J Med Syst (2018) 42: 142
https://doi.org/10.1007/s10916-018-0987-5

MOBILE & WIRELESS HEALTH

An Efficient Mutual Authentication Framework for Healthcare System in Cloud Computing

Vinod Kumar[1] · **Srinivas Jangirala**[2] · **Musheer Ahmad**[1]

Received: 6 March 2018 / Accepted: 1 June 2018 / Published online: 28 June 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

The increasing role of Telecare Medicine Information Systems (TMIS) makes its accessibility for patients to explore medical treatment, accumulate and approach medical data through internet connectivity. Security and privacy preservation is necessary for medical data of the patient in TMIS because of the very perceptive purpose. Recently, Mohit et al.'s proposed a mutual authentication protocol for TMIS in the cloud computing environment. In this work, we reviewed their protocol and found that it is not secure against stolen verifier attack, many logged in patient attack, patient anonymity, impersonation attack, and fails to protect session key. For enhancement of security level, we proposed a new mutual authentication protocol for the similar environment. The presented framework is also more capable in terms of computation cost. In addition, the security evaluation of the protocol protects resilience of all possible security attributes, and we also explored formal security evaluation based on random oracle model. The performance of the proposed protocol is much better in comparison to the existing protocol.

Keywords Cloud computing · TMIS · Mutual authentication · Signature · Medical data

Introduction

With the quick progress of information technology, the use of TMIS is increasing day by day. To offer conducive and rapid network services, a novel kind of cloud computing organization [8, 25] which contains of a large number of processors, memories, high-speed networks, and various appliances is expected by consumers through the internet. Cloud computing services are offered via a browser to access an online data applications. These computing methods can be achieved by the cloud platform. Further, the work [44] explained that the cloud services will develop in the future. Therefore, the security and privacy of the cloud computing have become important issues. Various articles have proposed different issues of their apprehensions, such as: cloud security [9, 54], personal privacy and cloud services [11, 52]. According to the article [12], several operations are associated to cloud services and their uses.

With the fast development of internet appliances, people can select an appropriate hospital for high excellence of healthcare [2, 40]. Furthermore, for the progress of medical center superiority and the medical trade struggles the healthcare center sustains hospitals in remote localities. The medical manufacturing offers more specialized medical apparatus and improves medical maintenance superiority. With the help of medical manufacturing the healthcare centers are trying to improve their services so that patients can get easy access of medical facilities [36]. For an example, if the electronic medical records shared very well, the healthcare centers can share their resources through the internet. Patients need not to depict their inspection reports. On the other side, as we know if the patient has come to healthcare center, the medical employees should try to get the patient's medical reports as early as possible for the

This article is part of the Topical Collection on *Mobile & Wireless Health*

✉ Srinivas Jangirala
jangiralasrinivas@maths.iitkgp.ernet.in;
getsrinunow1@gmail.com

Vinod Kumar
vinod.iitkgp13@gmail.com

Musheer Ahmad
mahmad@jmi.ac.in

[1] Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi, 110025, India

[2] Department of Mathematics, Indian Institute of Technology Kharagpur, Khargpur, 721302, India

preparation of medical treatment and to decrease errors. Moreover, the sensor planted in the patient's body is another option for the healthcare center to get his/her medical report.

In medical organization, the cloud users store medical data in the cloud database to recapture the data securely. As it is common that cloud is not completely secured, so a protected and authenticated framework required to prevent simple security attacks [33]. In newly years, there are several authentication schemes [3–5, 13, 18, 45] proposed for TMIS, where the patients find their treatment online. As indicated in [14] TMIS proficiency medical doctor and patients to begin a conversation via public channels to support healthcare assistances precisely in the patient's residence. As attribute of TMIS, both doctors and patients can perform together through the cloud server, i.e. a patient transfer his/her manifestations to the TMIS server and the doctor collect them and uploads diagnosis data report of the patient to the cloud server as if they are collaborating precisely, and it is happening via TMIS. Furthermore, the transmission is done via public channel, so it is important to know how to get extra benefits from medical resources with secure communication. Additionally, the security obligations, data confidentiality, patient anonymity, and patient authentication are the significant appearances to retain throughout the communication. In order to keep up patient anonymity [19, 28, 41], the identification of the patient need to differentiate from the others including eavesdropper. In TMIS, the patient's medical reports are extremely significant, and they have not to revealed widely. As the message shared between patients/doctors and cloud are very serious information and so, data are gathered strongly. As medical information comes under imperative data and collapse of it may reason deterioration of ones life [50], thus it is essential to prove a protected scheme so that no attacker can attempt to find patient's data and mistreatment it. Newly, there have been several protocols proposed to recognize anonymity concern. Mainly of these existing schemes are not relevant to offer patient anonymity in the healthcare system.

Related works

Smart card based authentication technique is the ordinary which adopted to avert unapproved access over the confident networks. There are various authentication scheme [35, 36, 42] obtainable using card [30], where the clients accept a password and imports a smart card with it. The authentication scheme is very favorable in different use, such as wireless sensor network, medical system and adhoc networks[6, 7, 20–28, 34, 40, 52–54]. Wu et al. first suggested a password-based user authentication protocol [47] and a reliable client authentication and key agreement protocol for network based hospital-acquired epidemic surveillance information system [49] then, Wu-Lee et al. [48] presented a secure authentication scheme for TMIS. Then, He et al. [18] accumulated that Wu et al.'s protocol[48] has different technical issues, like as an insider and impersonation attack, they also advised an improved scheme. In 2012, Wei et al. [46] observed that earlier schemes [18, 48] which are not secured across security flaws and recommended an appreciated protocol to prevent the occurring attacks. After that, Zhu [52] proved that Wei et al. [46] protocol is not protected against off-line password guessing attack and implemented a protected authentication protocol for TMIS, which based on the RSA cryptosystem. In 2013, Jiang et al.' [29] proposed privacy enhanced authentication scheme for TMIS. Kumari et al. [31] proposed cryptanalysis and improvement of a privacy enhanced scheme TMIS which claimed that [29] fails to offer online password guessing attack, impersonation attack, and stolen-verifier attack. Nonetheless, Mishra et al. [38] presented a secure and capable chaotic map-based authenticated key agreement protocol for TMIS in which they examined that the scheme [29] does not resist denial-of-service attack. In current year , Liu et al.'s [55] proposed authentication based a practical privacy preserving data aggregation scheme which is efficient in communication security aspects.

In 2013, Tan [43] suggested a capable biometrics based authentication scheme for TMIS which is a smart card based password authentication and key agreement protocol by implementing a biometric system, and the protocol is more secure. Further, Yan et al. [50] proposed a secure biometric-based authentication protocol for TMIS which validated that the scheme [51] not passes to resist Denial-of-Service attack. In 2014, Mishra et al. [37] presented cryptanalysis and improvement of Yan et al. Biometric-based authentication method for TMIS which described that scheme [50] have a number of security outlet, like as the client privacy, ineffectual password, insufficient login phase, password guessing attack, biometric update phase and three-factor authentication difficulty. To decide the above recognized complication, they as well presented an enhanced protocol. Li et al. [33] presented a secure chaotic maps, and smart card based password authentication and key agreement scheme with user anonymity for TMIS and declared that the Lee et al.'s [32] chaotic –maps based client authentication protocol bear security weaknesses like absence of client identifier in authentication phase, service misuse attacks, and advised a more effective explanation for accessing TMIS. In 2014 Chen et al. [16] associates the cloud computing environment with mobile devices to give medical resources and uses cryptographic infrastructure to defend the patients secret information. Then, the scheme has several security flaws. Chen et al. [15] also proposed a new scheme for the same environment based on the cloud computing environment, although the scheme does not support message authentication and patient anonymity. To improvise the security flaws in [15], Chiou et al. [17] adapted the occurring protocol and believed that the framework prepares real TMIS, message authentication and patient anonymity. In 2016, Liu et al.'s [56] proposed a privacy-preserving health data aggregation scheme. In 2017, Liu et al. [57] presented a lightweight pseudonym authentication scheme for multi-medical server architecture for TMIS. Furthermore, Mohit et al. [39] proposed mutual authentication framework for cloud environment based healthcare system, we found that it is vulnerable to stolen verifier attack, many logged-in patient attack, patient anonymity, impersonation attack and fails to protect session key.

Motivation and contribution

Recently, Mohit et al. [39] suggested a mutual authentication protocol for TMIS that can work in the cloud computing environment.

- It is analyzed and shown as follows:
  - Their scheme does not secure against stolen-verifier attack.
  - Their protocol does not support many logged in patient attack.
  - Their protocol does not ensure the anonymity of the patient.
  - Their protocol does not secure against impersonation attack.
  - Their protocol fails to protect session key.
- In this regard, to attain security against the aforementioned attacks and to ensure the security of an entire package, a mutual authentication framework for TMIS is presented which is suitable for the cloud computing. The proposed framework has many significant characteristics, such as:
  - Mutual authentication is accomplished between healthcare center and cloud server, patients and cloud server, doctor and cloud server, and patient and healthcare centers to strengthen the safety of a structure and transforming information.
  - Furthermore, the proposed protocol is strong against many security attributes, i.e., implements security against, patient anonymity, man-in-the-middle attack, strong replay attack, known key security property, data confidentiality, data non-repudiation, message authentication, impersonation attack, session key security, stolen mobile device attack, off-line password/identity guessing attack and many logged-in patient's attack.
  - We provided formal security analysis of our proposed protocol based on random oracle model.
  - We evaluate the proposed scheme with other existing works and found that our scheme gets minimum computational and communication expenditure, but ensures security of the system.

Road map of the paper

The rest of this paper is formulated as follows. In "Preliminaries", we describe the Preliminaries. Section "Review of Mohit et al.'s scheme", We reviewed Mohit et al.'s scheme. Section "Cryptanalysis of Mohit et al.'s scheme", The cryptanalysis of Mohit et al.'s Scheme. Section "Security model", Security model, Section "The proposed protocol", We proposed an improved mutual authentication protocol for healthcare system in cloud computing. Section "Security proof", formal security analysis of the proposed protocol. Section "Performance analysis", performance analysis of the proposed and earlier existing schemes. Finally, Section "Conclusion", discusses about the conclusion. Moreover, we make use of the notation/symbol throughout the paper as given in Table 1.

Preliminaries

Elliptic curve cryptography

Let q be the large prime and $\mathcal{E}$ denote an elliptic curve over the prime finite field F_q, an equation of elliptic curve over prime finite field is given by $y^2 = x^3 + ax + b mod q$ with $a, b \in F_q$ and $4a^3 + 27b^2 mod q \neq 0$. So, this is a non singular elliptic curve. Then, the additive elliptic curve group defined as $G = \{(x, y) : x, y \in F_q; (x, y) \in \mathcal{E}\} \bigcup \{\Theta\}$, where the point Θ is known as point at infinity which works as the identity element of G. The scalar multiplication on the group G is defined as $tP = P + P ... + P(t - times)$ and the point addition in G as: If $P = (x_1, y_1)$, $Q = (x_2, y_2) \in G$, then $P + Q = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2 mod q$, $y_3 = (\lambda(x_1 - x_2) - y_1) mod q$ where

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} mod q \quad if\, P \neq Q \\ \\ \frac{3x_1^2 + a}{2y_1} mod q \quad if\, P = Q \end{cases}$$

The more details of elliptic curve group are given in [20].

- **Elliptic Curve Discrete Logarithms Problem (ECDLP)**: For given $P, Q \in G$, find $k \in Z_q^*$ such that $P = kQ$, which is hard.

Table 1 Notations/Symbol used

Symbol	Description
l	The security parameter
$\mathcal{E}(F_q)$	Elliptic curve $\mathcal{E}$ over a prime finite field F_q
q	Large prime
Z_q^*	Additive group of order q
P	Patient
ID_i	Unique identity of entity i
CS	Cloud server
SID	Dynamic pseudo random number
BS	Body sensor
D	Doctor
$h(\cdot)$	Cryptographic one way hash function
HC	Healthcare center
$E_y(m)/D_y(m)$	Encryption/Decryption of message m using key y
MD_i	Message digest of i
$SK_{ij}(.)$	Session key between entities i and j
PU_i	Public key of entity i
$S_k(m)$	Signature of m with using key k
$i =?j$	Whether i equals j
$V_k(m)$	Verified signature of m with using key k
PR_i	Private key of entity i
K_i	The computing key of entity i
$\|$	Concatenation operation
OTP	One time password
$\oplus$	Bitwise XOR operation
E	Adversary
G	Elliptic curve group under addition
g	Generator of G
Sig_i	The signature of i^{th} participant
sn_i	The sequence number of i^{th} participant

- **Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP)**: For $a, b \in Z_q^*$ and g is the generator of G, for given (g, ag, bg), then to compute abg is hard for the group G.

Hash function

Definition A one-way hash function $H_i : \{0, 1\}^* \rightarrow \{0, 1\}^l$, inputs an arbitrary string length take $x \in \{0, 1\}^*$, and outputs a finite length string l bit message assimilate or hash value $h(x) \in \{0, 1\}^*$. A best hash function should contain the following properties:

- For any given input x, it is accessible to calculate the digest $h(x)$.
- **One-way:** For a given hash value $y = h(x)$, it is computationally not feasible to obtain x.
- **Weak-collision resistance:** For any given input x, obtaining any other input y, with $x \neq y$, such that $h(x) = h(y)$ is computationally infeasible.
- **Strong-collision resistance:** Finding a pair of inputs (x, y) with $x \neq y$, such that $h(x) = h(y)$ is also computationally not feasible.

Assumptions for the mutual authentication protocol

We take some assumptions to evaluation the invoked mutual authentication protocol.

Assumption 1 The hash results, the random number and secret numbers stored in cloud server. They reach the secure length l.

Assumption 2 The E_y (m), D_y (m) and $h(.)$ are capable. That is to tell, in polynomial time, anybody can not decrypt the encrypted string E_y (m) without knowing y and no one find the collision of $h(m)$, where m is the string [58–61].

Assumption 3 According to [2, 3, 60], both the identity and the one time password (OTP) of the entity have low entropy. There are two dictionaries in which one for identities and second for OTP. Advisory E can guess them in polynomial time.

Assumption 4 Adversary E can get the previous session keys, which is from the known-key attacks [60, 61].

Review of Mohit et al.'s scheme

Mohit et al. proposed a standard mutual authentication scheme for cloud based healthcare environment. There are five bodies : Patient, Cloud server, Doctor, Body sensor and Healthcare center. This scheme involves of four phases: (1) Healthcare center upload phase, (2) Patient data upload phase, (3) Treatment phase and (4) Checkup phase. Those are followed as:

Healthcare center upload phase (HUP)

The patient P registers herself/himself in the HC, and the HC provides OTP. The HC operates authentication with CS and uploads the P's inspection medical report to CS as described bellow as:

Step 1. The healthcare center generates inspection record $m_H = (ID_P, Data_P)$, and uses unique identity ID_H of healthcare center with a elected random number R. The HC sends message $\{ID_H, m_H\}$ to the CS via secure channel.

Step 2. On receiving messages, CS takes secure key x and executes $A = h(ID_H \| R \| x)$, $S_1 = h(A)$ and $B = ID_H \oplus x$. Sends $\{S_1, B\}$ to HC via public channel.

Step 3. On collecting message, HC computes $x' = B \oplus ID_H$, $A' = h(ID_H \| x' \| R)$ and checks whether $S_1' = ?h(A')$ holds or not. If it does not hold, HC exits the session. Otherwise HC computes session key $SK_{HC} =$

$h(ID_H \| A' \| B)$, $key_1 = h(ID_P \| OTP)$ and encrypts the record as $C_H = E_{key_1}(m_H)$. Further, the HC computes $MD_H = h(m_H)$, digital signature $Sig_H = S_{PR_H}(MD_H)$, encrypts $C_1 = E_{SK_{HC}}(ID_P, C_H, Sig_H, SID)$ and computes $S_2 = h(SK_{HC} \| C_1)$. Finally sends message $\{S_2, C_1\}$ to the CS via public channel.

Step 4. Upon gathering message, the CS computes session key $SK'_{HC} = h(ID_H \| A \| B)$ and checks whether $S'_2 \stackrel{?}{=} h(SK'_{HC} \| C_1)$ hold or not. If, it does, CS authenticates HC and decrypts the message using session key SK'_{HC} to get $(ID_P, C_H, Sig_H, SID) = D_{SK_{HC}}(C_1)$, and store ID_P, C_H, Sig_H, SID. Otherwise, it fails and goes to Step.

Patient data upload phase (PUP)

The BS is fixed in the P's body. The P requests BS, to assemble the reorganized health information, and presented it to the P through secure mobile device. The patient inputs identity ID_P and OTP of his/her mobile device. The cloud server provides a slot sequence number sn_i, inspection record card m_H to the patient which discussed below as:

Step 1. P obtains health information message $m_B = (ID_P, Data_B)$ from BS through mobile device. Then, P inputs his/her ID_P, SID and forwards message $\{ID_P, SID\}$ to the CS through a secure channel.

Step 2. On collecting messages, CS computes $I = sn_i \oplus SID$, $S_3 = h(SID \| I \| C_H \| Sig_H)$ and sends $\{I, S_3, C_H, Sig_H\}$ to P via open channel.

Step 3. On getting information, P computes $sn'_i = I \oplus SID$ and checks whether $S'_3 =?h(SID \| I \| C_H \| Sig_H)$ grips or not. If is does, P authenticates CS and calculates session key $SK_{PC} = h(ID_P \| SID)$, $key_1 = h(ID_P \| OTP)$. Then, P decrypts the ciphertext to find $m_H = D_{key_1}(C_H)$ and computes $MD_H = V_{PU_H}(Sig_H)$. After that, checks $m_H =?h(MD_H)$ holds or not. If is does, computes $key_{PD} = h(ID_P \| ID_D \| sn_i)$, encrypts $E_{key_{PD}}(m_H, m_B)$, computes $MD_P = h(m_B)$, generates signature $Sig_P = S_{MD_P}(MD_P)$ and computes $S_4 = h(SK_{PC} \| C_P \| Sig_P)$. Sends message $\{S_4, C_P, Sig_P\}$ to CS over public channel.

Step 4. On accepting messages, CS executes $SK'_{PC} = h(ID_P \| SID)$ and checks whether $S'_4 =?h(SK'_{PC} \| C_P \| Sig_P)$ holds or not. If is does, cloud store C_P, Sig_P. Otherwise, terminates the session.

Treatment phase (TP)

In this phase, doctor provides treatment of authenticated patient by acting authentication between the doctor and the cloud server. Cloud contains all the medical report of patients and sends to doctor. Doctor and cloud server perform as bellow:

Step 1. Doctor D sends his/her identity ID_D and random number RD to CS through secure public channel.

Step 2. On receiving message, CS sends identity ID_D of the P and sequence number sn_i to D via secure public channel. Then, CS computes $S_5 = h(RD \| Sig_P \| sn_i)$ and sends message $\{S_5, Sig_P, C_P\}$ to D through public channel.

Step 3. Upon receiving message, doctor verifies whether $S'_5 =?h(RD \| Sig_P \| P_P)$ holds or not. If it does, D authenticates the CS and computes session key $SK_{DC} = h(ID_P \| RD \| sn_i)$, else rejects the message. Moreover, D computes $key_{PD} = h(ID_P \| ID_D \| sn_i)$, and decrypts the received message as $(m_H, m_B) = D_{key_{PD}}(C_P)$, and verifies the patient's signature using public key of P, which is $MD_P = V_{PU_P}(Sig_P)$ and checks whether $MD_P =?h(m_B)$ hold or not. If it does, D generates medical report $m_D = (ID_P, Data_D)$, encrypts ciphertext $C_D = E_{key_{PD}}(m_H, m_B, m_D)$ and computes $MD_D = h(m_D)$, D signature $Sig_D = S_{PR_D}(MD_D)$, and $S_6 = h(SK_{DC} \| C_D \| Sig_D)$ and sends message $\{S_6, C_D, Sig_D\}$ to CS through public channel.

Step 4. On getting messages, CS computes $SK'_{DC} = h(ID_P \| RD \| sn_i)$ and check whether $S'_6 =?h\ (SK'_{DC} \| C_D \| Sig_D)$ holds or not. If it does, CS store C_D, Sig_D. Otherwise, terminates the session and goes to Step 1.

Check up phase (CP)

In this phase, the P authenticates CS to encrypted medical report of the patient. The detail of the narration of this section is as follows:

Step 1. The patient inputs identity ID_P, request and sends message $\{ID_P, Request\}$ to CS via secure public channel.

Step 2. On collecting message, CS executes $S_8 = h(ID_P \| ID_D \| Sig_D)$ and sends message $\{S_8, C_8, Sig_D\}$ to P via open channel.

Step 3. Upon getting information, P checks whether $S'_8 =?h(ID_P \| ID_D \| Sig_D)$ holds or not. If it does not hold, exits the session. Otherwise, the P decrypts the ciphertext with using key_{PD} to get $(m_H, m_B, m_D) = D_{key_{PD}}(C_D)$ and verifies the signature $Sig_D = V_{PU_D}(Sig_D)$ and checks whether $MD_D =?h(m_D)$ hold or not. If it does, P encrypts message $C_2 = E_{key_P}(m_H, m_B, m_D)$, computes $S_9 = h(SID \| C_2)$ and sends message $\{S_9, C_2\}$ to the CS through public channel.

Step 4. On receiving message, CS checks whether $S'_9 = ?h(SID \| C_2)$ holds or not. If it does, CS store C_2, otherwise terminates the session and goes to Step 1.

Cryptanalysis of Mohit et al.'s scheme

After reviewed the Mohit et al.'s scheme, we found five security weaknesses in the protocol. We have discussed below as:

Stolen-verifier attack

The stolen-verifier attack, means that an adversary stoles the password or identity-verifier from the CS database and applies an off-line guessing attack on it to get patient's correct OTP or identity ID_P. In Mohit et al.'s scheme, E stolen patient's mobile phone, and intercepts in PUP. There are two following cases possible:

Stolen-verifier password attack

If an adversary E retrieves the store parameter $key_1 = h(ID_P\|OTP)$, then he/she can successfully perform password guessing attack:

Step 1. An adversary E intercept in PUP, and retrieves ID_P.
Step 2. E guesses one time password OTP^* in one time password dictionary $|OTP|$ and computes $key_1 = h(ID_P\|OTP^*)$, verifies $h(ID_P\|OTP) = ?h(ID_P\|OTP^*)$.
Step 3. If the verification succeed, E consider OTP^* as a patients's one time password. Otherwise step 2 is repeated.

The illustration of the attack is shown in Fig. 1.

Stolen-verifier identity attack

If E retrieves the store parameter $key_1 = h(ID_P\|OTP)$, then he/she can successfully perform identity guessing attack:

Step 1. E intercept in PUP, and retrieves patient's OTP.
Step 2. E guesses an identity ID_E in identity dictionary $|ID|$ and executes $key_1 = h(ID_E\|OTP)$ and verifies $h(ID_E\|OTP) =?h(ID_P\|OTP)$.
Step 3. If the verification succeed, consider ID_E as the patient identity, Otherwise Step 2 is repeated.

The illustration of the attack is shown in Fig. 2.

Many logged-in patient attack

The many logged-in patient attack is defined as the simultaneous access of a legitimate patient's account of a CS by multiple adversaries using the same identity of the P. In Mohit et al.'s scheme, CS store the identity and OTP of the P in the database. But in this attack, we discuss only patient identities in PUP. Assume that legitimate identity ID_P is accountably exposed to many adversaries $E_1, E_2, E_3, ...E_j......E_m$, all knows ID_P and SID, then performed to CS at the same time by executing following steps:

Step 1. Each E_j sends the message $\{ID_P, SID\}$ to CS.
Step 2. The CS computes $I_1 = sn_{i1} \oplus SID$, $I_2 = sn_{i2} \oplus SID$, $I_3 = sn_{i3} \oplus SID$, ...$I_j = sn_{ij} \oplus SID$, ...$I_m = sn_{im} \oplus SID$ and $s_3^1 = h(SID\|I_1\|C_H\|Sig_H)$, $s_3^2 =$

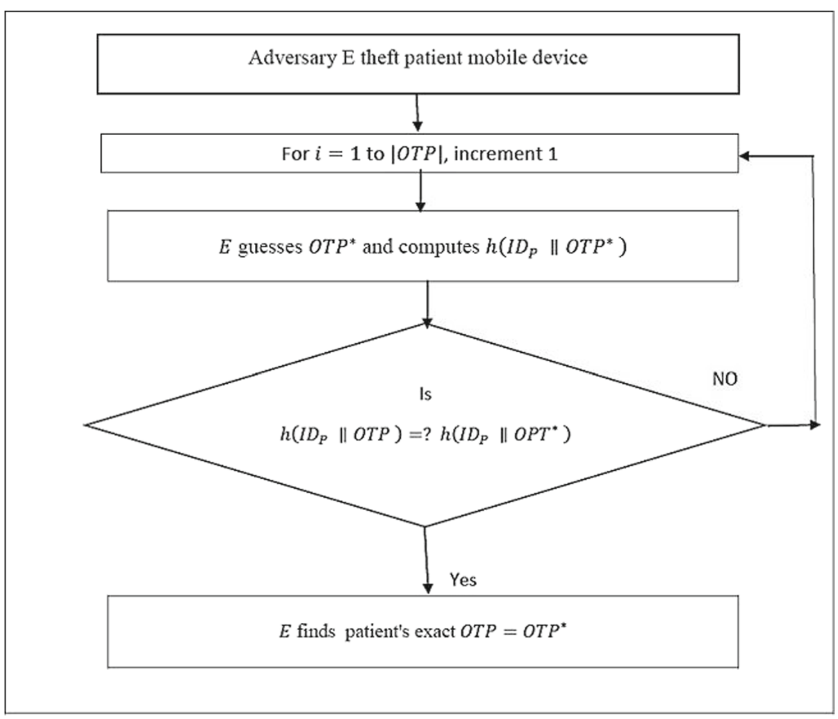


Fig. 1 Stolen-verifier password attack

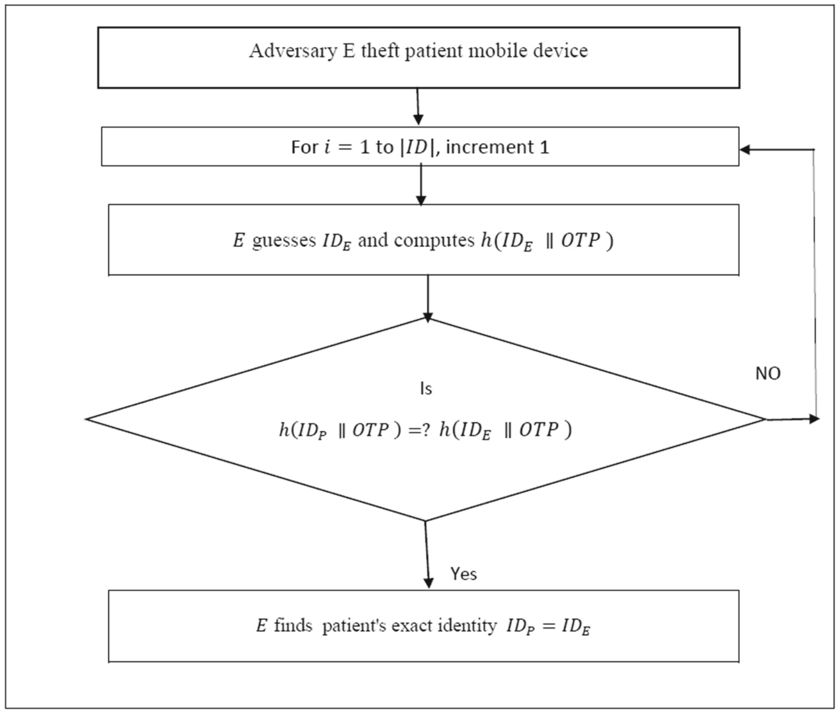


Fig. 2 Stolen-verifier identity attack

$h(SID\|I_2\|C_H\|Sig_H), s_3^3 = h(SID\|I_3\|C_H\|Sig_H)$.... $s_3^j = h(SID\|I_j\| C_H\|Sig_H)$.....$s_3^m = h(SID\|I_m\| C_H\|Sig_H)$. Thus, CS allows all $E_1, E_2, E_3, ...E_j$ ......E_m to communicates in concurrently (Fig. 3).

Patient anonymity

In Mohit et al.'s protocol, patient has the same identity in PUP, TP and CP. There was no anonymous identity use in these phases. These offer a chance for the attacker to track patient's activity over public network.

Impersonation attack

In HUP of Mohit et al.'s protocol, CS store parameters ID_P, C_H, Sig_H, SID in database and sn_i is public. If E intercepts in PUP and perform as:

Step 1. E computes $I_E = sn_i \oplus SID, S_{3E} = h(SID\| I_E\|C_H\|Sig_H)$ and sends $\{I_E, S_{3E}, C_H, Sig_H\}$ to P.

Step 2. On receiving message, P computes $sn_i' = I_E \oplus SID, S_3' = h(SID\|I_E\|C_H\| Sig_H)$ and verifies that $S_{3E} = S_3'$. Further, the P computes session key $SK_{PC} = h(ID_P\|SID), key_1 = h(ID_P\|OTP), m_H = D_{key_1}(C_H), MD_H = V_{PU_H}(Sig_H)$, where $m_H = h(MD_H)$ and computes $key_{PD} = h(ID_P\|ID_D\|sn_i)$, encrypts $C_P = E_{key_{PD}}(m_H, m_B)$, computes $MD_P = h(m_B)$, signature $Sig_P = S_{PR_P}(MD_P), S_4 = h(SK_{PC}\|C_P\|Sig_P)$ and sends $\{S_4, C_P, Sig_P\}$ to E.

Step 3. On receiving message, E computes $SK_{PC}^E = h(ID_P\|SID)$ and $S_4^E = h(SK_{PC}^E\|C_P\|Sig_P)$.

Here, SK_{PC}^E= SK_{PC} and S_4^E=S_4. Thus, Mohit et al.'s scheme fails to protect the impersonation attack.

Fails to protect the session key

In PUP of Mohit et al.'s protocol. Then, P computes session key $SK_{PC} = h(ID_P\| SID)$. From impersonation attack session 4.4, adversary E computes session key $SK_{PC}^E = h(ID_P\|SID)$. Thus E successfully computes the session key of the patient. Similarly, E got session key in HUP and TP. Hence, Mohit et al.'s scheme fails to support of session key.

Security model

In this section, we discuss the security model on the proposed scheme which is based on [1, 10, 53, 61]. There are two entities U and V, or every partner I with no difference in the proposed protocol $\mathcal{P}$. U has an identity ID_U and a password PW_U. V has an identity ID_V and a password PW_V. All passwords are in a dictionary with size $\mathcal{N}$, and elliptic curve group G has a generator g of order q.

Every party has several occurrence. Let U^S be the S^{th} occurrence of U. Similarly, V^S and I^S can be prescribed. E case is an oracle. We apply a simulator to provides the replay to input information. In this way, there are three cases for an oracle: accept, reject and $\perp$. If an oracle finds a ordinary information, the obtain state is achieved. If an incorrect information is collected, the reject case is arrived. Otherwise, if no response is generated, $\perp$ occurs. Once upon

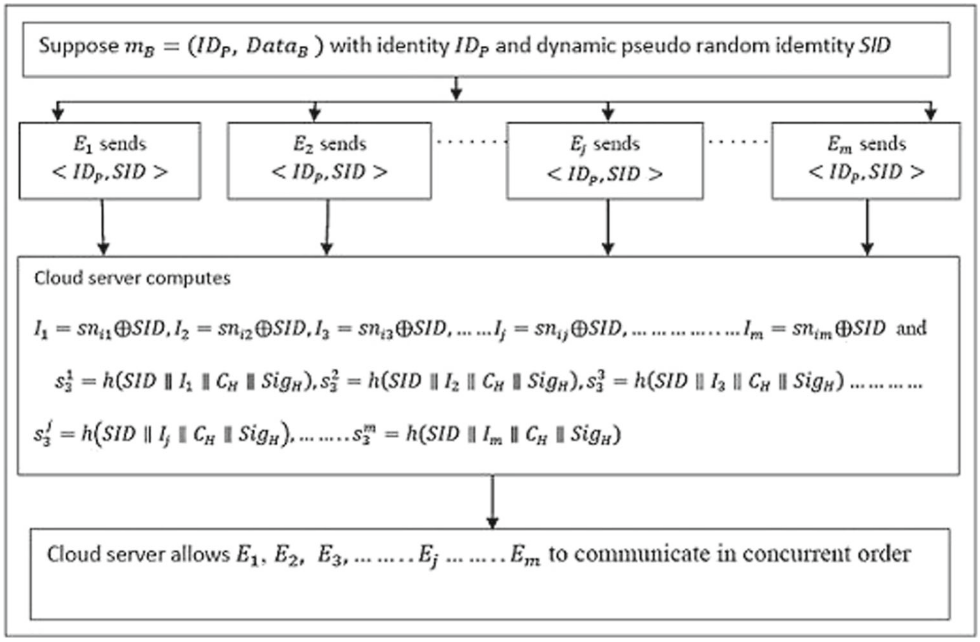


Fig. 3 Many logged-in patient attack in Mohit et al.'s scheme

a time the oracle U^i or V^j is established and determines a session key, each of them has the subsequent elements: a session identity (sid_{U^i}) or (sid_{V^j}), a partner identity (pid_{U^i}) or (pid_{V^j}), and a session key (SK_{U^i}) or (SK_{V^j}). E can totally run the simulator and query oracles to destroy the security of authentication or the session keys. We list all the oracles as followings:

Execute (U^i, V^j): This query simulates the passive attack, and permits the attacker E to learn all the transmitted communication between the instances of entities U^i and V^j.

Send (I, I_r^j, M): This query simulates the active attack and It makes that the body I forwards a message M to the occurance I_r^j. If M is exact message and I_r^j is prepared to accept the information, the simulator will return the message which I_r^j should develop. Otherwise, if M is wrong, the query is aborted.

Reveal (I^k): It expresses known-key attacks and for U and V. If I^k grasps the status of partnering, the adversary E can obtain the session key through asking this query.

Corrupt (I^k): This query is use to check the perfect forward security property of the session key on the oracle I^k. All the messages of I^k is obtained by adversary E after this query, since E has known some message in the system, we list the specific as follow:

- ***Corrupt*** (U^i): It allows the adversary E to concession the long-term private key of the session key of U^i.
- ***Corrupt*** (V^j): It allows the adversary E to concession the long-term private key of the session key of V^j.

Test (I^k): At last adversary E chooses a session to challenge. At this time I may be U or V. If I^k has not been approved or it is not able for the view $sfs - fresh$ which will disclosed below, the simulator will go back $\perp$. Otherwise a coin s is toss. The simulator will output the actual session key if $s = 1$ appear. If $s = 0$ appears, a random string say session key is returned to adversary E.

We use few definitions for the verification of proof as follows:

Partnering: As the session key is created between U^i and V^j, we call U^i and V^j are partners if and only if they are established and $sid_{U^i} = sid_{V^j}$, $pid_{U^i} = V^j$, $Pid_{V^j} = U^j$ and $SK_{U^i} = SK_{V^j}$.

pfs-fresh (fresh with perfect forward security): We use this opinion for only U^i and V^j, we say that I^k is the $pfs - fresh$ if no one the followings queries appears:

- E Reveal(I^k) occurs;
- E Reveal(pid_{I^k}) appears;
- Before Test arises, *Corrupt*(I^k) or *Corrupt*(pid_{I^k}) has been asked.

pfs- ake security: we define E's advantage against the protocol $\mathcal{P}$ is the probability that E properly guesses the coin s after Test(I^k) query. Of course, I^k is established and $pfs - fresh$.

The advantage of E is $Adv_{\mathcal{P}}^{pfs-ake}(E) = 2Pro[s = s'] - 1$.

Where E outputs s'. If Q_s is the number of *Send* queries and $Adv_{\mathcal{P}}^{pfs-ake}(E)$ is negligibly longer than $\frac{O(Q_s)}{N}$ with l, the protocol is $pfs - ake$ secure.

To show the protocol, we take two new assumptions for ECC. Those are based on the "Elliptic curve cryptography".

- **Elliptic Curve Decisional Diffie-Hellman problem (ECDDHP)**: Let $ag, bg, cg \in G$, The probability for E to determine whether $cg = abg$ polynomial time t is $Adv_E^{ECDDHP}(t)$ and ϵ is an ignorably small positive real number and in fact $Adv_E^{ECDDHP}(t) \leq \epsilon$.
- **Elliptic Curve Gap Diffie-Hellman problem (ECGDHP)**: Let $ag, bg \in G$, The probability for E to execute abg with an ECDDHP oracle in polynomial time t is $Adv_E^{ECDDHP}(t) \leq \epsilon$.

The proposed protocol

Architecture

There are five components associated in the proposed protocol for conversation are as follows:

(1) Patient: A person, who is applying for medical treatment.

(2) Doctor: A person, who has been skilled in medical science and offer treatment to patients.

(3) Healthcare center: A physical residence where the patient takes treatment.

(4) Cloud server: A server to collect patient's medical data or records.

(5) Body sensor: A device associated with a physical impression of the patient and sends information to the patient's mobile device.

The architecture of this proposed protocol is shown in the Fig. 4, and the details are as follows:

- Firstly P goes to HC for the routine-checkup/inspection and takes registration, where HC support the report of the P.
- HC uploads the medical report/data of P to the CS. B installed in the P's body collects the fitness information of the patient and forward to a P's mobile device securely.
- P upload current medical record by updating the earlier data of the HC with the developed record by BS to the CS.
- CS forwards the medical information of P to the appreciated D in order of sequence number.

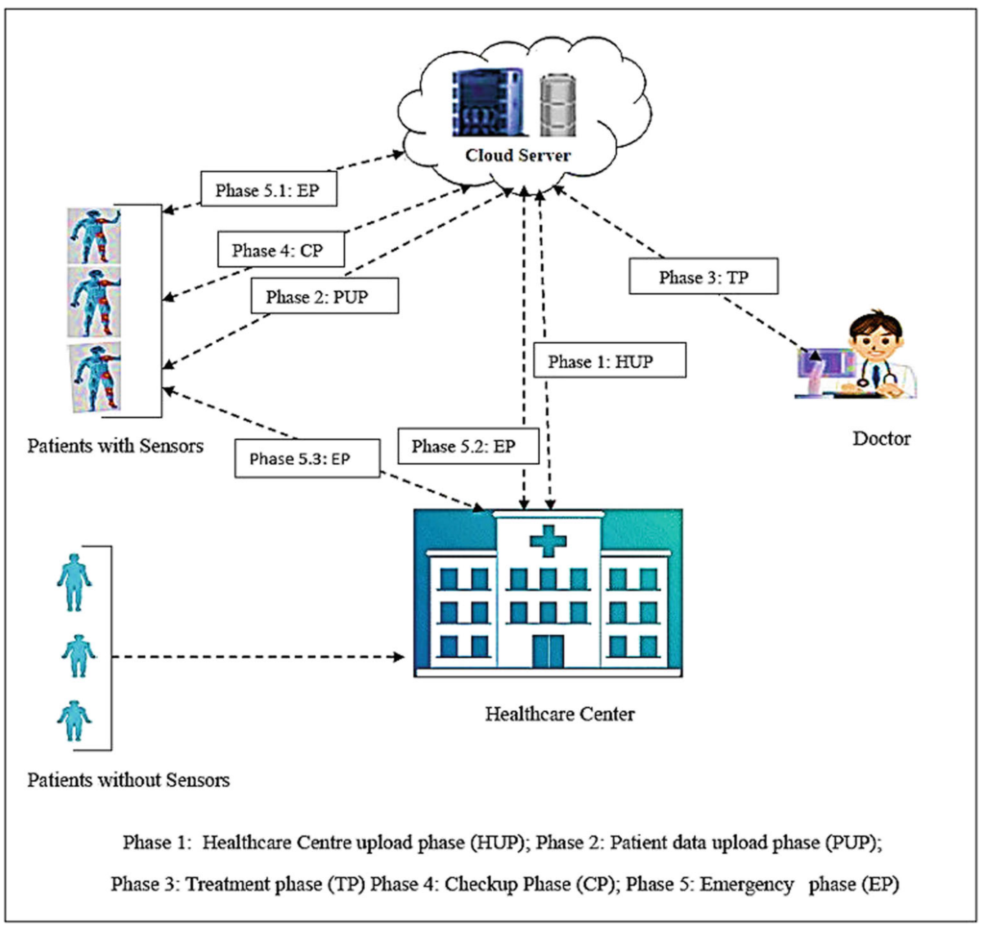


Fig. 4 Protocol architecture and authentication progress with ordering of phases

- D executes medical treatment by looking into the medical data and uploads latest information with the digital signature to the CS.
- CS sends the final medical report to P.

Protocol description

This scheme contains of five phases: (1) Healthcare center upload phase, (2) Patient data upload phase, (3) Treatment phase, (4) Checkup phase, and (5) Emergency phase. The details are as follows:

Healthcare center upload phase (HUP)

The patient registers herself/himself in HC, and HC assigns OTP and a dynamic pseudo random identity SID to P through secure mobile device. In this phase, HC performs mutual authentication with CS and uploads the P's medical report to CS as displayed in the Fig. 5 and expressed as below:

Step 1. The healthcare center generates inspection report $M_H = (ID_P, Data_P)$, random number $r \in Z_q^\star$, and inputs unique identity ID_H and r. Furthermore, HC sends $M_1 = \{ID_H, r, T_{H1}\}$ to CS via a secure channel.

Step 2. On collecting message, CS verifies $T_{C1} - T_{H1} \leq \triangle T$. If it does not hold, the CS terminates the session. Otherwise, generates random number $x \in Z_q^\star$ and computes $H_1 = h(ID_H \| r \| x)$, $A = ID_H \oplus x$, $H_2 = h(H_1 \| A \| r)$. Further, generates another random number $b \in Z_q^\star$ and sends message $M_2 = \{H_2, A, b, T_{C2}\}$ to HC via public channel.

Step 3. On getting messages, HC checks $T_{H2} - T_{C2} \leq \triangle T$. If it does not hold, HC terminates the session. Otherwise, computes $y = A \oplus ID_H$, $H_3 = h(ID_H \| r \| y)$, $H_2^* = h(H_3 \| A \| r)$ and verifies whether $H_2^* =?H_2$ hold or not. If it does not hold, HC exits the session. Otherwise, HC authenticates CS and generates random number $a \in Z_q^\star$. Further, HC computes $SK_{HC} = h(ID_H \| H_3 \| A \| brg)$, $K_1 = h(ID_P \| OTP \| ID_H)$, encrypts $C_H = E_{K_1}(M_H)$, computes $MD_H = h(M_H)$, $Sig_H = S_{PR_H}(MD_H)$, $H_4 = h(SK_{HC} \| C_H \| Sig_H \| abg \| T_{H3})$ and again encrypts $C_1 = E_{SK_{HC}}(ID_P, a, C_H, H_4, Sig_H, SID, T_{H3})$. Finally, the HC sends ,message $M_3 = \{C_1, T_{H3}\}$ to the CS via public channel.

Step 4. Upon receiving message, the CS verifies $T_{C3} - T_{H3} \leq \triangle T$. If it does not hold, CS terminate the session. Otherwise, computes $SK_{CH} = h(ID_H \| H_1 \| A \| brg)$, decrypts $(ID_P, a, C_H, H_4, Sig_H, SID, T_{H3}) = D_{SK_{CH}}(C_1)$, computes $H_5 = h(SK_{CH} \| C_H \| Sig_H \| abg \| T_{H3})$ and verifies whether $H_5 =?H_4$ hold or not. If it does, CS authenticates HC and CS stores ID_P, C_H, Sig_H and SID. Otherwise, CS terminates the session.

Patient data upload phase (PUP)

The patient requests to BS, to gather the updated fitness information, and arranges it to the P through the mobile device securely. The P makes the request using his/her

Fig. 5 Healthcare center upload phase (HUP)

Healthcare Center HC	Cloud Server CS
$M_H = (ID_P, Data_P)$ Generates $r \in Z_q^\star$ Inputs ID_H and r $M_1 = \{ID_H, r, T_{H1}\}$ $\cdots\cdots\longrightarrow$ (via secure channel)	
	$T_{C1} - T_{H1} \leq \triangle T$ $x \in Z_q^\star$ $H_1 = h(ID_H \| r \| x)$ $A = ID_H \oplus x$ $H_2 = h(H_1 \| A \| r)$ $b \in Z_q^\star$ $M_2 = \{H_2, A, b, T_{C2}\}$ $\longleftarrow\cdots\cdots$ (via public channel)
$T_{H2} - T_{C2} \leq \triangle T$ $y = A \oplus ID_H$ $H_3 = h(ID_H \| r \| y)$ $H_2^* = h(H_3 \| A \| r)$ $H_2^* =?H_2$ $a \in Z_q^\star$ $SK_{HC} = h(ID_H \| H_3 \| A \| brg)$ $K_1 = h(ID_P \| OTP \| ID_H)$ $C_H = E_{K_1}(M_H)$ $MD_H = h(M_H)$ $Sig_H = S_{PR_H}(MD_H)$ $H_4 = h(SK_{HC} \| C_H \| Sig_H \| abg \| T_{H3})$ $C_1 = E_{SK_{HC}}(ID_P, a, C_H, H_4, Sig_H, SID, T_{H3})$ $M_3 = \{C_1, T_{H3}\}$ $\cdots\cdots\longrightarrow$ (via public channel)	
	$T_{C3} - T_{H3} \leq \triangle T$ $SK_{CH} = h(ID_H \| H_1 \| A \| brg)$ $(ID_P, a, C_H, H_4, Sig_H, SID, T_{H3}) = D_{SK_{CH}}(C_1)$ $H_5 = h(SK_{CH} \| C_H \| Sig_H \| abg \| T_{H3})$ $H_5 =?H_4$ Stores ID_P, C_H, Sig_H, SID

identity ID_P and OTP of the mobile device. CS contributes an engagement sequence number sn_i, inspection data report M_H to P as displayed in the Fig. 6 and discussed as below:

Step 1. The patient gets report $M_B = (ID_P, Data_B)$ from body sensor via secure mobile device. Then, P takes his/her identity ID_P and dynamic pseudo random SID and sends message $M_4 = \{ID_P, SID, T_{P1}\}$ to CS via secure channel.

Step 2. Upon collecting message, CS verifies $T_{C4} - T_{P1} \leq \triangle T$. If it does not hold, CS terminate the session. Otherwise, computes $N = sn_i \oplus h(SID\|ID_P)$, generates random number $c \in Z_q^\star$, computes $H_6 = h(SID\|sn_i\|C_H\|Sig_H\|T_{C5})$, encrypts $L_1 = E_{sn_i}(Sig_H, C_H, H_6, ID_H, c, T_{C5})$ and sends message $M_5 = \{L_1, N, T_{C5}\}$ to P.

Step 3. On receiving message, P checks $T_{P2} - T_{C5} \leq \triangle T$. If it does not hold, P stops the session. Otherwise, computes $N_1 = N \oplus h(SID\| ID_P)$, decrypts $(Sig_H, C_H, H_6, ID_H, c, T_{C5}) = D_{N_1}(L_1)$, computes $H_7 = h(SID\|N_1\|C_H \|Sig_H \|T_{C5})$ and verifies whether $H_7 =?H_6$ hold or not. If it does not hold, P exits the session. Otherwise he/she authenticates CS, and generates random number $d \in Z_q^\star$. Further, computes $SK_{PC} = h(ID_P\|ID_H\|N_1\|H_7\|cdg)$ and $K_2 = h(ID_P\|OTP\|ID_H)$. Moreover, P decrypts the report $M_H^* = D_{K_2}(C_H)$ and checks whether $M_H^* =?M_H$ hold or not. If it does not hold, P exits the session. Otherwise, computes $MD_H^* = V_{PU_H}(Sig_H)$ and verifies $MD_H^* = ?MD_H$. if it hold, computes $K_{PC} = h(ID_P\|ID_H\|N_1)$, encrypts $C_P = E_{K_{PC}}(M_H, M_B)$, computes $MD_P = h(M_B)$, makes digital signature $Sig_P = S_{PR_P}(MD_P)$, computes $H_8 = h(SK_{PC}\| C_P\|Sig_P\|cdg\|T_{P3})$, again encrypts $L_2 = E_{N_1}(d, H_8, Sig_P, C_P, ID_P, T_{P3})$ and sends message $M_6 = \{L_2, T_{P3}\}$ to CS via public channel.

Step 4. Upon receiving message, the CS verifies $T_{C6} - T_{P3} \leq \triangle T$. If it does not hold, CS terminate the session. Otherwise, decrypts $(d, H_8, Sig_P, C_P, ID_P, T_{P3}) = D_{sn_i}(L_2)$, computes session key $SK_{CP} = h(ID_P\|ID_H\|sn_i\| H_6\|cdg)$, $H_9 = h(SK_{CP} \|C_P\|Sig_P\|cdg\|T_{P3})$ and checks whether $H_9 =?H_8$ hold or not. If it does, CS authenticates P and stores C_P, Sig_P. Otherwise, terminates the session.

Treatment phase (TP)

In this phase, the doctor and cloud server authenticates to each other and the doctor performs treatment of the patients. If they are valid entities, the cloud server uses the identity of doctor ID_D to find all of the D's requests by P, who have prepared medical appointments, and forwards the P's treatment description to doctor as displayed in the Fig. 7 and described as below:

Step 1. The Doctor generates random number $e \in Z_q^\star$ and sends message $M_7 = \{ID_D, e, T_{D1}\}$ to the CS via a secure channel.

Fig. 6 Patient data upload phase (PUP)

Patient P	Cloud Server CS
$M_B = (ID_P, Data_B)$ Inputs ID_P, SID $M_4 = \{ID_P, SID, T_{P1}\}$ $\cdots\cdots\cdots\longrightarrow$ (via secure channel)	
	$T_{C4} - T_{P1} \leq \triangle T$ $N = sn_i \oplus h(SID\|ID_P)$ $c \in Z_q^\star$ $H_6 = h(SID\|sn_i\|C_H\|Sig_H\|T_{C5})$ $L_1 = E_{sn_i}(Sig_H, C_H, H_6, ID_H, c, T_{C5})$ $M_5 = \{L_1, N, T_{C5}\}$ $\longleftarrow\cdots\cdots\cdots$ (via public channel)
$T_{P2} - T_{C5} \leq \triangle T$ $N_1 = N \oplus h(SID\|ID_P)$ $(Sig_H, C_H, H_6, ID_H, c, T_{C5}) = D_{N_1}(L_1)$ $H_7 = h(SID\|N_1\|C_H\|Sig_H\|T_{C5})$ $H_7 =?H_6$ $d \in Z_q^\star$ $SK_{PC} = h(ID_P\|ID_H\|N_1\|H_7\|cdg)$ $K_2 = h(ID_P\|OTP\|ID_H)$ $M_H^* = D_{K_2}(C_H)$ $M_H^* =?M_H$ $MD_H^* = V_{PU_H}(Sig_H)$ $MD_H^* =?MD_H$ $K_{PC} = h(ID_P\|ID_H\|N_1)$ $C_P = E_{K_{PC}}(M_H, M_B)$ $MD_P = h(M_B)$ $Sig_P = S_{PR_P}(MD_P)$ $H_8 = h(SK_{PC}\|C_P\|Sig_P\|cdg\|T_{P3})$ $L_2 = E_{N_1}(d, H_8, Sig_P, C_P, ID_P, T_{P3})$ $M_6 = \{L_2, T_{P3}\}$ $\cdots\cdots\cdots\longrightarrow$ (via public channel)	
	$T_{C6} - T_{P3} \leq \triangle T$ $(d, H_8, Sig_P, C_P, ID_P, T_{P3}) = D_{sn_i}(L_2)$ $SK_{CP} = h(ID_P\|ID_H\|sn_i\|H_6\|cdg)$ $H_9 = h(SK_{CP}\|C_P\|Sig_P\|cdg\|T_{P3})$ $H_9 =?H_8$ Stores C_P, ID_P, Sig_P

Fig. 7 Treatment phase (TP)

Doctor D	Cloud Server CS
Inputs ID_D and random number $e \in Z_q^\star$ $M_7 = \{ID_D, e, T_{D1}\}$ $\cdots\cdots\cdots\cdots\longrightarrow$ (via secure channel)	$T_{C7} - T_{D1} \le \triangle T$ $N_2 = sn_i \oplus h(SID\|ID_D\|ID_P)$ $f \in Z_q^\star$ $H_{10} = h(e\|sn_i\|Sig_P\|C_P\|T_{C8})$ $L_3 = E_{sn_i}(Sig_P, C_P, ID_P, ID_H, H_{10}, f, T_{C8})$ $M_8 = \{L_3, N_2, T_{C8}\}$ $\longleftarrow\cdots\cdots\cdots\cdots$ (via public channel)
$T_{D2} - T_{C8} \le \triangle T$ $N_3 = N_2 \oplus h(SID\|ID_D\|ID_P)$ $(Sig_P, C_P, ID_P, ID_H, H_{10}, f, T_{C8}) = D_{N_3}(L_3)$ $H_{11} = h(e\|N_3\|Sig_P\|C_P\|T_{C8})$ $H_{11} =?H_{10}$ $SK_{DC} = h(ID_P\|ID_D\|N_3\|H_{11}\|efg)$ $K_{DC} = h(ID_P\|ID_H\|N_3)$ $(M_H, M_B) = D_{K_{DC}}(C_P)$ $MD_P^* = V_{PU_P}(Sig_P)$ $MD_P^* =?MD_P$ $M_D = (ID_P, Data_D)$ $C_D = E_{K_{DC}}(M_H, M_B, M_D)$ $MD_D = h(M_D)$ $Sig_D = S_{PR_D}(MD_D)$ $H_{12} = h(SK_{DC}\|C_D\|Sig_D\|Sig_P\|efg\|T_{D3})$ $L_4 = E_{N_3}(Sig_D, C_D, H_{12}, T_{D3})$ $M_9 = \{L_4, T_{D3}\}$ $\cdots\cdots\cdots\cdots\longrightarrow$ via public channel	$T_{C9} - T_{D3} \le \triangle T$ $(Sig_D, C_D, H_{12}, T_{D3}) = E_{sn_i}(L_4)$ $SK_{CD} = h(ID_P\|ID_D\|sn_i\|H_{10}\|efg)$ $H_{13} = h(SK_{CD}\|C_D\|Sig_D\|Sig_P\|efg\|T_{D3})$ $H_{13} =?H_{12}$ Stores C_D, Sig_D

Step 2. On receiving message, CS verifies $T_{C7} - T_{D1} \le \Delta T$. If it does not hold, CS exits the session. Otherwise, computes $N_2 = sn_i \oplus h(SID\| ID_D\|ID_P)$, generates random number $f \in Z_q^\star$, computes $H_{10} = h(e\|sn_i\|Sig_P\|C_P\|T_{C8})$, $L_3 = E_{sn_i}(Sig_P, C_P, ID_P, ID_H, H_{10}, f, T_{C8})$. Further, sends the message $M_8 = \{L_3, N_2, T_{C8}\}$ to D via public channel.

Step 3. On receiving message, D checks $T_{D2} - T_{C8} \le \Delta T$. If it does not hold, D terminates the session. Otherwise, computes $N_3 = N_2 \oplus h(SID\| ID_D\|ID_P)$ and decrypts $(Sig_P, C_P, ID_P, ID_H, H_{10}, f, T_{C8}) = D_{N_3}(L_3)$. Further, D computes $H_{11} = h(e\|N_3\|Sig_P\|C_P\|T_{C8})$, verifies whether $H_{11} =?H_{10}$ hold or not. If it does not hold, D exits the session. Otherwise, he/she authenticates to the CS and computes $SK_{DC} = h(ID_P\| ID_D\|N_3\|H_{11}\|efg)$, $K_{DC} = h(ID_P\|ID_H\|N_3)$. Moreover, D decrypts the report as $(M_H, M_B) = D_{K_{DC}}(C_P)$, computes $MD_P^* = V_{PU_P}(Sig_P)$ and checks whether $MD_P^* =?MD_P$ hold or not. If it does not hold, then D stops the session. Otherwise, D makes a medical diagnosis report based on $M_D = (ID_P, Data_D)$ and encrypts $C_D = E_{K_{DC}}(M_H, M_B, M_D)$. Furthermore, D computes $MD_D = h(M_D)$ and makes digital signature message $Sig_D = S_{PR_D}(MD_D)$. In additionally, D computes $H_{12} = h(SK_{DC}\|C_D\ \|Sig_D\|Sig_P\|efg\|T_{D3})$, encrypts L_4=$E_{N_3}(Sig_D,\ C_D,\ H_{12},\ T_{D3})$ and sends message $M_9 = \{L_4, T_{D3}\}$ to CS via public network.

Step 4. On accepting message, CS verifies $T_{C9} - T_{D3} \le \Delta T$. If it does not hold, CS terminates the session. Otherwise, CS decrypts $(Sig_D, C_D, H_{12}, T_{D3}) = E_{sn_i}(L_4)$, computes $SK_{CD} = h(ID_P \|ID_D\|sn_i\|H_{10}\|efg)$, $H_{13} = h(SK_{CD}\|C_D\ \|Sig_D \|Sig_P\|efg\|T_{D3})$ and checks whether $H_{13} =?H_{12}$ hold or not. If it does, CS authenticates D and stores C_D, Sig_D. Otherwise, D terminates the session.

Checkup phase (CP)

In this phase, P and CS authenticate to each other. Then, CS sends the encrypted the report to P. The detail description of this phase as displayed in the Fig. 8 and explained as below:

Step 1. The patient takes his/her identity ID_P, as request and sends message $M_{10} = \{ID_P, request, T_{P4}\}$ to CS via a secure channel.

Step 2. Upon collecting message, CS verifies $T_{C10} - T_{P4} \le \Delta T$. If it does not hold, CS exits the session. Otherwise, computes $N_4 = h(ID_P\|sn_i)$. Further, generates random number $f_1 \in Z_q^\star$, computes $H_{14} = h(ID_P\|C_D\|Sig_D\|Sig_P\|T_{C11})$, $L_5 = E_{N_4}(H_{14}, Sig_D, C_D, f_1, T_{C11})$ and sends message $M_{11} = \{L_5, T_{C11}\}$ to P via public channel.

Step 3. On receiving message, P checks $T_{P4} - T_{C11} \le \Delta T$. If it does not hold, P stop the session. Otherwise, computes $N_5 = h(ID_P\|N_1)$, decrypts $(H_{14}, Sig_D, C_D, f_1, T_{C11}) = D_{N_5}(L_5)$, and computes $H_{14}^* = h(ID_P\|C_D\|Sig_D\|Sig_P\|\ T_{C11})$, and verifies whether $H_{14}^* =?H_{14}$ hold or not. If it does not hold, D stops the session. Otherwise he/she authenticates CS. Then, P decrypts the report as $(M_H, M_B, M_D) = D_{K_{PC}}\ (C_D)$,

Fig. 8 Checkup phase (CP)

Patient P	Cloud Server CS
Inputs $ID_P, request$ $M_{10} = \{ID_P, request, T_{P4}\}$ $\cdots\cdots\cdots\longrightarrow$ (via secure channel)	$T_{C10} - T_{P4} \leq \triangle T$ $N_4 = h(ID_P \| sn_i)$ $f_1 \in Z_q^\star$ $H_{14} = h(ID_P \| C_D \| Sig_D \| Sig_P \| T_{C11})$ $L_5 = E_{N_4}(H_{14}, Sig_D, C_D, f_1, T_{C11})$ $M_{11} = \{L_5, T_{C11}\}$ $\longleftarrow\cdots\cdots\cdots$ (via public channel)
$T_{P5} - T_{C11} \leq \triangle T$ $N_5 = h(ID_P \| N_1)$ $(H_{14}, Sig_D, C_D, f_1, T_{C11}) = D_{N_5}(L_5)$ $H_{14}^* = h(ID_P \| C_D \| Sig_D \| Sig_P \| T_{C11})$ $H_{14}^* =?H_{14}$ $(M_H, M_B, M_D) = D_{K_{PC}}(C_D)$ $MD_D^* = V_{PU_D}(Sig_D)$ $MD_D^* =?h(M_D)$ $f_2 \in Z_q^\star$ $C_2 = E_{K_{PC}}(M_H, M_B, M_D, f_2)$ $H_{15} = h(N_5 \| C_2 \| Sig_P \| Sig_D \| f_1 f_2 g \| T_{P6})$ $L_6 = E_{N_5}(C_2, H_{15}, f_2, T_{P6})$ $M_{12} = \{L_6, T_{P6}\}$ $\cdots\cdots\cdots\longrightarrow$ (via public channel)	$T_{C12} - T_{P6} \leq \triangle T$ $(C_2, H_{15}, f_2, T_{P6}) = D_{N_4}(L_6)$ $H_{15}^* = h(N_4 \| C_2 \| Sig_P \| Sig_D \| f_1 f_2 g \| T_{P6})$ $H_{15}^* =?H_{15}$ stores C_2

and computes $MD_D^* = V_{PU_D}(Sig_D)$ to checks whether $MD_D^* =?h(M_D)$ hold or not. If it does not hold, then stops the session. Otherwise, generates random number $f_2 \in Z_q^\star$, encrypts $C_2 = E_{K_{PC}}(M_H, M_B, M_D, f_2)$, computes $H_{15} = h(N_5 \| C_2 \| Sig_P \| Sig_D \| f_1 f_2 g \| T_{P6})$, again encrypts $L_6 = E_{N_5}(C_2, H_{15}, f_2, T_{P6})$ and sends message $M_{12} = \{L_6, T_{P6}\}$ to CS via public channel.

Step 4. Upon receiving message, CS verifies $T_{C12} - T_{P5} \leq \triangle T$. If it does not hold, CS terminates the session. Otherwise, CS decrypts computes $(C_2, H_{15}, f_2, T_{P6}) = D_{N_4}(L_6)$, computes $H_{15}^* = h(N_4 \| C_2 \| Sig_P \| Sig_D \| f_1 f_2 g \| T_{P6})$ and also verifies whether $H_{15}^* =?H_{15}$ hold or not. If it does, CS authenticates P and stores C_2. Otherwise, terminates the session.

Emergency phase (EP)

The patients use the body sensors network, and relocate the regular medical information to the cloud server. If the patient has an emergency, then the patient inputs his/her identity, sequence number and request sends to CS. Then, CS sends the information to HC. After verification the doctor provides treatment to the patients. The detail description of this phase as shown in the Fig. 9 and discussed as below:

Step 1. P inputs his/her identity $ID_P, N_5, request$, computes $H_{16} = h(ID_P \| N_5 \| T_{EP1})$, encrypts $L_7 = E_{N_5}(H_{16}, T_{EP1})$ and sends message $M_{E1}=\{L_7, T_{EP1}\}$ to CS via public channel.

Step 2. On receiving message, CS verifies $T_{EC1} - T_{EP1} \leq \triangle T$. If it does not hold, CS terminates the session. Otherwise, decrypts $(H_{16}, T_{EP1}) = D_{N_4}(L_7)$, computes $H_{16}^* = h(ID_P \| N_4 \| T_{EP1})$ and checks whether $H_{16}^* =?H_{16}$ hold or not. If it does not hold, CS terminates the session. Otherwise, generates random number $p \in Z_q^\star$, computes $H_{17} = h(ID_P \| ID_H \| Sig_H \| Sig_P \| T_{EC2})$, $L_8 = E_{SK_{CH}}(H_{17}, p, Sig_P, ID_P, T_{EC2})$ and sends message $M_{E2} = \{L_8, T_{EC2}\}$ to HC via public network.

Step 3. On receiving messages, HC verifies $T_{EH1} - T_{EC2} \leq \triangle T$. If it does not hold, HC terminates the session. Otherwise, decrypts $(H_{17}, p, Sig_P, ID_P, T_{EC2}) = D_{SK_{HC}}(L_8)$, computes $H_{17}^* = h(ID_P \| ID_H \| Sig_H \| Sig_P \| T_{EC2})$ and verifies $H_{17}^* =?H_{17}$ hold or not. If it does not hold, HC terminates the session. Otherwise, generates random number $s \in Z_q^\star$, computes $SK_{HP} = h(ID_P \| ID_H \| Sig_H \| Sig_P \| psg)$, $H_{18} = h(ID_P \| ID_H \| p \| s \| T_{EH2})$, $L_9 = E_{SK_{HC}}(s, H_{18}, T_{EH2})$ and sends message $M_{E3} = \{L_9, T_{EH2}\}$ to CS via public channel.

Step 4. On receiving message, CS verifies $T_{EC3} - T_{EH2} \leq \triangle T$. If it does not hold, CS stops the session. Otherwise, CS decrypts $(s, H_{18}, T_{EH2}) = D_{SK_{CH}}(L_9)$, computes $H_{18}^* = h(ID_P \| ID_H \| p \| s \| T_{EH2})$ and verifies $H_{18}^* =?H_{18}$ hold or not. If it does not hold, CS terminates the session. Otherwise, authenticates HC by computing $H_{19} = h(Sig_P \| Sig_H \| pg \| sg \| T_{EC4})$, encrypts $L_{10} = E_{N_4}(ID_H, p, s, H_{19}, T_{EC4})$ and sends message $M_{E4} = \{L_{10}, T_{EC4}\}$ to P via public network.

Step 5. Upon receiving message, P verifies $T_{EP2} - T_{EC4} \leq \triangle T$. If it does not hold, P terminates the session. Otherwise, decrypts $(ID_H, p, s, H_{19}, T_{EC4}) = D_{N_5}(L_{10})$, computes $H_{19}^* = h(Sig_P \| Sig_H \| pg \| sg \| T_{EC4})$, and verifies whether $H_{19}^* =?H_{19}$ hold or not. If it does not hold, P terminates the session.

Patient P	Cloud server CS	Healthcare Center HC
Inputs ID_P, N_5, request $H_{16} = h(ID_P \| N_5 \| T_{EP1})$ $L_7 = E_{N_5}(H_{16}, T_{EP1})$ $M_{E1} = \{L_7, T_{EP1}\}$ $\cdots\cdots\cdots\cdots\rightarrow$ (via public channel)	$T_{EC1} - T_{EP1} \leq \triangle T$ $(H_{16}, T_{EP1}) = D_{N_4}(L_7)$ $H^*_{16} = h(ID_P \| N_4 \| T_{EP1})$ $H^*_{16} =?H_{16}$ $p \in Z^*_q$ $H_{17} = h(ID_P \| ID_H \| Sig_H \| Sig_P \| T_{EC2})$ $L_8 = E_{SK_{CH}}(H_{17}, p, Sig_P, ID_P, T_{EC2})$ $M_{E2} = \{L_8, T_{EC2}\}$ $\cdots\cdots\cdots\cdots\rightarrow$ (via public channel)	
		$T_{EH1} - T_{EC2} \leq \triangle T$ $(H_{17}, p, Sig_P, ID_P, T_{EC2}) = D_{SK_{HC}}(L_8)$ $H^*_{17} = h(ID_P \| ID_H \| Sig_H \| Sig_P \| T_{EC2})$ $H^*_{17} =?H_{17}$ $s \in Z^*_q$ $SK_{HP} = h(ID_P \| ID_H \| Sig_H \| Sig_P \| psg)$ $H_{18} = h(ID_P \| ID_H \| p \| s \| T_{EH2})$ $L_9 = E_{SK_{HC}}(s, H_{18}, T_{EH2})$ $M_{E3} = \{L_9, T_{EH2}\}$ $\leftarrow\cdots\cdots\cdots\cdots$ (via public channel)
	$T_{EC3} - T_{EH2} \leq \triangle T$ $(s, H_{18}, T_{EH2}) = D_{SK_{CH}}(L_9)$ $H^*_{18} = h(ID_P \| ID_H \| p \| s \| T_{EH2})$ $H^*_{18} =?H_{18}$ $H_{19} = h(Sig_P \| Sig_H \| pg \| sg \| T_{EC4})$ $L_{10} = E_{N_4}(ID_H, p, s, H_{19}, T_{EC4})$ $M_{E4} = \{L_{10}, T_{EC4}\}$ $\leftarrow\cdots\cdots\cdots\cdots$ (via public channel)	
$T_{EP2} - T_{EC4} \leq \triangle T$ $(ID_H, p, s, H_{19}, T_{EC4}) = D_{N_5}(L_{10})$ $H^*_{19} = h(Sig_P \| Sig_H \| pg \| sg \| T_{EC4})$ $H^*_{19} =?H_{19}$ $SK_{HP} = h(ID_P \| ID_H \| Sig_H \| Sig_P \| psg)$		

Fig. 9 Emergency phase (EP)

Otherwise, P authenticates CS and computes session key $SK_{HP} = h(ID_P \| ID_H \| Sig_H \| Sig_P \| psg)$.

Security proof

Formal proof of the proposed protocol

Theorem: Patient data upload phase (PUP) of our protocol $\mathcal{P}$ employees a additive cyclic group G on an elliptic curve with a large prime order q. $\mathcal{N}$ is the size of one time password dictionary $\mathcal{D}$. If adversary E makes no more than Q_s send queries, Q_h hash queries, and Q_e execute queries, then

$$Adv^{pfs-ake}_{\mathcal{P}}(E) \leq \frac{O(Q_h)^2 + O(Q_s + Q_e)^2}{2^l} + \frac{O(Q_s + Q_e)^2}{(q-1)} + \frac{O(Q_h) + O(Q_s)}{2^{l-1}} + \frac{O(Q_s)}{\mathcal{N}} + O(Q_h(Q_s + Q_e)^2 + 1) \times Adv^{ECDDH}_E(t')$$

Where $t' = t + (O(Q_e) + O(Q_s))T_M$ and T_M is the time of one multiplication in G.

Proof: We prove this theorem with the help of a sequence of games. There are total eight games from G_0 to G_7. $Succ_j$ is the action for adversary E accurately guessing the coin s through the investigation session in Game G_j. Since, there is one patient P in these games, E want to computes or guesses P's identity ID_P. We have to discuss the games following as:

- *Game* G_0: This game is the actual game against the proposed authentication scheme of PUP with the random oracle model, from the definition, we have

$$Adv^{pfs-ake}_{\mathcal{P}}(E) = 2Pro[Succ_0] - 1 \quad (1)$$

  Furthermore, If various atypical circumstances occur, a random s^* is called as a report. The list of the atypical circumstances as follows:

  - The game exit or cancels or since E does not present the predicted s^*.
  - More queries than the prearranged upper bound are used by E.
  - More time than the deliberated upper bound is used by E.

- *Game* G_1: In this game, we take addition of all counterfeited queries. Moreover, there are only three lists to accumulate the answers to the queries.

  - L_H: For the answer to all hash queries.
  - L_P: For the transcription of the communication.
  - L_E: It is for the respond of the two random oracles queried precisely by adversary E.

  The queries are established in Fig. 10. According to the situations mentioned above, *Game* G_1 and

Fig. 10 Simulation of queries

Simulation of queries
For a hash query, if there exists a record data (s, r) in L_H, r is returned as the reply. Otherwise, the simulator chooses a random string $r \in \{0,1\}^l$, reply with r and sets (s, r) in L_H. For $h_1(s)$, like steps have to be completed the record $(1, s, r)$
For a $Send(P^i, INIT)$ query, the simulator executes the following steps: $M_B = (ID_P, Data_B)$ Inputs ID_P, SID Return $M_4 = \{ID_P, SID, T_{P1}\}$ as the answer.
For a Send (P^i, CS^j, M_4) query, the simulator does the following steps: Verify $T_{C4} - T_{P1} \leq \triangle T$, Computes $N = sn_i \oplus h(SID\|ID_P)$ Generates Checks $c \in Z_q^\star$ Computes $H_6 = h(SID\|sn_i\|C_H\|Sig_H\|T_{C5})$ Encrypts $L_1 = E_{sn_i}(Sig_H, C_H, H_6, ID_H, c, T_{C5})$ Then answer the query with message $M_5 = \{L_1, N, T_{C5}\}$
For a $Send$ (CS^j, P^i, M_5) query, the simulator computes the following steps: Verify $T_{P2} - T_{C5} \leq \triangle T$ Computes $N_1 = N \oplus h(SID\|ID_P)$ Decrypts $(Sig_H, C_H, H_6, ID_H, c, T_{C5}) = D_{N_1}(L_1)$ Computes $H_7 = h(SID\|N_1\|C_H\|Sig_H\|T_{C5})$ Verifies $H_7 =?H_6$ If does not hold, exit the session. Otherwise Generates $d \in Z_q^\star$ Computes $SK_{PC} = h(ID_P\|ID_H\|N_1\|H_7\|cdg)$, $K_2 = h(ID_P\|OTP\|ID_H)$, $M_H^* = D_{K_2}(C_H)$ Verifies $M_H^* =?M_H$ If does not hold, exit the session. Otherwise computes $MD_H^* = V_{PU_H}(Sig_H)$ Again verifies $MD_H^* =?MD_H$ If does not hold, terminates the session. Otherwise, computes $K_{PC} = h(ID_P\|ID_H\|N_1)$, $C_P = E_{K_{PC}}(M_H, M_B)$,$MD_P = h(M_B)$ $Sig_P = S_{PR_P}(MD_P)$, $H_8 = h(SK_{PC}\|C_P\|Sig_P\|cdg\|T_{P3})$ Encrypts $L_2 = E_{N_1}(d, H_8, Sig_P, C_P, ID_P, T_{P3})$ Then answer the query with message $M_6 = \{L_2, T_{P3}\}$
For a Send (P^i, CS^j, M_6) query, the simulator does the following steps: Verify $T_{C6} - T_{P3} \leq \triangle T$ Decrypts $(d, H_8, Sig_P, C_P, ID_P, T_{P3}) = D_{sn_i}(L_2)$ Computes $SK_{CP} = h(ID_P\|ID_H\|sn_i\|H_6\|cdg)$, $H_9 = h(SK_{CP}\|C_P\|Sig_P\|cdg\|T_{P3})$ Checks $H_9 =?H_8$ If does not verify, rejects the query. Otherwise, Stores C_P, Sig_P
For an $Execute$ (P^i, CS^j) query, all $Send$ queries are consecutively completed and the massage (M_4, M_5, M_6) is returned.
For a $Reveal$ (I^K) query, if the occurrence I^K has been established and produced a session key, return SK_{CP} or SK_{PC}. Otherwise a $\perp$ is the reply.
For a $Corrupt$ (I^K) query, all the information of I^K is ouput.
For a $Test$ (I^K) query, if I^K is not $pfs - fresh$, return $\perp$. Otherwise a coin s is tossed If $s = 0$, a random string with the length l is returned If $s = 1$,the exact session key is returned.

$Game\ G_0$ are indistinguishable and we can notice that

$$Pro[Succ_1] = Pro[Succ_0] \quad (2)$$

- $Game\ G_2$: In this game, we avoid the collisions in the transcriptions. There are three types of collisions. As stated in the birthday paradox, we display the probabilities of them:
  - $c, d \in Z_q^\star$ may collide particular session and upper bound for the case is

    $$\frac{O(Q_s + Q_e)^2}{2(q-1)}$$

  - Dynamic pseudo random identity $SID \in Z_q^\star$ may collide in different session and upper bound for the case is

    $$\frac{O(Q_s + Q_e)^2}{2^{l+1}}$$

  - The hash function results may collide and upper bound for the case is

    $$\frac{O(Q_h)^2}{2^{l+1}}.$$

From $Game\ G_2$ and $Game\ G_1$ are indistinguishable except the collisions occur. We observe that

$$|Pro[Succ_2] - Pro[Succ_1]| \leq \frac{O(Q_s + Q_e)^2}{2(q-1)} + \frac{O(Q_h)^2 + O(Q_s + Q_e)^2}{2^{l+1}} \quad (3)$$

- $Game\ G_3$: In this game, we consider the probability of the attack that adversary E fakes message M_4. Since the simulator permits the answer as CS, we attach some steps on $Send(P^i, CS^j, M_4)$ the simulator wants to verify if $M_4 \in L_P$. If it is failing the query will stop. Here $Game\ G_3$ and $Game\ G_2$ are indistinguishable if the verifiers are under deliberation. We can obtain

$$|Pro[Succ_3] - Pro[Succ_2]| \leq \frac{O(Q_s)}{2^l} \quad (4)$$

- $Game\ G_4$: In this game, we deal with the probability of the attack that adversary E fakes message M_5. Since the simulator permits the answer as P, we attach some steps on $Send(CS^j, P^i, M_5)$ the simulator wants to verify if $M_5 \in L_P$ and, $(\star\|ID_P, \star)$, $(\star\|sn_i\|C_H\|Sig_H\|T_{C5}, H_6) \in L_E$. If it is failing the query will stop. Here $Game\ G_4$ and $Game\ G_3$ are indistinguishable if the verifiers are under deliberation. We can obtain

$$|Pro[Succ_4] - Pro[Succ_3]| \leq \frac{O(Q_s + Q_e)}{2^l} \quad (5)$$

- $Game\ G_5$: In this game, we consider the probability of fake message M_6. Since the simulator gives the response as the CS. We append some steps on $Send(P^i, CS^j, M_6)$. The simulator wants to validate if $M_6 \in L_P$ and $(\star\|ID_P, \star)$, $(\star\| \star \|C_H\|Sig_H\| T_{C5}, H_7)$, $(1, ID_P\| ID_H\| \star \| \star \|\star, \star)$, $(ID_P\|ID_H\|\star, K_{PC})$, $(\star\|C_P\| Sig_P\| \star \|T_{P3}, H_8) \in L_E$. If it is failing the query will stop. Here $Game\ G_5$ and $Game\ G_4$ are indistinguishable if the verifiers are under deliberation. We can obtain $(\star\|ID_P, \star)$, $(\star\| \star \|C_H\|Sig_H\|T_{C5}, H_7)$, $(1, ID_P\|ID_H\| \star \| \star \|\star, \star)$, $(ID_P\|ID_H\|\star, K_{PC})$, $(\star\|C_P\|Sig_P\| \star \|T_{P3}, H_8) \in L_E$. So we found that

$$|Pro[Succ_5] - Pro[Succ_4]| \leq \frac{O(Q_h + Q_s)}{2^l} \quad (6)$$

- $Game\ G_6$: In this game, we take on ECGDHP. If adversary E can obtain the actual session key via hash oracle and be the success, we judge that E crack the problem. We adjust the hash oracle as follows: On one occasion E queries $(1, ID_P\|ID_H\|sn_i\|X\|X, X)$, $(X\|C_P\|Sig_P\|X\|X\|T_{P3})$, the simulator first verifies if $(1, ID_P\|ID_H\|sn_i\|H_6\|\star, SK_{PC})$, $(SK_{PC}\|C_P\|Sig_P\| \star\| T_{P3}) \in L_E$. If it is in, SK_{PC} is returned. Otherwise, the simulator utilizes the ECGDHP oracle to evaluator $X\ =?ECGDHP(cg, dg)$. If it is unsuccessful, the query is dropped. Otherwise, the simulator selects a random string $SK_{PC} \in \{0, 1\}^l$ outputs it and adds $(1, ID_P\|ID_H\|sn_i\| X\|X, SK_{PC})$ to L_E.

  We analyze this game with two characteristics: the active attack and the passive attack. First E asks a $Corrupt$ query and obtains all information:

  - It is for online OTP guessing attacks. E could embrace judge a OTP from the dictionary. Since E can utilize Send query Q_s and the size of OTP dictionary is $\mathcal{N}$, the probability for E to guess the exact OTP by loading a session is bounded by $\frac{Q_s}{\mathcal{N}}$.
  - For the passive attacks. There are two methods in this case:
    - ◇ The first is E finds information, he/she asks *Execute* queries. At the end E asks the hash query to succeed and cracks ECGDHP. We can find cdg. From L_E with the probability $1/Q_h$. So the probability for this case is bounded by $Q_h Adv_E^{ECGDHP}(t + O(Q_e)T_M)$.
    - ◇ The other is E asks Send queries successively. Like the first kind of a passive attack, we can find that the upper bound probability of this case is $Q_h Adv_E^{ECGDHP}(t + O(Q_s)T_M)$

The probability for the two types of the passive attack is

$$Q_h Adv_E^{ECGDHP}(t + O(Q_e)T_M) + Q_h Adv_E^{ECGDHP} \times(t + O(Q_s)T_M) \leq Q_h Adv_E^{ECGDHP}.(2t + [O(Q_s) + O(Q_e)]T_M)$$

Let $t' = (2t + [O(Q_s) + O(Q_e)]T_M)$, then we got

$$|Pro[Succ_6] - Pro[Succ_5]| \leq \frac{Q_s}{\mathcal{N}} + Q_h Adv_E^{ECGDHP}(t') \quad (7)$$

- $Game\ G_7$: This game is for perfect forward security. E can determine all planned $Corrupt$ queries. But according to the approach of $sfs - fresh$, *Corrupt* queries should be asked after the *Test* query. So adversary E can only exploit the historical queries and transcripts. In this last game, we can obtain $(1, ID_P\|ID_H\|sn_i\|X\| X, SK_{PC})$ in L_E. The probability of getting cg and dg in the same session is $1/(Q_s + Q_e)^2$ and we have

$$|Pro[Succ_7]-Pro[Succ_6]| \leq Q_h(Q_s+Q_e)^2 Adv_E^{ECGDHP}(t') \quad (8)$$

Combining the above games, there is no benefit for E to guess the session key and $Pro[Succ_6] = \frac{1}{2}$. Taking the sum of all results of these games, Theorem can be proved.

Remark : Similarly the formal security proof of Healthcare center upload phase (HUP), Treatment phase (TP) and Emergency phase (EP) can also be analyzed.

Informal security analysis

In this phase, we evaluated that the prospective scheme has the capability to resist different cryptographic attacks.

Proposition 1 *The proposed framework could assure the man-in-the-middle attack.*

Proof In our protocol, every step of every phase has time-stamp condition $T_i - T_j \leq \triangle T$ and hash condition $H_i = ?H_j$. If possible, an attacker inter in these phases after verifying the times-tamp condition then, check the hash condition H_i=?H_j which not possible by the definition one way hash function is secure. Thus adversary will not get success in any phase. Therefore, our protocol protects the man-in middle attack. □

Proposition 2 *The proposed protocol could assure the patient anonymity.*

Proof We describe the patient anonymity in each authentication phase:

- During HUP, the patient identity ID_P is encrypted by screening original identity. Here, patient identity ID_P in encrypted with session key $SK_{HC} = h(ID_H \| H_1 \| A \| brg)$, as get $C_1 = E_{SK_{HC}}(ID_P, a, C_H, H_4, Sig_H, SID, T_{H3})$ and only be decrypt by cloud server $(ID_P, a, C_H, H_4, Sig_H, SID, T_{H3}) = D_{SK_{CH}}(C_1)$ with containing session key $SK_{CH} = h(ID_H \| H_1 \| A \| brg)$ and verifies the condition $H_5 = ?H_4$ then, stores ID_P, C_H, Sig_H, SID.
- During PUP, the patient identity ID_P is encrypted by screening original identity. Here, patient identity ID_P in encrypted with session key $N_1 = N \oplus h(ID_(SID \| ID_P)$, as get $L_2 = E_{N_1}(d, H_8, Sig_P, C_Pa, C_H, ID_P, T_{P3})$ and only be decrypt by cloud server $(d, H_8, Sig_P, C_P, a, C_H, ID_P, T_{P3}) = D_{sn_i}(L_2)$, where, sn_i is the sequence number of patient and verified hash condition $H_9 = ?H_8$ then, stores C_P, ID_P, Sig_P.

Similarly, the patient anonymity is hold in TP, CP and EP. Therefore, our protocol provides patient anonymity. □

Proposition 3 *The proposed protocol could protect the strong replay attack.*

Proof Replay attack is a common attack in authentication procedure. However, the common countermeasures are time-stamps and random number. In our protocol, we adopt the time-stamp and random number as a counter-measure in every steps of every phases, receiver will check it. the timestamps is legal or not by checking the valid time interval with equation $T_i - T_j \leq \triangle T$, where $\triangle T$ is the valid time interval. Further, random number used random number to computing session key, hash values and different keys. Therefore, replay attack could not work in the proposed protocol. □

Proposition 4 *The proposed protocol could provide the known-key security property.*

Proof The proposed scheme describes the different session keys in different phases:

- During HUP, the HC computes session key $SK_{HC} = h(ID_H \| H_3 \| A \| brg)$ and CS computes session key $SK_{CH} = h(ID_H \| H_1 \| A \| brg)$.
- During PUP, P computes session key $SK_{PC} = h(ID_P \| ID_H \| N_1 \| H_7 \| cdg)$ and CS computes $SK_{CP} = h(ID_P \| ID_H \| sn_i \| H_6 \| cdg)$.
- During TP, D computes session key $SK_{DC} = h(ID_P \| ID_D \| N_3 \| H_{11} \| efg)$ and CS computes $SK_{CD} = h(ID_P \| ID_D \| sn_i \| H_{10} \| efg)$.
- During EP, P computes session key $SK_{PH} = h(ID_P \| ID_H \| Sig_P \| Sig_H \| psg)$ and HC computes $SK_{HP} = h(ID_P \| ID_H \| Sig_P \| Sig_H \| psg)$.

The proposed protocol, presents different session key in a different phase. Even if the adversary abducts the earlier session key, she/he cannot computes the session key for the new phase. Thus, the proposed scheme has the quality of known-key security. □

Proposition 5 *The proposed framework could protect the data Confidentiality.*

Proof Confidentiality is the method to security on transferring of data from the attacker. The encryption and description of data are given below:

- During HUP, HC encrypts the report as $C_H = E_{K_1}(M_H)$ with using key $K_1 = h(ID_P \| OTP \| ID_P)$ and upload to cloud server.
- During PUP, the patient encrypts $C_P = E_{K_{PC}}$ (M_H, M_B) with using key $K_{PC} = h(ID_P \| ID_D \| N_1)$ and upload to CS.

- During TP, D encrypts ciphertext $C_D = E_{K_{DC}}(M_H, M_B, M_D)$ with using key $K_{DC} = h(ID_P \| ID_D \| N_3)$ and upload to CS.
- During CP, P decrypts $C_2 = E_{K_{PC}}(M_H, M_B, M_D, f_2)$ using key $K_{PC} = h(ID_P \| ID_D \| N_1)$ and upload to CS.

Thus, if an attacker tries to find data information during the transmission, she/he encrypts message which cannot be decrypted without the key and the hash value of inputs, as the definition of hash function is secure and one way. Therefore, the proposed protocol protect the confidentiality. □

Proposition 6 *The proposed scheme could protect the data Non-repudiation.*

Proof The proposed protocol describes data Non-repudiation in different phases:

- During HUP, HC signs a message $Sig_H = S_{PR_H}(MD_H)$.
- During PUP, P verified HC's signature by computing $MD_H^* = V_{PU_H}(Sig_H)$ and Verifies if $MD_H^* \stackrel{?}{=} MD_H$ hold or not. After that, P computes signature $Sig_P = S_{PR_P}(MD_P)$.
- During TP, D verified P's signature by computing $MD_P^* = V_{PU_P}(Sig_P)$, checks whether $MD_P^* =?M_P$ hold or not and makes signature $Sig_D = S_{PR_D}(MD_D)$.
- During CP, P verified D's signature by computing $MD_D^* = V_{PU_D}(Sig_D)$, checks whether $MD_D^* = ?h(MD_D)$ hold or not.

Thus, the patient verifies the health records. If the medical data have similar complications, the responsible person cannot be refused. The non-repudiation facts are stored in the cloud. Therefore, our proposed protocol protested data non-repudiation (Table 2). □

Proposition 7 *The proposed protocol could provide Message authentication.*

Proof Message authentication is a method used to authenticate the integrity of the information. We describe message authentication in different phases below as:

- In HUP, HC receives message $M_2 = \{H_2, A, b, T_{C2}\}$ and verifies the validity by checking time-stamps condition $T_{H2} - T_{P2} \leq \Delta T$ and hash function $H^* = ?H_2$. Similarly, CS receives message $M_3 = \{C_1, T_{H3}\}$ and verifies the validity by checking timestamps condition $T_{C3} - T_{H3} \leq \Delta T$, and hash function $H_5 = ?H_4$. If any attacker endeavors alter any change of the message CS will recognize it.
- In PUP, P receives message $M_5 = \{L_1, N, T_{C5}\}$, verifies the validity by checking time-stamps condition $T_{P2} - T_{C5} \leq \Delta T$, hash condition $H_7 =?H_6$ and $M_H^* =?M_H$, $MD_H^* =?MD_H$. Similarly, CS receives message $M_6 = \{L_2, T_{P3}\}$ and verifies the validity by checking time-stamps condition $T_{C6} - T_{P3} \leq \Delta T$ and hash condition $H_9 =?H_8$. If any of the validation fails message will not be established.
- In TP, D receives message $M_8 = \{L_3, N_2, T_{C8}\}$ and verifies the validity by checking the time-stamp condition $T_{D2} - T_{C8} \leq \Delta T$ and hash function $H_{11} =?H_{10}$ and $MD_P^* =?MD_P$. Further, CS receives message $M_9 = \{L_4, T_{D3}\}$ and verifies the validity by checking the time-stamp condition $T_{C9} - T_{D3} \leq \Delta T$

Table 2 Comparison of Functionality features

Security Attack	Chen et al. [15]	Yang et al. [16]	Chiou et al. [17]	Mohit et al. [39]	Proposed
P_1	√	√	√	√	√
P_2	×	×	×	×	√
P_3	√	√	√	√	√
P_4	×	×	×	√	√
P_5	√	√	√	√	√
P_6	√	√	√	√	√
P_7	√	√	√	√	√
P_8	√	×	√	√	√
P_9	×	√	√	√	√
P_{10}	√	√	√	√	√
P_{11}	√	√	√	×	√
P_{12}	√	×	×	×	√

Note $\Longrightarrow$ √: Attributes protected by the protocol, ×: Attributes not protected by the protocol, P_1:Man-in-the-middle attack, P_2:Patient anonymity, P_3:Replay attack, P_4:Stolen mobile device attack, P_5:Known-key security property, P_6:Data Confidentiality, P_7:Data Non-repudiation, P_8:Message authentication, P_9:Session key security, P_{10}:Off-line password/ identity guessing attack, P_{11}:Many logged-in patient's attack and P_{12}:Session key security.

and hash function $H_{13} =?H_{12}$. Message authentication verified between the D and the CS.

- In CP, P receives message $M_{11} = \{L_5, T_{C11}\}$ and verifies the validity by checking the time-stamp condition $T_{P4} - T_{C11} \leq \triangle T$ hash function $H^*_{14} =?H_{14}$ and $MD^*_D =?h(M_D)$. Again CS receives message $M_{12} = \{L_6, T_{P6}\}$ and verifies the validity by checking time-stamps condition $T_{C12} - T_{P6} \leq \triangle T$ and hash condition $H^*_{15} =?H_{15}$. If any of the verification fails message will not be accepted.
- In EP, HC receives message $M_{E2} = \{L_7, T_{EC2}\}$ and verifies the validity by checking time-stamps condition $T_{EH1} - T_{EC2} \leq \triangle T$ and hash condition $H^*_{17} = ?H_{17}$. CS receives message $M_{E1} = \{L_8, T_{EP1}\}$ and verifies the validity by checking time-stamps condition $T_{EC1} - T_{EP1} \leq \triangle T$ and hash condition $H^*_{16} =?H_{16}$, and CS also receives message $M_{E3} = \{L_9, T_{EH2}\}$ and verifies the validity by checking time-stamps condition $T_{EC3} - T_{EH2} \leq \triangle T$ and hash condition $H^*_{18} =?H_{18}$. Further, P receives message $M_{E4} = \{L_{10}, T_{EC4}\}$ and verifies the validity by checking time-stamps condition $T_{EP2} - T_{EC4} \leq \triangle T$, hash function $H^*_{19} =?H_{19}$. If any of the verification fails message will not be accepted.

Therefore, this protocol protects the message authentication in every phase. □

Proposition 8 *The proposed protocol could protect the impersonation attack.*

Proof We discussed the details of impersonation attacks in HUP as below:

- Any E tries to masquerade as a valid CS, and eavesdrop the transferred information message $M_2 = \{H_2, A, b, T_{C2}\}$ and tries to computes H_2, where $H_1 = h(ID_H\|r\|x)$, $A = ID_H \oplus x$, $H_2 = h(H_1\|A\|r)$. E cannot compute H_1, which the hash attribute of parameters ID_H, r, x where ID_H is the unique identity of the HC, r is a random number which generated by the HC and x is the secret value of CS. Note that, guessing of all three value at the same time is impossible. Further, E cannot compute H_2 which the hash value of H_1, A, r. Thus the adversary cannot impersonate as valid CS.
- E tries to impersonate as a valid a HC. If E breaks the time-stamp condition $T_{H2} - T_{C2} \leq \triangle T$, guesses the identity of HC as $ID_E = ID_H$ and random number r. Then, computes $y_E = A \oplus ID_E$, $H_{E3} = h(ID_E\|r\|y_E)$ and $H_{E2} = h(H_3\|A\|r)$. Verifies the condition $H^*_{E2} =?H_2$ which not hold, as H^*_2 is the hash value of parameters H_3, A, and r. By the definition of hash function, H^*_2 is the secure value. Thus, E cannot impersonate as the valid HC.

Similarly, impersonation attacks not possible in PUP, TP, CP and EP phases. Hence, the protocol is secured against the impersonation attack. □

Proposition 9 *The proposed scheme could protect the session key security.*

Proof The proposed protocol having four session keys those are compute between 1) HC and CS, 2) P and CS, 3) D and CS, and 4) P and HC. Here, we have discuss the session key security of HUP. However, the approach is the similar other remaining phases.

- In HUP, the session key between the HC and CS is $SK_{HC} = SK_{HC}$, where $SK_{HC} = h(ID_H\| H_3\|A\|brg)$ and $SK_{CH} = h(ID_H\|H_1\|A\|brg)$. E cannot computes the session key SK_{HC} or SK_{HC}, where $H_1 = h(ID_H\|r\|x)$, $H_3 = h(ID_H\| r\|y)$, $A = ID_H \oplus x$, $H^*_2 = h(ID_H\|A\|r)$. With the help of Proposition 8, H_1 and H^*_2 cannot be computed by E. Further, For $b, r \in Z^\star_q$ and g is the generator of G, given (g, bg, rg), then compute brg is hard to the group G by ECCDHP in the elliptic curve cryptography. Thus, the session key can only be generated by the authenticated party.

Similarly, session key generated in PUP, TP and EP. Hence the proposed scheme could protect the session key □

Proposition 10 *The proposed framework could protect the stolen mobile device attack.*

Proof Suppose that E stolen the mobile phone of the authorized P, E cannot find any secret communication of the P. As the mobile phone accepts the message, which is reachable only by inputs valid identity of P and OTP of mobile phone. In PUP, P computes key $K_2 = h(ID_P\|OTP\|ID_H)$. Where OTP is the unique one time password of P's mobile device and h(.) is the one way hash function which is secure by define it. Therefore, E cannot break the system even if she/he gets the mobile device of the valid patient.

Similarly, in HUP adversary does not break the system even if she/he grabs the mobile phone of the registered P. Thus, the proposed framework assures the stolen mobile device attack. □

Proposition 11 *The proposed protocol could protect the off-line password/ identity guessing attack.*

Proof We discussed this attack in PUP. If possible, any E interprets in PUP and guesses the identity ID_E of valid P, then compute $N_E = N \oplus h(SID\|ID_E)$ and key $K_{E1} = h(ID_E\|OTP\|ID_H)$, Where OTP is the unique OTP of P. Thus, $N_E \neq N_1$ and $K_E \neq K_2$ because SID is unique for each patient, $N_1 = N \oplus h(SID\|ID_P)$ and ID_H is identity of HC which is also unique. On the other hand, if possible he/she guesses one time password OTP_E of legal patient, then computes $K_{E2} = h(ID_E\|OTP_E\|ID_H)$. As a result, $K_{E2} \neq K_2$. Hence, off-line password/identity guessing attack cannot work in PUP of the proposed protocol.

Similarly, off-line password/identity guessing attack not possible in HUP, TP, CP and EP. Thus, off-line password/ identity guessing attack not possible in the proposed framework. □

Proposition 12 *The proposed framework could resist many logged-in patient attack.*

Proof We discussed this attack in PUP. Suppose that many adversaries $E_1, E_2, E_3, ...E_j.. ..E_m$ having same identity ID_P and sends messages $\{ID_P, SID, T_{E_1}\}, \{ID_P, SID, T_{E_2}\}, \{ID_P, SID, T_{E_3}\}, ...\{ID_P, SID, T_{E_j}\}, ...\{ID_P, SID, T_{E_m}\}$ to CS, where T_{E_j} is current message sending time of j^{th} adversary. Here, we discuss about only adversary E_j. On receiving message from adversary E_j, then CS verifies $T_{C4} - T_{E_j} \leq \Delta T$, If possibly hold, the CS computes $N = sn_i \oplus h(SID\|ID_P)$, generates random number $c \in Z_q^\star$, computes $H_6 = h(SID\|sn_i\|C_H\|Sig_H\|T_{C5})$, encrypts $L_1 = E_{sn_i}(Sig_H, C_H, H_6, ID_H, c, T_{C5})$ and sends message $M_5 = \{L_1, N, T_{C5}\}$ to E_j via public channel. On receiving message E_j verifies $T_{E_j} - T_{C5} \leq \Delta T$ and computes $N_1^E = N \oplus h(SID\|ID_P)$, decrypts $(Sig_H, C_H, H_6, ID_H, c, T_{C5}) = D_{N_1^E}(L_1)$, computes $H_7^E = h(SID\|N_j^E\|C_H\|Sig_H\|T_{C5})$. Here, $H_7^{E_j} \neq H_6$ as sn_i, SID, T_{C5} are different and unique for each patients. Thus many logged-in patient's attack is not work in PUP.

Similarly, many logged-in patient attack does not work in in CP and EP. Therefore, our protocol protected against many logged-in patient's attack. □

Performance analysis

In this section, we estimate performance of the proposed framework with the relevant schemes worked in cloud environment for secure medical data communication, such as Chen et al. [15], Chiou et al. [17], Chen-Yang et al. [16] and Mohit et al. [39] protocols. The comparison performed in all the phases of framework like HUP, PUP, TP, CP and EP bellow as:

We have adopted different cryptographic operations in this paper based on the information applicable in Chiou et al. [17] to test the computation cost of the proposed protocol still existing relevant research. Chiou et al. [17], Windows 7 OS and Android phone used and the system structure of mobile phone is Android 4.4.4KTU84P along with a 2GB RAM and 1.8 GHz processor. The configurations of computer system is Windows 7, Professional with an

Table 3 Computation cost of our protocol with related protocols

Protocol	Chen et al. [15]	Yang et al. [16]	Chiou et al. [17]	Mohit et al. [39]	Proposed
HUP	$1T_{Sign}$+$1T_M$+$2T_P$ +$4T_S$+$2T_H$+$3T_A$	$1T_{Sign}$+$4T_M$+$4T_P$ +$2T_S$+$6T_H$+$1T_A$	$1T_{Sign}$+$3T_P$ +$2T_S$+$7T_H$	$1T_{Sign}$+$3T_S$ +$10T_H$	$1T_{Sign}$+$3T_S$ +$10T_H$
PUP	$1T_M$+$2T_P$ +$4T_S$+$2T_H$+$3T_A$	$1T_{Sign}$+$4T_M$+$4T_P$ +$3T_S$+$6T_H$+$1T_A$	$1T_{Sign}$+$4T_P$ +$2T_S$+$12T_H$	$2T_{Sign}$+$2T_S$ +$11T_H$	$2T_{Sign}$+$6T_S$ +$11T_H$
TP	$2T_{Sign}$+$1T_M$+$2T_P$ +$7T_S$+$2T_H$+$4T_A$	$2T_{Sign}$+$4T_M$+$4T_P$ +$4T_S$+$6T_H$	$2T_{Sign}$+$4T_M$ +$4T_P$+$4T_S$+$6T_H$	$2T_{Sign}$+$2T_S$ +$9T_H$	$2T_{Sign}$+$6T_S$ +$10T_H$
CP	NA	NA	$1T_{Sign}$+$2T_P$ $2T_S$+$8T_H$	$1T_{Sign}$+$2T_S$ +$5T_H$	$1T_{Sign}$+$6T_S$ +$10T_H$
EP	NA	$2T_{Sign}$+$2T_P$ $6T_S$+$4T_H$	NA	NA	$8T_S$+$10T_H$
Total cost	$3T_{Sign}$+$3T_M$+$6T_P$ +$15T_S$+$6T_H$+$10T_A$ ≈4.7091 sec	$6T_{Sign}$+$12T_M$+$15T_P$ +$15T_S$+$22T_H$+$2T_A$ ≈4.379 sec	$5T_{Sign}$+$4T_M$+$13T_P$ +$10T_S$+$33T_H$ ≈2.7705 sec	$6T_{Sign}$+$9T_S$ +$35T_H$ ≈2.086 sec	$6T_{Sign}$+$9T_S$ +$39T_H$ ≈2.179 sec

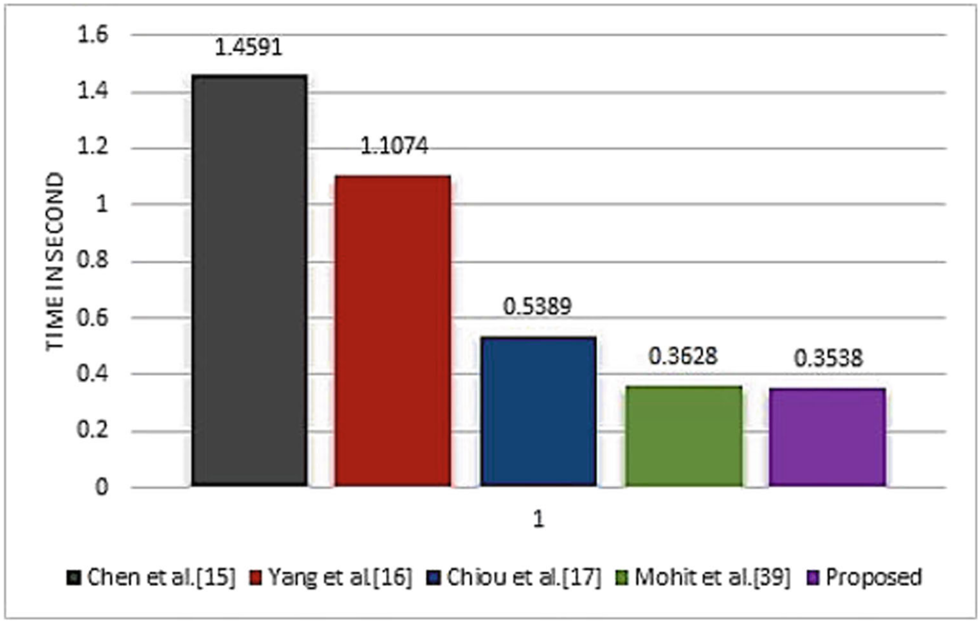


Fig. 11 Computation cost in HUP

Intel (R) core (TM) 2 Quad CPU Q8300, 2GB RAM and @2.50Hz. The execution time in second for the different time complexity symbols are as follows:

- T_{Sign} :The time for calculating execute/verify a signature ($T_{Sign} \approx 0.3317sec$).
- T_A : the time for calculating asymmetric encryption/decryption operation ($T_A \approx 0.3057sec$).
- T_M : the time for calculating multiplication operation ($T_M \approx 0.0503sec$).
- T_P : the time for calculating a bilinear pairing operation ($T_P \approx 0.0621sec$).
- T_S : the time for calculating symmetric encryption/decryption operation ($T_S \approx 0.0087sec$).
- T_H : the time for calculating one-way hash function ($T_H \approx 0.0005sec$).

Table 3 recaps the computation cost of the proposed scheme with relevant schemes. It is famous that the computational cost of XOR ($\oplus$) and concatenation ($\|$) operations treated as imperceptible analyzed to other operations like as symmetric encryption/decryption, multiplication, pairing free, bilinear pairing, etc. There are following observation about computation cost and security information:

- In Fig. 11 shows that the computation cost of the HUP of the protocol is $\approx$ 0.3538sec which is greater than Mohit et al.'s scheme[39]. The proposed scheme is

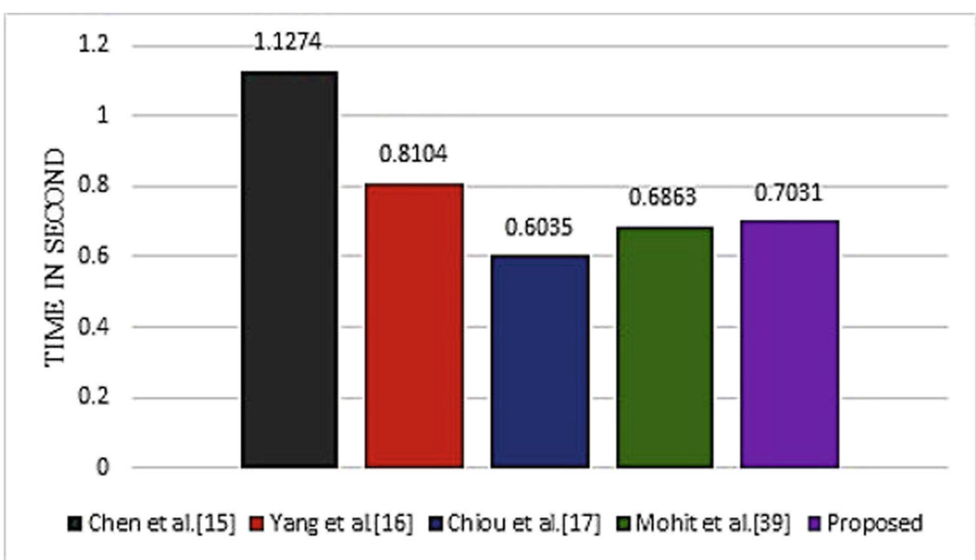


Fig. 12 Computation cost in PUP

Fig. 13 Computation cost in TP

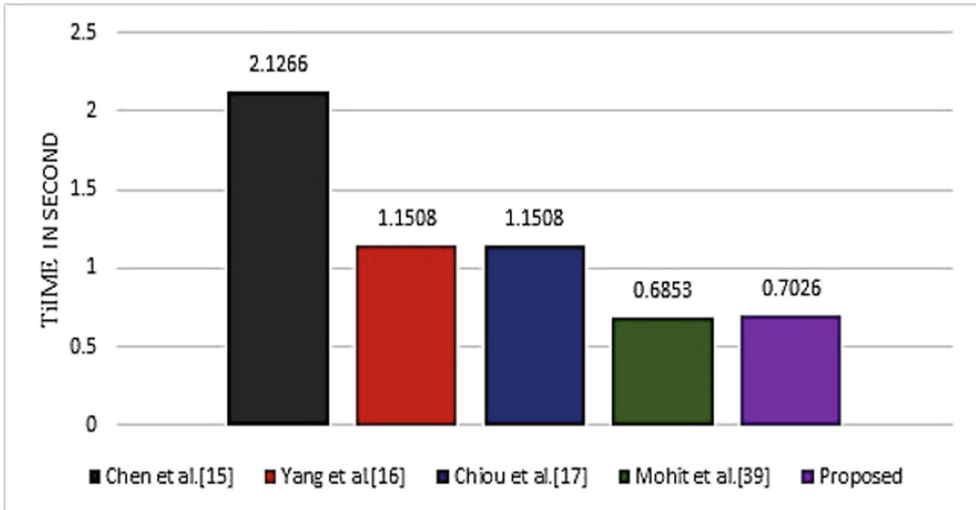


Fig. 14 Computation cost in CP

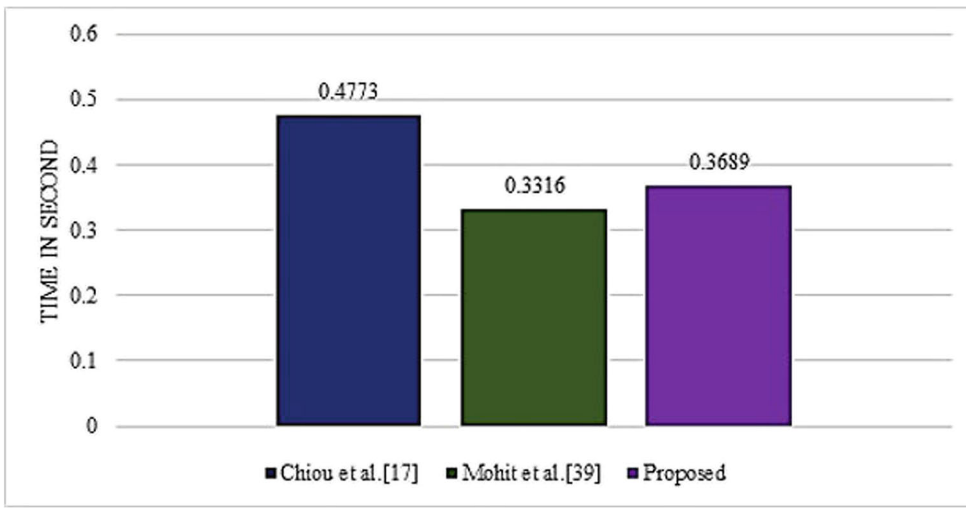
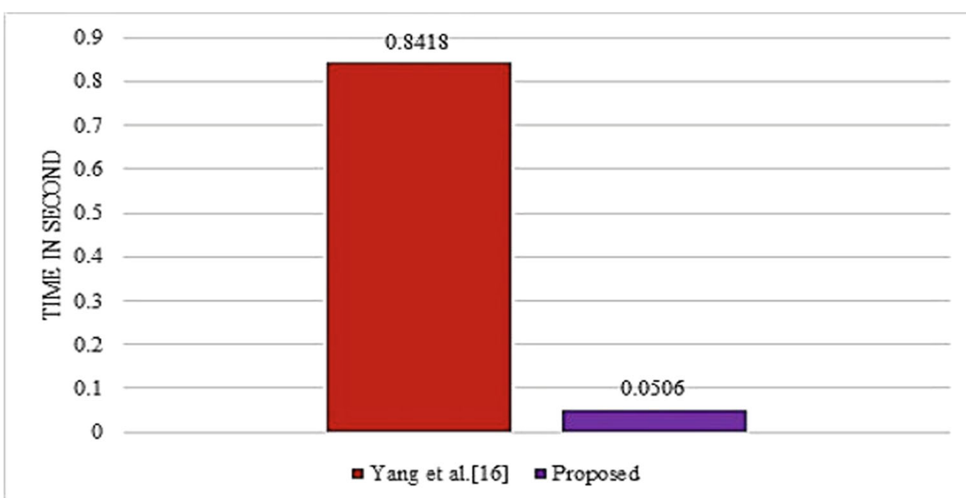


Fig. 15 Computation cost in EP



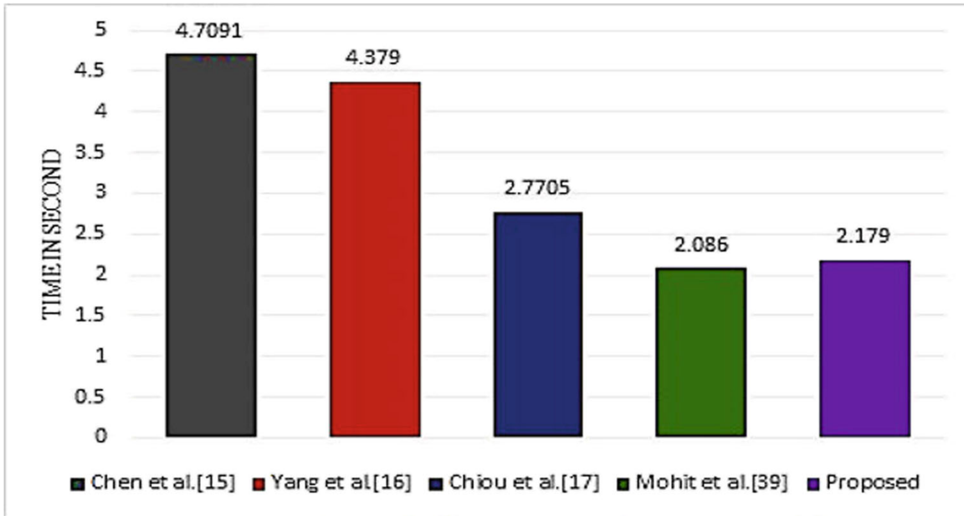


Fig. 16 Total computation cost of our protocol with related protocols in seconds

secured but Mohit et al. and other schemes have security weaknesses.

– In Fig. 12 shows that the computation cost of the PUP of the protocol is ≈ 0.7031sec which is greater than Mohit et al.'s scheme[39], Chiou et al.'s [17] and less than Chen et al.'s [15] and Yang et al.'s [16]. The proposed protocol is secured but other relative schemes have security weaknesses.
– In Fig. 13 shows that the computation cost of the TP of the protocol is ≈ 0.7026sec which is greater than Mohit et al.'s scheme[39]. The proposed framework is secured but Mohit et al. and other schemes have security weaknesses.
– In Fig. 14 shows that the computation cost of the CP of the protocol is ≈ 0.3689sec which is greater than Mohit et al.'s scheme[39] and less than Chiou et al.'s [17]. Therefore, the proposed protocol is secured but Mohit et al.'s and Chiou et al.'s scheme are not secure.
– In Fig. 15 shows that the computation cost of the EP of the protocol is ≈ 0.0506sec which is more less than Yang et al.'s scheme[16]. But in this phase, presented protocol is secured and efficient but Yang et al.'s scheme is not secure and efficient.

Liu et al.'s [57] is lightweight pseudonym authentication scheme for multi-server architecture in TMIS. This work efficient for authentication and key agreement process in TMIS. In the proposed protocol, we used single cloud server and patient, doctor and healthcare center. So, Liu et al.'s scheme is not applicable in this domain. It is clear from Fig. 16 that the the proposed protocol has less computation cost than the earlier protocols worked in a cloud environment for medical communication of data exchange. The computation cost of the proposed protocol is greater than Mohil et al.'s protocol, but Mohit et al.'s scheme has no emergency phase and have some security weaknesses.

Conclusion

The evolution of information technology offers conveniences to humanize medical services, maintaining patients with effectual treatment with enlarged convenience and security. In this paper, we have reviewed Mohit et al.'s mutual authentication scheme described for a TMIS using cloud computing environment. On cryptanalysis, we found that the protocol is susceptible to stolen-verifier attack, many logged-in patient attack, patient anonymity, impersonation attack and fails to protect session key. Then, we proposed an improved, secure and efficient mutual authentication scheme in the same environment. Further, we proved that the proposed protocol provides better security than other previous protocols by the security analysis. The proposed protocol is also profitable in terms of performance like as computation overheads.

References

1. Abdalla, M., Izabachene, M., and Pointcheval, D., Anonymous and transparent gateway-based password-authenticated key exchange. In: *International conference on cryptology and network security*, pp. 133–148. Berlin: Springer, 2008.
2. Abor, P. A., and Agrizzi, D., Healthcare governance and patients' perception of service quality. In: Annual conference on innovations in business & Management, London, pp. 21–23, 2012.
3. Amin, R., Cryptanalysis and efficient dynamic ID based remote user authentication scheme in multi-server environment using smart card. *IJ Netw. Secur.* 18(1):172–181, 2016.
4. Amin, R., and Biswas, G. P., Cryptanalysis and design of a three-party authenticated key exchange protocol using smart card. *Arab. J. Sci. Eng.* 40(11):3135–3149, 2015.

5. Amin, R., and Biswas, G. P., A secure three-factor user authentication and key agreement protocol for tmis with user anonymity. *J. Med. Syst.* 39(8):1–19, 2015.
6. Amin, R., Hafizul Islam, S. K., Biswas, G. P., Khan, M. K., and Kumar, N., A robust and anonymous patient monitoring system using wireless medical sensor networks. *Futur. Gener. Comput. Syst.* 80:483–495, 2018.
7. Amin, R., Sk, H. I., Biswas, G. P., Khan, M. K., and Li, X., Cryptanalysis and enhancement of anonymity preserving remote user mutual authentication and session key agreement scheme for e-health care systems. *J. Med. Syst.* 39(11):1–21, 2015.
8. Bajpai, D., Vardhan, M., Gupta, S., Kumar, R., and Kushwaha, D. S., Security service level agreements based authentication and authorization model for accessing cloud services. In: *Advances in computing and information technology*, pp. 719–728. Berlin: Springer, 2012.
9. Balduzzi, M., Zaddach, J., Balzarotti, D., Kirda, E., and Loureiro, S., A security analysis of amazon's elastic compute cloud service. In: Proceedings of the 27th annual ACM symposium on applied computing, pp. 1427–1434 ACM, 2012.
10. Bresson, E., Chevassut, O., and Pointcheval, D., Security proofs for an efficient password-based key exchange. In: Proceedings of the 10th ACM conference on Computer and communications security, pp. 241–250 ACM, 2003.
11. Cao, B.-Q., Li, B., and Xia, Q.-M., A service-oriented qos-assured and multi-agent cloud computing architecture. In: *IEEE international conference on cloud computing*, pp. 644–649. Berlin: Springer, 2009.
12. Casalicchio, E., and Silvestri, L., Mechanisms for SLA provisioning in cloud-based service providers. *Comput. Netw.* 57(3):795–810, 2013.
13. Chaudhry, S. A., Khan, M. T., Khan, M. K., and Shon, T., A multiserver biometric authentication scheme for tmis using elliptic curve cryptography. *J. Med. Syst.* 40(11):230, 2016.
14. Chaudhry, S. A., Naqvi, H., Shon, T., Sher, M., and Farash, M. S., Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems. *J. Med. Syst.* 39(6):65–75, 2015.
15. Chen, C.-L., Yang, T.-T., Chiang, M.-L., and Shih, T.-F., A privacy authentication scheme based on cloud for medical environment. *J. Med. Syst.* 38(11):1–16, 2014.
16. Chen, C.-L., Yang, T.-T., and Shih, T.-F., A secure medical data exchange protocol based on cloud environment. *J. Med. Syst.* 38(9):1–12, 2014.
17. Chiou, S.-Y., Ying, Z., and Liu, J., Improvement of a privacy authentication scheme based on cloud for medical environment. *J. Med. Syst.* 40(4):1–15, 2016.
18. Debiao, H. E., Jianhua, C., and Rui, Z., A more secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(3):1989–1995, 2012.
19. Gope, P., and Amin, R., A novel reference security model with the situation based access policy for accessing ephr data. *J. Med. Syst.* 40(11):41–53, 2016.
20. Hankerson, D., Menezes, A. J., and Vanstone, S., Guide to elliptic curve cryptography. Springer Science & Business Media, 2006.
21. He, D., Kumar, N., Chen, J., Lee, C.-C., Chilamkurti, N., and Yeo, S.-S., Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimed. Syst.* 21(1):49–60, 2015.
22. He, D., Kumar, N., Shen, H., and Lee, J.-H., One-to-many authentication for access control in mobile pay-TV systems. *Sci. China Inf. Sci.* 59(5):1–14, 2016.
23. He, D., Kumar, N., Wang, H., Wang, L., Choo, K.-K. R., and Vinel, A., A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network. IEEE Transactions on Dependable and Secure Computing, 2016.
24. He, D., and Wang, D., Robust biometrics-based authentication scheme for multiserver environment. *IEEE Syst. J.* 9(3):816–823, 2015.
25. He, D., Zeadally, S., Kumar, N., and Lee, J.-H., Anonymous authentication for wireless body area networks with provable security. *IEEE Syst. J.* 11(4):2590–2601, 2017.
26. Hwang, J.-J., Chuang, H.-K., Hsu, Yi-C., and Wu, C.-H., A business model for cloud computing based on a separate encryption and decryption service. In: International conference on information science and applications (ICISA), pp. 1–7. IEEE, 2011.
27. Islam, S. K., Obaidat, M. S., and Amin, R., An anonymous and provably secure authentication scheme for mobile user. *Int. J. Commun. Syst.* 29(9):1529–1544, 2016.
28. Islam, S. K. H., Amin, R., Biswas, G. P., Farash, M. S., Li, X., and Kumari, S., An improved three party authenticated key exchange protocol using hash function and elliptic curve cryptography for mobile-commerce environments. *J. King Saud Univ. Comput. Inf. Sci.* 29(3):311–324, 2017.
29. Jiang, Q., Ma, J., and Ma, Z., A privacy enhanced authentication scheme for telecare medical information systems. *J. Med. Syst.* 37(1):1–8, 2013.
30. Karati, A., Amin, R., and Biswas, G. P., Provably secure threshold-based abe scheme without bilinear map. *Arab. J. Sci. Eng.* 41(8):3201–3213, 2016.
31. Kumari, S., Khan, M. K., and Kumar, R., Cryptanalysis and improvement of 'a privacy enhanced scheme for telecare medical information systems'. *J. Med. Syst.* 37(4):1–11, 2013.
32. Lee, C.-C., Hsu, C.-W., Lai, Y.-M., and Vasilakos, A., An enhanced mobile-healthcare emergency system based on extended chaotic maps. *J. Med. Syst.*. 37(5):1–12, 2013.
33. Li, C.-T., Lee, C.-C., and Weng, C.-Y., A secure chaotic maps and smart cards based password authentication and key agreement scheme with user anonymity for telecare medicine information systems. *J. Med. Syst.* 38(9):1–11, 2014.
34. Li, X., Niu, J., Karuppiah, M., Kumari, S., and Fan, W. U., Secure and efficient two-factor user authentication scheme with user anonymity for network based e-health care applications. *J. Med. Syst.* 40(12):267–277, 2016.
35. Li, X., Niu, J., Khan, M. K., and Liao, J., An enhanced smart card based remote user password authentication scheme. *J. Netw. Comput. Appl.* 36(5):1365–1371, 2013.
36. Maitra, T., Obaidat, M. S., Amin, R., Islam, S. K., Chaudhry, S. A., and Giri, D., A robust ElGamal based password authentication protocol using smart card for client server communication. International Journal of Communication Systems 30(11), 2017.
37. Mishra, D., Mukhopadhyay, S., Chaturvedi, A., Kumari, S., and Khan, M. K., Cryptanalysis and improvement of Yan others.'s biometric-based authentication scheme for telecare medicine information systems. *J. Med. Syst.* 38(6):1–12, 2014.
38. Mishra, D., Srinivas, J., and Mukhopadhyay, S., A secure and efficient chaotic map-based authenticated key agreement scheme for telecare medicine information systems. *J. Med. Syst.* 38(10):1–10, 2014.
39. Mohit, P., Amin, R., Karati, A., Biswas, G. P., and Khan, M. K., A standard mutual authentication protocol for cloud computing based health care system. *J. Med. Syst.* 41(4):1–13, 2017.
40. Ramez, W. S., Patients' perception of health care quality, satisfaction and behavioral intention: an empirical study in Bahrain. *Int. J. Bus. Soc. Sci.* 3(18):131–141, 2012.
41. Sureshkumar, V., Anitha, R., Rajamanickam, N., and Amin, R., A lightweight two-gateway based payment protocol ensuring accountability and unlinkable anonymity with dynamic identity. *Comput. Electr. Eng.* 57:223–240, 2017.

42. Sutrala, A. K., Das, A. K., Odelu, V., Wazid, M., and Kumari, S., Secure anonymity-preserving password-based user authentication and session key agreement scheme for telecare medicine information systems. *Comput. Methods Prog. Biomed.* 135:167–185, 2016.
43. Tan, Z., An efficient biometrics-based authentication scheme for telecare medicine information systems. *Network* 2(3):200–204, 2013.
44. Tsai, Y. L., Cloud computing security. *Commun. CCISA* 18(2):62–68, 2012.
45. Wazid, M., Das, A. K., Kumari, S., Li, X., and Fan, W. U., Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS. *Secur. Commun. Netw.* 9(13):1983–2001, 2016.
46. Wei, J., Xuexian, H. U., and Liu, W., An improved authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(6):3597–3604, 2012.
47. Wu, Z.-Y., Chung, Y., Lai, F., and Chen, T.-S., A password-based user authentication scheme for the integrated EPR information system. *J. Med. Syst.* 36(2):631–638, 2012.
48. Wu, Z.-Y., Lee, Y.-C., Lai, F., Lee, H.-C., and Chung, Y., A secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(3):1529–1535, 2012.
49. Wu, Z.-Y., Tseng, Y.-J., Chung, Y., Chen, Y.-C., and Lai, F., A reliable user authentication and key agreement scheme for web-based hospital-acquired infection surveillance information system. *J. Med. Syst.* 36(4):2547–2555, 2012.
50. Yan, X., Li, W., Li, P., Wang, J., Hao, X., and Gong, P., “A secure biometrics-based authentication scheme for telecare medicine information systems. *J. Med. Syst.* 37(5):1–6, 2013.
51. Yang, H., Kim, H., and Mtonga, K., An efficient privacy-preserving authentication scheme with adaptive key evolution in remote health monitoring system. *Peer-to-Peer Netw. Appl.* 8(6):1059–1069, 2015.
52. Srinivas, J., Das, A. K., Kumar, N., and Rodrigues, J., Cloud centric authentication for wearable healthcare monitoring system. IEEE Transactions on Dependable and Secure Computing, 2018.
53. Zhu, Z., An efficient authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(6):3833–3838, 2012.
54. Srinivas, J., Mishra, D., and Mukhopadhyay, S., A Mutual Authentication Framework for Wireless Medical Sensor Networks. *J. Med. Syst.* 41(5):80, 2017.
55. Mishra, D., Kumar, V., and Mukhopadhyay, S., A pairing-free identity based authentication framework for cloud computing. In: *International conference on network and system security*, pp. 721–727. Berlin: Springer, 2013.
56. Liu, Y., Guo, W., Fan, C.-I., Chang, L., and Cheng, C., A practical privacy-preserving data aggregation (3PDA) scheme for smart grid. IEEE Transactions on Industrial Informatics, 2018.
57. Liu, Y., Liu, G., Cheng, C., Xia, Z., and Shen, J., A Privacy-Preserving Health Data Aggregation Scheme. *TIIS* 10(8):3852–3864, 2016.
58. Liu, X., Li, Y., Juan, Q. U., and Ding, Y., A lightweight pseudonym authentication and key agreement protocol for multi-medical server architecture in TMIS. *KSII Trans. Internet & Inf. Syst.* 11(2):924–943, 2017.
59. Xu, L., and Fan, W. U., An improved and provable remote user authentication scheme based on elliptic curve cryptosystem with user anonymity. *Secur. Commun. Netw.* 8(2):245–260, 2015.
60. Menezes, A J, Van Oorschot, P C, and Vanstone, S. A., Handbook of applied cryptography. CRC Press, 1996.
61. Wu, F., Lili, X. U, Kumari, S., and Li, X., A new and secure authentication scheme for wireless sensor networks with formal proof. *Peer-to-Peer Netw. Appl.* 10(1):16–30, 2017.