time complexity used are T_{H}: time to compute hash, T_{EXP} : time to compute modular exponential, T_{PM}: Time to compute elliptic curve point multiplication, T_{Pair}: pairing computation cost, T_{RC}: RC5 computation cost, T_{PA}: time to compute elliptic curve point addition, T_{PU}: time to compute public key, T_{PR}: time to compute private key T_{AES}: time to compute asymmetric encryption, T_{E}: time to compute elliptic curve polynomial. It is also considered that ($T_{\mathrm{PU}} \gg T_{\mathrm{H}}$ and $T_{\mathrm{PR}} \gg T_{\mathrm{H}}$).

The computational cost comparison of the protocol with the other similar protocols is shown in Table 5. The proposed protocol computes a total of seven hash operations. Hash operations are very lightweight as compared to the exponential or point multiplication, which makes registration a lost cost operation. For the login and authentication phases, our protocol uses one exponential and seven hash operations. On

Table 4 Performance comparison on the basis of attack resistance

Authors	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Watro et al. [2]	×	✓	✓	✓	×	×	✓	✓	×	✓
Benson et al. [3]	×	×	✓	✓	×	✓	✓	✓	✓	×
Wong et al. [4]	×	×	✓	✓	✓	×	×	✓	✓	✓
Das et al. [7]	×	×	×	✓	✓	×	×	✓	✓	×
Yuan et al. [13]	✓	×	×	✓	×	×	✓	✓	✓	×
Yeh et al. [16]	×	×	×	×	×	×	✓	×	✓	×
Chen et al. [17]	×	×	×	✓	✓	×	✓	✓	✓	✓
Yoon et al. [18]	✓	×	×	✓	✓	✓	✓	✓	✓	✓
Ohood et al. [23]	✓	✓	×	✓	✓	✓	✓	✓	✓	✓
Wenbo et al. [30]	×	×	✓	×	×	✓	×	✓	✓	×
Quan et al. [41]	✓	✓	✓	✓	×	✓	✓	✓	✓	✓
Proposed protocol	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

T1 denial attack, *T2* denial of service attack, *T3* compromise attack, *T4* replay attack, *T5* impersonation attack, *T6* insider attack, *T7* forgery attack, *T8* stolen verifier attack, *T9* guessing attack, *T10* man-in-the middle

Table 5 Computational cost analysis

Authors	Registration phase	Login and authentication phase
Watro et al. [2]	$3T_{\text{H}}$	$2\ T_{\text{H}} + 2T_{\text{PR}} + 2T_{\text{PU}}$
Benson et al. [3]	$1T_{\text{EXP}} + 4T_{\text{H}}$	$2n\ T_{\text{H}} + 3nT_{\text{EXP}}$
Wong et al. [4]	$2T_{\text{H}}$	$4\ T_{\text{H+}}2T_{\text{PU}}$
Das et al. [7]	$1T_{\text{H}}$	$5\ T_{\text{H}}$
Yuan et al. [13]	$4T_{\text{H}}$	$9T_{\text{H}}$
Yeh et al. [16]	$4T_{\text{H}} + 2T_{\text{PM}}$	$11T_{\text{H}} + 4T_{\text{PA}} + 6T_{\text{PM}} + 2T_{\text{ECC}}$
Chen et al. [17]	$5T_{\text{H}}$	$7T_{\text{H}}$
Yoon et al. [18]	$3T_{\text{H}}$	$10T_{\text{H}}$
Ohood et al. [23]	$2T_{\text{H}}$	$4\ T_{\text{RC}} + 8T_{\text{H}}$
Wenbo et al. [30]	$3T_{\text{H}} + 1T_{\text{PM}}$	$15\ T_{\text{H}} + 6T_{\text{PM}}$
Quan et al. [41]	$4T_{\text{H}} + 4T_{\text{PM}} + 3T_{\text{AES}}$	$14\ T_{\text{H}} + 6T_{\text{Pair}} + 8T_{\text{AES}}$
Proposed protocol	$7T_{\text{H}}$	$1T_{\text{EXP}} + 7T_{\text{H}}$

comparing with other related schemes both the phases are low cost.

9 Conclusion

In this paper, we have proposed ECC-based authentication protocol real-time remote patient monitoring and tracking based on cloud-IoT environments. In the protocol, an MP can access in real time any remote patient's sensor data stored on a remote cloud server and based on the observed data he can take care of the registered patient. The proposed scheme satisfies all the desirable security requirements including mutual authentication, confidentiality, forward secrecy, scalability and integrity. We have simulated the scheme for formal security verification using the widely accepted web-based AVISPA tool and show that the proposed protocol is secure against passive and active attacks including the replay and man-in-the-middle attacks. In addition, the protocol maintains session key freshness at any time every time the cloud server is accessed by the MP. Also, the protocol sets up a symmetric secret session key between MP and cloud server for use in secure data access and communication. The performance analysis confirms that the proposed protocol is efficient as compared to other existing schemes in terms of computation costs, security requirements and resistance to several attacks.

References

1. Abdmeziem MR, Tandjaoui D (2015) An end-to-end secure key management protocol for e-health applications. Comput Electr Eng 44:184–197
2. Watro R, Kong D, Cuti S, Gardiner C, Lynn C, Kruus P (2004) TinyPK: securing sensor networks with public key technology. In:

Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks, ACM, pp 59–64

3. Benenson Z, Gedicke N, Raivio O (2005) Realizing robust user authentication in sensor networks. Real-World Wirel Sens Netw 14:52
4. Wong KHM, Zheng Y, Cao J, Wang S (2006) A dynamic user authentication scheme for wireless sensor networks. In: IEEE international conference on sensor networks, ubiquitous, and trustworthy computing (SUTC'06), IEEE, vol 1, pp 8
5. Tseng H-R, Jan R-H, Yang W (2007) An improved dynamic user authentication scheme for wireless sensor networks. In: IEEE GLOBECOM 2007-IEEE global telecommunications conference, pp 986–990
6. Hu F, Jiang M, Wagner M, Dong D-C (2007) Privacy-preserving telecardiology sensor networks: toward a low-cost portable wireless hardware/software codesign. IEEE Trans Inf Technol Biomed 11(6):619–627
7. Das ML (2009) Two-factor user authentication in wireless sensor networks. IEEE Trans Wirel Commun 8(3):1086–1090
8. Huang Y-M, Hsieh M-Y, Chao H-C, Hung S-H, Park JH (2009) Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks. IEEE J Select Areas Commun 27(4):400–411
9. Malasri K, Wang L (2009) Design and implementation of a securewireless mote-based medical sensor network. Sensors 9(8):6273–6297
10. Sriram JC, Shin M, Choudhury T, Kotz D (2009) Activity-aware ECG-based patient authentication for remote health monitoring. In: Proceedings of the 2009 international conference on multimodal interfaces, pp 297–304
11. Sarier ND (2010) Improving the accuracy and storage cost in biometric remote authentication schemes. J Netw Comput Appl 33(3):268–274
12. Venkatasubramanian KK, Banerjee A, Gupta SKS (2010) PSKA: usable and secure key agreement scheme for body area networks. IEEE Trans Inf Technol Biomed 14(1):60–68
13. Yuan J, Jiang C, Jiang Z (2010) A biometric-based user authentication for wireless sensor networks. Wuhan Univ J Nat Sci 15(3):272–276
14. Chen T-H, Chen Y-C, Shih W-K, Wei H-W (2011) An efficient anonymous authentication protocol for mobile pay-TV. J Netw Comput Appl 34(4):1131–1137
15. Le XH, Khalid M, Sankar R, Lee S (2011) An efficient mutual authentication and access control scheme for wireless sensor networks in healthcare. J Netw 6(3):355–364
16. Yeh H-L, Chen T-H, Liu P-C, Kim T-H, Wei H-W (2011) A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 11(5):4767–4779
17. Chen H, Ge L, Xie L (2015) A user authentication scheme based on elliptic curves cryptography for wireless ad hoc networks. Sensors 15(7):17057–17075
18. Yoon E-J, Yoo K-Y (2011) A new biometric-based user authentication scheme without using password for wireless sensor networks. In: 2011 20th IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises (WETICE), pp 279–284
19. Drira W, Renault E, Zeghlache D (2012) A hybrid authentication and key establishment scheme for WBAN. In: 2012 IEEE 11th international conference on trust, security and privacy in computing and communications, pp 78–83
20. He D, Chen C, Chan S, Bu J, Vasilakos AV (2012) ReTrust: attack-resistant and lightweight trust management for medical sensor networks. IEEE Trans Inf Technol Biomed 16(4):623–632
21. Kumar P, Ylianttila M, Gurtov A, Lee S-G, Lee H-J (2014) An efficient and adaptive mutual authentication framework for heterogeneous wireless sensor network-based applications. Sensors 14(2):2732–2755
22. Zhang Z, Wang H, Vasilakos AV, Fang H (2012) ECG-cryptography and authentication in body area networks. IEEE Trans Inf Technol Biomed 16(6):1070–1078
23. Althobaiti O, Al-Rodhaan M, Al-Dhelaan A (2013) An efficient biometric authentication protocol for wireless sensor networks. Int J Distrib Sens Netw 9(5):407971
24. Barua M, Lu R, Shen X (2013) SPS: secure personal health information sharing with patient-centric access control in cloud computing. In: 2013 IEEE global communications conference (GLOBECOM), pp 647–652
25. Divi K, Liu H (2013) Modeling of WBAN and cloud integration for secure and reliable healthcare. In: Proceedings of the 8th international conference on body area networks, pp 128–131
26. Li M, Yu S, Guttman JD, Lou W, Ren K (2013) Secure ad hoc trust initialization and key management in wireless body area networks. ACM Trans Sens Netw 9(2):18
27. Lv C, Ma M, Li H, Ma J, Zhang Y (2013) An novel three-party authenticated key exchange protocol using one-time key. J Netw Comput Appl 36(1):498–503
28. Shi L, Li M, Yu S, Yuan J (2013) BANA: body area network authentication exploiting channel characteristics. IEEE J Select Areas Commun 31(9):1803–1816
29. Xue K, Ma C, Hong P, Ding R (2013) A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. J Netw Comput Appl 36(1):316–323
30. Shi W, Gong P (2013) A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. Int J Distrib Sens Netw 9(4):730831
31. Almashaqbeh G, Hayajneh T, Vasilakos AV, Mohd BJ (2014) QoS-aware health monitoring system using cloud-based WBANs. J Med Syst 38(10):1–20
32. Han ND, Han L, Tuan DM, In HP, Jo M (2014) A scheme for data confidentiality in cloud-assisted wireless body area networks. Inf Sci 284:157–166
33. Mishra D, Srinivas J, Mukhopadhyay S (2014) A secure and efficient chaotic map-based authenticated key agreement scheme for telecare medicine information systems. J Med Syst 38(10):1–10
34. Tan Z (2014) A user anonymity preserving three-factor authentication scheme for telecare medicine information systems. J Med Syst 38(3):1–9
35. Thilakanathan D, Chen S, Nepal S, Calvo R, Alem L (2014) A platform for secure monitoring and sharing of generic health data in the Cloud. Futur Gener Comput Syst 35:102–113
36. Xu J, Zhu W-T, Feng D-G (2009) An improved smart card based password authentication scheme with provable security. Comput Stand Interfaces 31(4):723–728
37. Zhao Z (2014) An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem. J Med Syst 38(2):1–7
38. Ullah S, Imran M, Alnuem M (2014) A hybrid and secure priority-guaranteed MAC protocol for wireless body area network. Int J Distrib Sens Netw 10(2):481761
39. Yang H, Kim H, Mtonga K (2015) An efficient privacy-preserving authentication scheme with adaptive key evolution in remote health monitoring system. Peer-to-Peer Netw Appl 8(6):1059–1069
40. Shankar SK, Tomar AS, Tak GK (2015) Secure medical data transmission by using ECC with mutual authentication in WSNs. Procedia Comput Sci 70:455–461
41. Quan Z, Chunming T, Xianghan Z, Chunming R (2015) A secure user authentication protocol for sensor network in data capturing. J Cloud Comput 4(1):1–12
42. Lu Y, Li L, Peng H, Yang Y (2015) An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. J Med Syst 39(3):1–8

43. Arshad H, Nikooghadam M (2014) Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. J Med Syst 38(12):1–12
44. Hossain MS, Muhammad G (2015) Cloud-assisted speech and face recognition framework for health monitoring. Mobile Netw Appl 20(3):391–399
45. Amin R, Biswas GP (2015) A secure three-factor user authentication and key agreement protocol for tmis with user anonymity. J Med Syst 39(8):1–19
46. Xu X, Zhu P, Wen Q, Jin Z, Zhang H, He L (2014) A secure and efficient authentication and key agreement scheme based on ecc for telecare medicine information systems. J Med Syst 38(1):1–7
47. Liu C, Chung Y (2017) Secure user authentication scheme for wireless healthcare sensor networks. Comput Electr Eng 59:250–261
48. Moosavi SR et al (2016) End-to-end security scheme for mobility enabled healthcare Internet of Things. Futur Gener Comput Syst 64:108–124
49. Wu F, Xu L, Kumari S, Li X, Das AK, Shen J (2017) A lightweight and anonymous RFID tag authentication protocol with cloud assistance for e-healthcare applications. J Ambient Intell Hum Comput 2017:1–12
50. Dhillon PK, Kalra S (2017) A lightweight biometrics based remote user authentication scheme for IoT services. J Inf Secur Appl 34:255–270
51. Li C-T, Wu T-Y, Chen C-L, Lee C-C, Chen C-M (2017) An efficient user authentication and user anonymity scheme with provably security for IoT-based medical care system. Sensors 17(7):1482
52. Dhillon PK, Kalra S (2017) Secure multi-factor remote user authentication scheme for Internet of Things environments. Int J Commun Syst 30(16)
53. Góodor G, Szendi P, Imre S (2010) Elliptic curve cryptography based authentication protocol for small computational capacity RFID systems. In: Proceedings of the 6th ACM workshop on QoS and security for wireless and mobile networks, pp 98–105
54. Niu X, Jiao Y (2008) An overview of perceptual hashing. Acta Electron Sin 36(7):1405–1411
55. Armando A, Basin D, Cuellar J, Rusinowitch M, Viganò L (2006) AVISPA: automated validation of internet security protocols and applications. ERCIM News 64