Digital Communications and Networks 8 (2022) 150–161

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

An enhanced scheme for mutual authentication for healthcare services

Salman Shamshad [a], Muhammad Faizan Ayub [a], Khalid Mahmood [a,e,f], Saru Kumari [b,*], Shehzad Ashraf Chaudhry [c], Chien-Ming Chen [d]

[a] Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, 57000, Pakistan
[b] Department of Mathematics, Ch. Charan Singh University, Meerut, Uttar Pradesh, 250004, India
[c] Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey
[d] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China
[e] Riphah School of Computing & Innovation (RSCI), Riphah International University, Lahore Campus, Lahore 55150, Pakistan
[f] Department of Mathematics, University of Padua, 35131 Padua, Italy

ARTICLE INFO

Keywords:
Authentication protocol
Security protocol
Anonymous protocol
Impersonation attack
TMIS

ABSTRACT

With the advent of state-of-art technologies, the Telecare Medicine Information System (TMIS) now offers fast and convenient healthcare services to patients at their doorsteps. However, this architecture engenders new risks and challenges to patients' and the server's confidentiality, integrity and security. In order to avoid any resource abuse and malicious attack, employing an authentication scheme is widely considered as the most effective approach for the TMIS to verify the legitimacy of patients and the server. Therefore, several authentication protocols have been proposed to this end. Very recently, Chaudhry et al. identified that there are vulnerabilities of impersonation attacks in Islam et al.'s scheme. Therefore, they introduced an improved protocol to mitigate those security flaws. Later, Qiu et al. proved that these schemes are vulnerable to the man-in-the-middle, impersonation and offline password guessing attacks. Thus, they introduced an improved scheme based on the fuzzy verifier techniques, which overcome all the security flaws of Chaudhry et al.'s scheme. However, there are still some security flaws in Qiu et al.'s protocol. In this article, we prove that Qiu et al.'s protocol has an incorrect notion of perfect user anonymity and is vulnerable to user impersonation attacks. Therefore, we introduce an improved protocol for authentication, which reduces all the security flaws of Qiu et al.'s protocol. We also make a comparison of our protocol with related protocols, which shows that our introduced protocol is more secure and efficient than previous protocols.

1. Introduction

With the rapid advancement in wireless and networking technologies, the Telecare Medicine Information System (TMIS) now offers convenient and efficient connections between patients, doctors and their corresponding TMIS servers. Offering healthcare and medical services at the doorsteps of patients is considered as a major objective of the TMIS. These services are specially provisioned for patients who are unable to visit the hospital due to any reason. A generic view of the TMIS is illustrated in Fig. 1, which mainly comprises two participants, the user and the TMIS server. The health information such as sugar level or blood pressure of a user is gathered through some smart sensors (e.g. smartwatches, fitness bands, etc.) and delivered to the TMIS server over the Internet. The authorized physician can review the stored data from the TMIS server and examine the user's information with the appropriate method. This stored information helps the physician to have a comprehensive view of a user's health to make adequate decisions.

Even though this system offers great convenience because it saves patients' time and considerable expenses to attain the same medical care they can get at the hospital, it engenders new risks and challenges to patients' and the server's confidentiality, integrity and secrecy because of an insecure and unreliable wireless communication channel. Thus, to preserve the confidentiality, integrity and security of transmitting information in the TMIS, a reasonable authentication protocol is required. Two-factor authentication protocols are considered most suitable for such systems because these protocols ensure authenticity and confidentiality even if an adversary revealed the user's password or the user's smart card was stolen. Moreover, the two-factor protocols have the least

* Corresponding author.
E-mail addresses: salmanshamshad01@gmail.com (S. Shamshad), faizanayub9@gmail.com (M.F. Ayub), khalid.mahmood@cuisahiwal.edu.pk (K. Mahmood), saryusiirohi@gmail.com (S. Kumari), ashraf.shehzad.ch@gmail.com (S.A. Chaudhry), chienmingchen@ieee.org (C.-M. Chen).

https://doi.org/10.1016/j.dcan.2021.07.002
Received 26 September 2019; Received in revised form 3 June 2021; Accepted 7 July 2021
Available online 16 July 2021

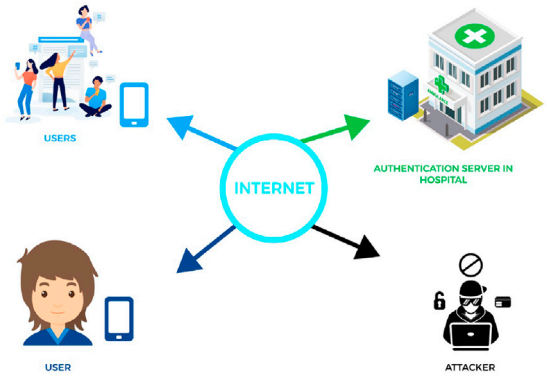

Fig. 1. Generic view of telecare medical information system.

computation and communication overhead. Therefore, a number of authentication protocols have been proposed for the TMIS.

Early on, the scheme originally proposed and developed in 1999 [1] for authentication purposes was based on the HyperText Transport Protocol (HTTP). Soon Yang et al. [2] noticed that it was susceptible to server spoofing and offline password guess attacks and introduced an enhanced scheme on the basis of the Diffie-Hellman key exchange scheme. Later, Huang et al. [3] identified that Yang et al.'s protocol is susceptible to numerous attacks, such as stolen-verifier, offline password guess and denning-Sacco attacks. Furthermore, Huang et al. also realized that Yang et al.'s protocol is inadequate for resource constraint devices due to its huge computational cost. In 2005, on the basis of efforts made by Yang et al., Durlanik and Sogkapinar [4] presented an efficient protocol on the basis of Elliptic Curve Cryptography (ECC). Compared with conventional public-key cryptography, the ECC is able to offer the same level of security with a relatively smaller key size. Consequently, a huge number of authentication protocols have been introduced so far on the basis of the ECC [5–26]. However, many security flaws have also been noticed among these schemes for the TMIS. Therefore, designing a secure key agreement scheme becomes a challenging task.

Xu et al. [27] in 2013 introduced an ECC-based protocol for the TMIS, which is efficient and secure for key agreement and authentication. Their scheme uses dynamic identities for patients to offer anonymity. Later, Islam et al. [28] in 2014 figured out that for realistic considerations, Xu et al.'s protocol is not suitable because (i) it could not resist the replay attack; (ii) it fails to resist the stolen/lost smart card attacks; (iii) its password modification phase is incorrect; and (iv) during the authentication and login phases, it fails to offer mutual authentication. Islam et al. developed an improved ECC-based two-factor scheme to overcome the security deficiencies of Xu et al.'s protocol. Islam et al.'s scheme offers anonymity and provable security. However, in 2015, Chaudhry et al. [29] identified that this protocol is vulnerable to the server and user impersonation attacks. Furthermore, Chaudhry et al. presented an improved protocol that increases the security features of Islam et al.'s scheme by overcoming all the vulnerabilities.

Qiu et al. in 2018 reviewed Chaudhry et al.'s protocol and identified that Chaudhry et al.'s scheme is susceptible to offline password-guessing, server and user masquerade and man-in-middle attacks. Moreover, they found that the offline identity guess is a severe attack for Chaudhry et al.'s protocol. Therefore, Qiu et al. introduced an improved scheme based on fuzzy verifier techniques to overcome all the security flaws of Chaudhry et al.'s scheme. Moreover, the underlying protocol not only prevents all security weaknesses of Islam et al.'s and Chaudhry et al.'s protocols, but also retains the benefits of their schemes as shown in Table 5. Qiu et al. also claim that their protocol is more efficient and secure as compared to its counterparts.

1.1. Our contribution

In this paper, we recall Qiu et al.'s scheme [30] and find that their scheme has an incorrect notion of perfect anonymity. Moreover, it is susceptible to user masquerade attacks. To overcome the above-mentioned problems, we introduce a more efficient and secure protocol on the basis of ECC. The underlying protocol offers anonymity for users and prevents the server and user masquerade attacks. The security evaluation of our designed protocol endorses its robustness against well-known security threats. The performance analysis proves that our designed protocol outperforms other relevant protocols. The results show that our designed protocol is superior in security and efficiency.

1.2. Paper organization

The paper is organized as follows: Section 2 shows preliminaries along with the threat model and notations used in the paper. Section 3 gives a brief review of Qiu et al.'s scheme. Cryptanalysis of their scheme is shown in Section 4. Section 5 presents our introduced scheme. The security analysis of our scheme is shown in Section 6, and the comparison of the efficiency of our scheme with those of the related schemes is shown in Section 7. We concluded our article in Section 8.

2. Preliminaries

In this segment, we show some of the adversary's capabilities for the authentication scheme and important cryptographic primitives. Some of the notations we use in our paper are shown in Table 1.

2.1. Threat model

In this article, according to Refs. [31,32], we sum up the capabilities of adversary $\mathcal{A}$ as follows:

1. $\mathcal{A}$ has the capability to manage all the communication of the public channel and can resend, block, delete, modify, and interpret the message.
2. In polynomial time, all pairs of (ID_p, PW_p) can be listed by adversary $\mathcal{A}$ from $(\mathcal{S}_{PW}, \mathcal{S}_{ID})$, where $\mathcal{S}_{PW}$ denotes the password space and $\mathcal{S}_{ID}$ denotes the identity space, respectively.
3. Either extracting the smart card's parameters or intercepting the password via the malicious device can be performed by adversary $\mathcal{A}$ once at a time, but these tasks can not be performed at the same time.
4. Adversary $\mathcal{A}$ can encompass the user's password or the private key of the server while calculating the forward secrecy.

2.2. Elliptic Curve Cryptography (ECC)

An elliptic-curve $s^2 = r^3 + ar + b \ mod \ q$ can be recognized as a set of

Table 1
Notation guide.

Notations	Description
$\mathcal{U}_p$	User of the system
$\mathcal{S}_q$	TMIS Server
ID_p	Identity of $\mathcal{U}_p$
PW_p	Password of $\mathcal{U}_p$
r_p, a_p	Random numbers of $\mathcal{U}_p$
k_q	Secret key of $\mathcal{S}_q$
r_q, c_q	Random Numbers of $\mathcal{S}_q$
$\mathcal{SC}$	Smart card of $\mathcal{U}_p$
$E_q(a, b)$	An Elliptic Curve
P	Base Point of $E_q(a, b)$

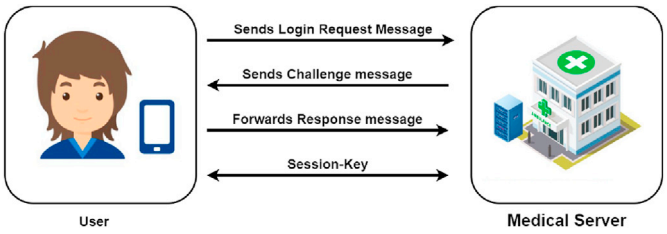

Fig. 2. Illustration of login and authentication environment.

restricted $E_q(a, b)$ keys [33]. Therefore, $(r,s)\ \varepsilon Z_q^*$, a, b are picked wisely to take in $4a^3 + 27b^2\ mod\ q \neq 0$ while q is a prodigious-prime and the length of q in bits refers to ≥160. The multiplication of scalar is gained with repeated addition, i.e. $nt = t_1 + t_2 + t_3 + \ldots\ \ldots + t(ntimes)$, a multiplier n and against a definite point t on $E_q(a, b)$. The constraints (a, b, q, t, n) should be a fragment of the countable field F_q. E is supposed to be an abelian-group, whereas O at immensity is chosen as the point of identity.

Definition 1. (Discrete Logarithm Problem aimed at Elliptic Curve (ECDLP)) In contrast to defining two random points X, $Y\ \varepsilon E_q(a, b)$, we calculate n, which is a scalar such that $X = nY$. The probability where a malicious user $\mathcal{A}$ can compute n through polynomial-time (t) is given as: $Adv_{\mathcal{A}}^{ECDLP}(t) = \mathcal{PRB}[\mathcal{A}(X,\ Y) = x : x\ \varepsilon Z_q]$. The ECDLP assumptions ascertain such that $Adv_{\mathcal{A}}^{ECDLP}(t) \leq \varepsilon$.

2.3. Hash function

The cryptographic collision-resistant hash function h: {1,0}* →

Patient $\mathcal{U}_p$	*Server* $\mathcal{S}_q$
Registration Phase:	
Enter User ID_p and PW_p	
Choose arbitrary number $r_p \in Z_P^*$	
Calculates $PR_p = h(ID_p \| PW_p \| r_p)$	
$\xrightarrow{\{ID_p, PR_p\}}$	
	Choose an integer $2^4 \leq n_o \leq 2^8$
	Calculates $M_p = h((h(ID_p) \oplus h(PR_p))\ \ mod\ \ n_o)$
	Choose arbitrary number $r_q \in Z_p^*$
	Computes $N = h(ID_p \| k_q \| r_q)$
	$O_p = N \oplus PR_p$
	Stores $\{ID_p, r_q\}$ in repository
	Stores $\{M_p, O_p, P, n_o, h()\}$ in a new $\mathcal{SC}$
$\xleftarrow{\{\mathcal{SC}\}}$	
Stores r_p *in* $\mathcal{SC}$	
Login and Authentication Phase:	
Insert SC and input ID_p and PW_p	
Then Calculates $PR_p = h(ID_p \| PW_p \| r_p)$	
$M_p = h((h(ID_p) \oplus h(PR_p)))\ mod\ n_o$	
Verifies $\acute{M}_p \stackrel{?}{=} M_p$	
Calculates $N = O_p \oplus PR_p$	
Calculates $a_p \in Z_p^*$	
Calculates $C_p = a_p P$	
$PID_p = N \oplus (ID_p \| C_p)$	
$X_p = h(ID_p \| N \| C_p)$	
$\xrightarrow{\{PID_p, X_p\}}$	
	Calculates $N = h(ID_p \| k_q \| r_q)$
	Using data in repository
	$(I\acute{D}_p \| \acute{C}_p) = PID_p \oplus N$
	Verifies $ID_p \stackrel{?}{=} I\acute{D}_p$
	Calculates $\acute{X}_p = h(I\acute{D}_p \| N \| \acute{C}_p)$
	Verifies $\acute{X}_p \stackrel{?}{=} X_p$
	Generates $c_q \in Z_p^*$
	Calculates $Y_q = c_q P$
	$SK = h(I\acute{D}_p \| N \| c_q \acute{C}_p \| \acute{C}_p \| Y_q)$
	$Z_q = h(SK \| Y_q \| N \| \acute{C}_p)$
$\xleftarrow{\{Y_q, Z_q\}}$	
$S\acute{K} = h(ID_p \| N \| a_p Y_q \| C_p \| Y_q)$	
$\acute{Z}_q = h(S\acute{K} \| Y_q \| N \| C_p)$	
Verifies $\acute{Z}_q \stackrel{?}{=} Z_q$	
if valid then S_q is authenticated	
Calculates $Q_p = h(S\acute{K} \| N \| C_p)$	
$\xrightarrow{\{Q_p\}}$	
	Calculates $\acute{Q}_p = h(SK \| N \| \acute{C}_p)$
	Verifies $\acute{Q}_p \stackrel{?}{=} Q_p$
	if valid then U_p is authenticated
$\longleftarrow\ \boxed{Sk = \acute{Sk}}\ \longrightarrow$	

Fig. 3. Registration, login and authentication phase of Qiu et al.'s scheme.

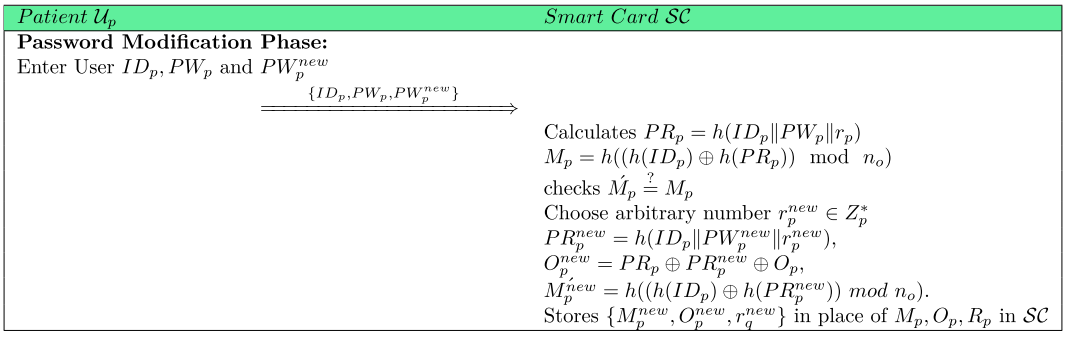

Fig. 4. Password modification of Qiu et al.'s scheme.

$\{1,0\}^{ln}$ works as a deterministic function that takes the variable length binary string $\lambda_0 \varepsilon \{1,0\}^*$ as input and produces the predetermined $ln-$bit output of the size $h(\lambda_0) \varepsilon \{1,0\}^{ln}$ [23,34]. Assume $Adv_{\mathcal{A}}^{hash-func}(\eta)$ as adversary $\mathcal{A}$'s advantage in having the collision:

$$Adv_{\mathcal{A}}^{hash-func}(\eta) = Prb\left[\left(\acute{\lambda_0}, \lambda_0\right) \leftarrow_{\mathcal{R}} \mathcal{A} : \acute{\lambda_0} \neq \lambda_0 \ and \ h(\acute{\lambda_0}) = h(\lambda_0)\right]$$

where $Prb[ET]$ presents the event ET's probability and $(\acute{\lambda_0}, \lambda_0) \leftarrow_{\mathcal{A}}$, and $\mathcal{A}$ is a pair $(\acute{\lambda_0}, \lambda)$ arbitrarily chosen by the probabilistic $\mathcal{A}$ $mathcalAdvnt$ of $\mathcal{A}$ collision-resistant. $h(.)$ can be determined with the execution time η over the random selection $(\acute{\lambda_0}, \lambda_0)$ if $Adv_{\mathcal{A}}^{hash-func}(\eta) \leq \upsilon h$ holds for a trivially small υ.

3. Review of Qiu et al.'s scheme

This section elaborates Qiu et al.'s [30] authentication protocol for the TMIS environment. Their protocol consists of three phases, including registration, authentication and password modification phases. An illustration of the login and authentication environment is given in Fig. 2. Registration, login and authentication phases of Qui et al.'s scheme are displayed by Fig. 3.

3.1. Registration phase

In the registration phase, any user $\mathcal{U}_p$ who interacts with the system for the first time has to register itself with the system. The registration phase consists of the following steps:

Step 1. Initially, $\mathcal{U}_p$ selects its unique password PW_p and identity ID_p. Thereafter, $\mathcal{U}_p$ chooses an arbitrary number $r_p \varepsilon Z_p^*$ and calculates $PR_p = h(ID_p \| PW_p \| r_p)$.

Step 2. $\mathcal{U}_P \Rightarrow \mathcal{S}_q : \{ID_p, PR_p\}$.

Step 3. On receiving the requested message $\{ID_p, PR_p\}$ for registration from $\mathcal{U}_p$, the server $\mathcal{S}_q$ chooses an arbitrary number $r_q \varepsilon Z_q^*$ and calculates: $M_p = h((h(ID_p) \oplus h(PR_p)) \ mod \ n_o)$, $N = h(ID_p \| k_q \| r_q)$ and $O_p = N \oplus PR_p$. Next, $\mathcal{S}_q$ stores $\{ID_p, r_q\}$ in its repository, where n_o is an integer within the range of $2^4 \leq n_o \leq 2^8$.

Step 4. $\mathcal{S}_q \Rightarrow \mathcal{U}_p : \mathcal{SC}\{M_p, O_p, P, n_o, h()\}$.

Step 5. In the end, $\mathcal{U}_p$ stores r_p in $\mathcal{SC}$.

3.2. Login phase

In this phase, $\mathcal{U}_p$ logs into the system. For this purpose, $\mathcal{U}_p$ has to insert his smart card $\mathcal{SC}$ into the card reader and enters its ID_p and PW_p.

Step 1. Whenever $\mathcal{U}_p$ inputs its ID_p and PW_p, $\mathcal{SC}$ firstly calculates $PR_p = h(ID_p \| PW_p \| r_p)$ and $M = h((h(ID_p) \oplus h(PR_p)) \ mod \ n_o)$. Next, $\mathcal{SC}$ validates whether $\acute{M}$ holds a true value by comparing the value of M stored inside the smart card $\mathcal{SC}$. If $\acute{M} = M$, then PW_p and ID_p are approved as valid. Else, the session is aborted. After that, $\mathcal{SC}$ continues calculating: $N = O_p \oplus PR_p$ and selects an arbitrary number $a_p \varepsilon Z_p^*$. $\mathcal{SC}$ further calculates: $C_p = a_p P$, $PID_p = N \oplus (ID_p \| C_p)$ and $X_p = h(ID_p \| N \| C_p)$.

Step 2. $\mathcal{U}_p \rightarrow \mathcal{S}_q : \{PID_p, X_p\}$.

3.3. Authentication phase

In this phase, $\mathcal{S}_q$ and $\mathcal{U}_p$ will authenticate each other. If they mutually authenticate each other, then the session key will be exchanged between them for further communication.

Step 1. On receiving the login request message $\{PID_p, X_p\}$ from $\mathcal{U}_p$, $\mathcal{S}_q$ computes $N = h(ID_p \| K_q \| r_q)$ by using the repository and his secrete key. $\mathcal{S}_q$ then calculates $\{\acute{ID_p} \| \acute{C_p}\} = PID_p \oplus N$ and verifies $\acute{ID_p} \stackrel{?}{=} ID_p$ by searching in its repository. If it holds, $\mathcal{S}_q$ inspects that the password PW_p entered by $\mathcal{U}_p$ is wrong. Otherwise, $\mathcal{S}_q$ ends the session. Once the time for entering the wrong password exceeds the threshold value (such as 5), the server perceives it as adversary $\mathcal{A}$ that has the stolen smart card $\mathcal{SC}$. If the timer exceeds the threshold value, $\mathcal{S}_q$ will latch the smart card $\mathcal{SC}$ until $\mathcal{U}_p$ sends a request for re-regeneration/revocation. Otherwise, $\mathcal{S}_q$ calculates $\acute{X_p} = h(\acute{ID_p} \| N \| \acute{C_p})$ and checks $\acute{X_p} \stackrel{?}{=} X_p$. In the case of inequality, $\mathcal{S}_q$ terminates the session and calculates a number $N_p = 1$. Further, $\mathcal{S}_q$ rejects the smart card $\mathcal{SC}$ until $\mathcal{U}_P$ re-registers in such a condition when N_p increases from some threshold values (such as 5). Otherwise, $\mathcal{S}_q$ generates an arbitrary number c_q and calculates $Y_q = c_q P$, $SK = h(\acute{ID_p} \| N \| c_q \acute{C_p} \| C_p \| Y_q)$ and $Z_q = h(SK \| Y_q \| N \| \acute{C_p})$.

Step 2. $\mathcal{S}_q \rightarrow \mathcal{U}_p : \{Y_q, Z_q\}$.

Step 3. On receiving the challenge message $\{Y_q, Z_q\}$ from $\mathcal{S}_q$, $\mathcal{U}_p$ calculates $\acute{SK} = h(ID_p \| N \| a_p Y_q \| C_p \| Y_q)$ and $\acute{Z_q} = h(SK \| Y_q \| N \| C_p)$. After that, $\mathcal{U}_p$ checks whether $\acute{Z_q} \stackrel{?}{=} Z_q$. The session will be aborted if the condition is not equal. Otherwise, $\mathcal{S}_q$ is validated by $\mathcal{U}_P$ and $\mathcal{U}_p$ accepts $\acute{SK}$. Further, $\mathcal{U}_p$ calculates $Q_p = h(SK \| N \| C_p)$.

Step 4. $\mathcal{U}_p \rightarrow \mathcal{S}_q : \{Q_p\}$

Step 5. Whenever $\mathcal{S}_q$ receives the response message $\{Q_p\}$, $\mathcal{S}_q$ calculates: $Q_p = h(SK \| N \| \acute{C_p})$ and verifies whether $\acute{Q_p} \stackrel{?}{=} Q_p$. If verified, then $\mathcal{U}_p$ has been authenticated.

Step 6. Finally, the common session key $SK = \acute{SK}$ is shared between $\mathcal{S}_q$ and $\mathcal{U}_p$ for secure communication of the particular session.

3.4. Password modification phase

This phase enables $\mathcal{U}_p$ to modify his password. The password modification phase is illustrated in Fig. 4. The following is a step-by-step process that $\mathcal{U}_p$ has to follow:

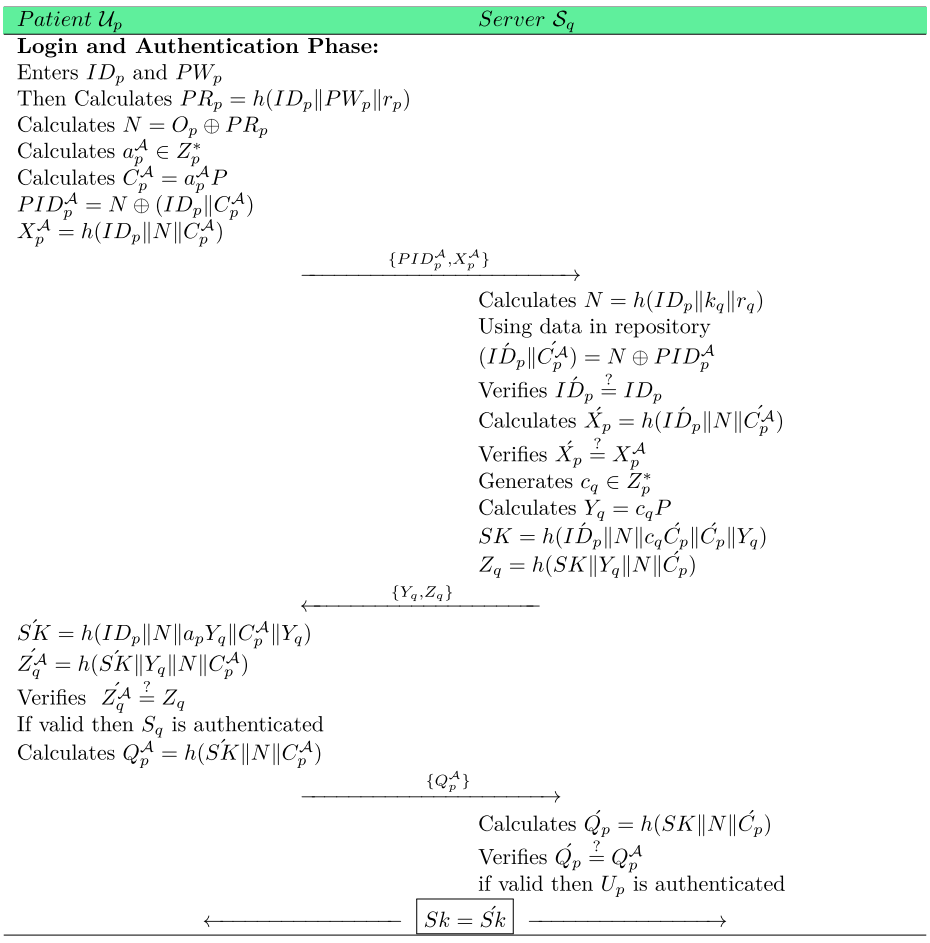

Fig. 5. User masquerade attack on Qiu et al.'s protocol.

Step 1. First, $\mathcal{U}_p$ inserts his smart card $\mathcal{SC}$. After that, $\mathcal{U}_p$ enters ID_p, PW_p and a new password PW_p^{new}. Next, $\mathcal{SC}$ computes $PR_p = h(ID_p \| PW_p \| r_p)$ and $\acute{M}_p = h((h(ID_p) \oplus h(PR_p)) \bmod n_o)$. Afterwards, $\mathcal{SC}$ checks whether $\acute{M}_p$ is equal to M_p. $\mathcal{SC}$ rejects $\mathcal{U}_p$ to renew the password if the condition is different.

Step 2. Otherwise, $\mathcal{SC}$ produces an arbitrary number r_p^{new} and calculates: $PR_p^{new} = h(ID_p \| PW_p^{new} \| r_p^{new})$, $O_p^{new} = PR_p \oplus PR_p^{new} \oplus O_p$ and $M_p^{\acute{n}ew} = h((h(ID_p) \oplus h(PR_p^{new})) \bmod n_o)$.

Step 3. Finally, $\mathcal{SC}$ stores $\{M_p^{new}, r_p^{new}, O_p^{new}\}$ in place of $\{M_p, r_p, O_p\}$.

4. Cryptanalysis of the former scheme

This section exposes the compelling inadequacy in Qiu et al.'s [30] scheme. We observe that their protocol is incorrect in the user's anonymity. Moreover, it is susceptible to offline password guess, privileged insider, server and user impersonation and reply attacks. The above-mentioned attacks reflect the assumptions that the channel through which the communication takes place across $\mathcal{S}_q$ and $\mathcal{U}_p$ in the login and authentication phase is accessible to adversary $\mathcal{A}$. Thus, in such a case, $\mathcal{A}$ can intercept the message transferred from either side over the public channel.

4.1. Incorrectness

Qiu et al. presented a novel notion of perfect anonymity in their protocol, in which the server $\mathcal{S}_q$ remains unable to identify the patient $\mathcal{U}_p$'s identity ID_p requested for login. Such a notion of perfect anonymity, in our opinion, is erroneous. It is worth noting that a patient $\mathcal{U}_p$ initiates a login request in the form of $\{PID_p, X_p\}$ and the server gets it and tries to determine $N = h(ID_p, K_q, r_q)$. The problem here is that the identity is not known to $\mathcal{S}_q$ at the moment when he is getting $\{PID_p, X_p\}$. Then how can he use ID_p to determine N of that particular user? Therefore, $\mathcal{S}_q$ must decide PID_p first to determine ID_p of that particular user. Only in that condition can he do the subsequent calculation.

4.2. Offline password guess attack

In this attack, a malicious user $\mathcal{A}$ steals a legitimate user $\mathcal{U}_p$'s smart card $\mathcal{SC}$ and takes out the useful parameters from it. Then, $\mathcal{A}$ attempts to make a prediction about $\mathcal{U}_p$'s true identity and password. This subsection shows that Qiu at el.'s scheme [30] fails to resist the offline password guessing attack in the following two cases:

- **Case 1: Verification using the value inside the smart card**

Initially, $\mathcal{A}$ takes out $\{M, r_p, n_o\}$ saved in the smart card $\mathcal{SC}$. Afterwards, $\mathcal{A}$ performs the following steps to estimate $\mathcal{U}_p$'s identity and password:

Step 1. Firstly, $\mathcal{A}$ estimates $PW^*{}_p$ and $ID^*{}_p$ from the password dictionary space D_{pw} and then identify dictionary space D_{ID}, respectively.

Step 2. $\mathcal{A}$ calculates $PR_p^* = h(ID^*{}_p \| PW^*{}_p \| r_p)$ and $M_p = h((ID^*{}_p) \oplus h(PR^*{}_p) \bmod n_o)$.

Step 3. $\mathcal{A}$ checks if $\acute{M}_p = M_p$, $\mathcal{A}$ then detects the accurate password and

Fig. 6. Proposed scheme.

$\mathcal{U}_p$'s identity. Until $\mathcal{A}$ finds these parameters, $\mathcal{A}$ will keep on repeating these steps (1) (2) (3).

Step 1. $\mathcal{A}$ first guesses $PW^*{}_p$, $ID^*{}_p$ from the password vocabulary space D_{pw} and the vocabulary space of identity D_{ID}, respectively.

Step 2. $\mathcal{A}$ calculates $PR^* = h(ID_p^* \| PW_p^* \| r_p)$.

Step 3. $\mathcal{A}$ calculates $\acute{N} = O_p \oplus PR_p^*$, where O_p is stolen from the smart card $\mathcal{SC}$. $\mathcal{A}$ can then determine $(ID_p \| \acute{C}_p) = N \oplus PID_p$ and. $X_p^* = h(\{ID_p \| N \| X_p\})$.

Step 4. $\mathcal{A}$ checks whether X_p^* is equal to X_p in the request message and then finds the correct password and identity. Else, until $\mathcal{A}$ finds these parameters, $\mathcal{A}$ will keep on repeating Steps 1-4.

- **Case 2: Verification using the value in the open channel**

4.3. Privileged insider and user impersonation attacks

Assuming that an insider can access the information of registration $\{ID_p, PR_p\}$ of a valid user $\mathcal{U}_p$ and becomes an adversary $\mathcal{A}$ by determining $\acute{N} = O_p \oplus PR_p$ and $(I\acute{D}_p \| \acute{C}_p) = N \oplus PID_p$, if T, ID_p and $\acute{C}_p$ can be

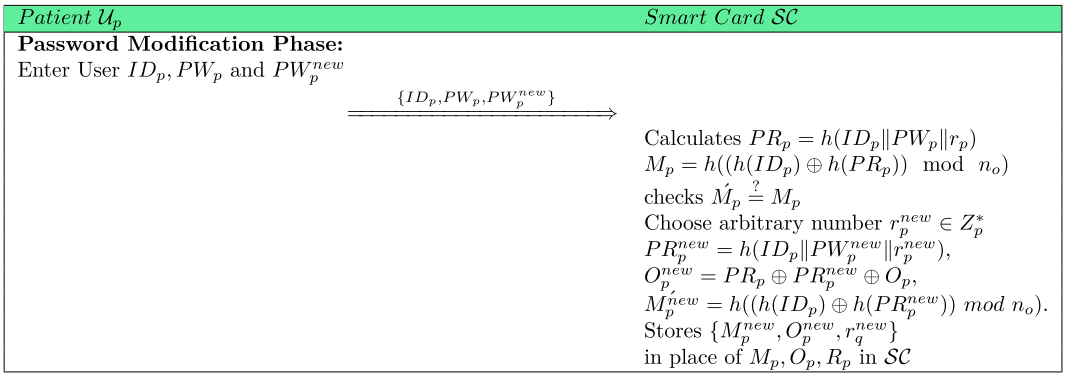

Fig. 7. Password modification of proposed scheme.

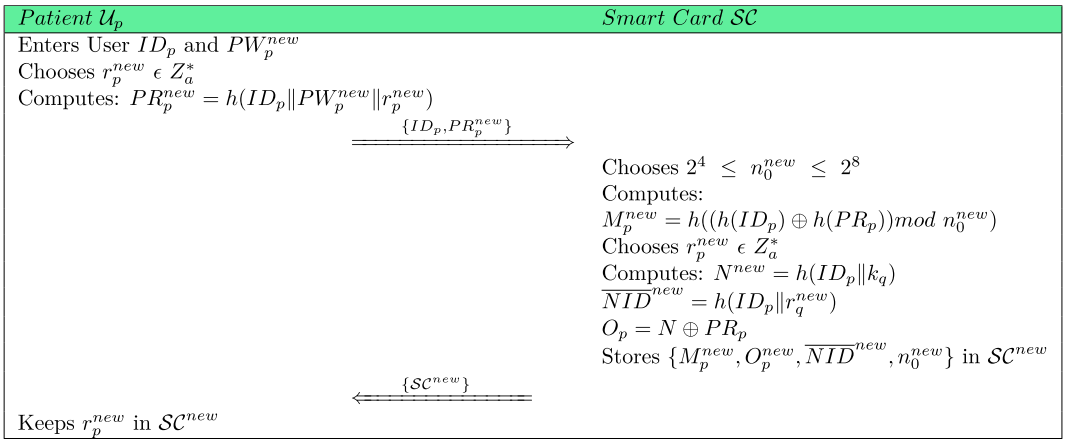

Fig. 8. User re-registration/Revocation phase.

determined, then $\mathcal{A}$ can easily implement the user impersonation by determining $PID_p = N \oplus (ID_p \| C_p)$, $X_p = h(ID_p \| N \| C_p)$ and send $\{PID_p, X_p\}$ on behalf of the legal user $\mathcal{U}_p$. The user impersonation $\mathcal{U}_p$ attack is further illustrated in Fig. 5.

5. Proposed scheme

In order to strengthen the security of the previously presented protocol by Qui et al. and to enhance the efficiency of former protocols, we devise an enhanced version of the protocol based on ECC. Moreover, we introduce encryption and decryption to make our scheme robust against known attacks. Furthermore, we present an additional phase, namely, a user re-registration/revocation phase, that allows the user to re-register itself. The details of our designed protocol, as shown in Fig. 6, is given below:

5.1. Registration phase

In the registration phase, any user who interacts with the system for the first time has to register with the system. The registration phase comprises the following steps:

Step 1. Initially, $\mathcal{U}_p$ selects a unique password PW_p, an identity ID_p and an arbitrary number $r_P \varepsilon Z_p^*$. Thereafter, $\mathcal{U}_p$ calculates: $PR_p = h(ID_p \| PW_p \| r_p)$.

Step 2. $\mathcal{U}_p \Rightarrow: \{ID_p, PR_p\}$.

Step 3. On receiving the registration message $\{ID_p, PR_p\}$ from $\mathcal{U}_p$, $\mathcal{S}_q$ chooses an arbitrary number $r_q \varepsilon Z_q^*$ and computes: $M_p = h((h(ID_p) \oplus h(PR_p)) \ mod \ n_o)$, $N = h(ID_p \| k_q)$, $N \ ID = Enc_{k_q}(N \| r_q)$, $O_p = N \oplus PR_p$, where n_o is an integer chosen within the range of $2^4 \leq n_o \leq 2^8$.

Step 4. $\mathcal{S}_q \Rightarrow \mathcal{U}_p : \mathcal{SC}\{M_p, O_p, P, N \ ID, n_0, h()\}$.

Step 5. In the end, $\mathcal{U}_p$ stores R_p in the smart card.

5.2. Login phase

In this phase, $\mathcal{U}_p$ logs into the system. For this purpose, $\mathcal{U}_p$ insert his $\mathcal{SC}$ into the card reader and enters his ID_p and PW_p.

Step 1. Whenever $\mathcal{U}_p$ inputs his ID_p and PW_p, $\mathcal{SC}$ firstly calculates: $PR_p = h(ID_p \| PW_i \| r_p)$ and $M = h((h(ID_p) \oplus h(PR_p)) \ mod \ n_o)$. Thereafter, $\mathcal{SC}$ validates the correctness of $\acute{M}$ by comparing the value of M stored within $\mathcal{SC}$. If $\acute{M} = M$ succeeds, PW_i, ID_p is approved as valid. Otherwise, the session is aborted. Next, $\mathcal{SC}$ continues calculating: $N = O_p \oplus PR_p$ and selects an arbitrary number $a_p \varepsilon Z_p^*$. Further, $\mathcal{SC}$ calculates: $C_p = a_p P$, $PID_p = N \oplus \{ID_p \| C_p\}$ and $X_p = h(ID_p \| N \| C_p)$.

Step 2. $\mathcal{U}_p \Rightarrow \mathcal{S}_q : \{N \ ID, PID_p, X_p\}$.

5.3. Authentication phase

In this phase, $\mathcal{S}_q$ and $\mathcal{U}_p$ will authenticate each other. If they mutually authenticate each other, then the session key will be exchanged between them for further secure communication.

Step 1. After obtaining the request message $\{N \ ID, PID_p, X_p\}$, $\mathcal{S}_q$ decrypts $(N \| r_q) = Dec_{k_q}(N \ ID)$ using its private key k_q. After that, $\mathcal{S}_q$

computes: $(I\acute{D}_p \| \acute{C}_p) = PID_p \oplus N$. Further, $\mathcal{S}_q$ calculates: $\acute{X}_p = h(I\acute{D}_p \| N \| \acute{C}_p)$ and checks $\acute{X}_p \stackrel{?}{=} X_p$. If they are not equal, $\mathcal{S}_q$ inspects that the password PW_p recently entered by $\mathcal{U}_p$ is wrong. Once the time for entering the incorrect password(s) exceeds the threshold value, the server perceives that adversary $\mathcal{A}$ has stolen the smart card $\mathcal{SC}$ and is trying to initiate a session. If the time exceeds the threshold value, $\mathcal{S}_q$ will latch SC unless $\mathcal{U}_p$ sends a request for re-regeneration. In the case of unauthorization, $\mathcal{S}_q$ terminates the session and calculates a number $N_p = 1$. $\mathcal{S}_q$ rejects the smart card $\mathcal{SC}$ until the user $\mathcal{U}_p$ re-registers in such a condition if N_p increases from some threshold values. Otherwise, $\mathcal{S}_q$ generates an arbitrary number c_q and calculates $Y_q = c_q P$, $SK = h(I\acute{D}_p \| N \| c_q \acute{C}_p \| C_p \| Y_q)$, $Z_q = h(SK \| Y_q \| N \| \acute{C}_p)$ and $R_q = Enc_{k_q}(N \| r_q^{new}) \oplus \acute{C}_p$.

Step 2. $\mathcal{S}_q \Rightarrow \mathcal{U}_p : \{Y_q, Z_q, R_q\}$.

Step 3. On receiving the challenge message $\{Y_q, Z_q, R_q\}$, $\mathcal{U}_p$ calculates: $\acute{SK} = h(I\acute{D}_p \| N \| a_p Y_q \| C_p \| Y_q)$ and $\acute{Z}_q = h(\acute{SK} \| Y_q \| N \| C_p)$. Afterwards, $\mathcal{U}_p$ checks whether $\acute{Z}_q \stackrel{?}{=} Z_q$. If it does not hold, the session will be aborted. Otherwise, $\mathcal{S}_q$ is authenticated, and $\mathcal{U}_p$ updates the value of $N\ ID$ to $N\ ID^{new}$ in its $\mathcal{SC}$.

Step 4. Finally, both $\mathcal{U}_p$ and $\mathcal{S}_q$ agree on a shared common session key $\acute{SK} = SK$ for secure communication in a particular session.

5.4. Password modification phase

This phase enables $\mathcal{U}_p$ to modify his password according to his own will. (see Fig. 7). The step-by-step process of the password modification phase is given in the subsequent steps while its illustration is given in Fig. 8:

Step 1. Firstly, $\mathcal{U}_p$ inserts his $\mathcal{SC}$ inside the card reader. $\mathcal{U}_p$ then enters his ID_p, PW_p and a new password PW_p^{new}. Thereafter, $\mathcal{SC}$ calculates: $PR_p = h(ID_p \| PW_p \| r_p)$ and computes: $\acute{M}_p = h((h(ID_p) \oplus h(PR_p))\ mod\ n_o)$. Subsequently, $\mathcal{SC}$ verifies whether $\acute{M}_p$ is equal to M_p. If this condition is not satisfied, any change of $\mathcal{U}_p$'s password will be rejected by $\mathcal{SC}$.

Step 2. Otherwise, the smart card $\mathcal{SC}$ selects a nonce r_p^{new} and then computes $PR_p^{new} = h(ID_p \| PW_p^{new} \| r_p^{new})$, $O_p^{new} = PR_p \oplus PR_p^{new} \oplus O_p$ and $M_p^{\acute{n}ew} = h((h(ID_p) \oplus h(PR_p^{new}))\ mod\ n_o)$.

Step 3. Finally, the smart card $\mathcal{SC}$ stores $M_p^{new}, r_p^{new}, O_p^{new}$ in place of M_p, r_p, O_p.

5.5. User re-registration/revocation phase

Through this phase, a registered user $\mathcal{U}_p$ can recover his stolen/lost smart card $\mathcal{SC}$. The details of this phase are given in the subsequent steps:

Step 1. Initially, $\mathcal{U}_p$ inputs its previous identity ID_p and new password PW_p^{new}. Thereafter, $\mathcal{U}_p$ chooses a new arbitrary number $r_p^{new}\ \varepsilon\ Z_a^*$ and computes: $PR_p^{new} = h(ID_p \| PW_p^{new} \| r_p^{new})$. Afterwards, $\mathcal{U}_p$ forwards the re-registration/revocation request to $\mathcal{S}_q$ via a secure medium.

Step 2. On getting the re-registration/revocation request from $\mathcal{U}_p$, $\mathcal{S}_q$ selects $2^4 \le n_0^{new} \le 2^8$ and computes: $M_p^{new} = h((h(ID_p) \oplus h(PR_p))mod\ n_0^{new})$. Next, $\mathcal{S}_q$ chooses $r_p^{new}\ \varepsilon\ Z_a^*$ and computes: $N^{new} = h(ID_p \| k_q)$, $N\ ID^{new} = h(ID_p \| r_q^{new})$ and $O_p = N \oplus PR_p$. After that, $\mathcal{S}_q$ stores $\{M_p^{new}, O_p^{new}, N\ ID^{new}, n_0^{new}\}$ in a new smart card $\mathcal{SC}^{new}$. Finally, $\mathcal{S}_q$ issues $\mathcal{SC}^{new}$ to $\mathcal{U}_p$.

Step 3. On getting $\mathcal{SC}^{new}$, $\mathcal{U}_p$ keeps r_p^{new} in $\mathcal{SC}^{new}$.

6. Security analysis

This section presents a formal security evaluation of our designed protocol as well as a discussion on its resilience against various attacks. The subsequent subsections give the details:

6.1. Informal security

This section discusses the resilience of our protocol against well-known attacks.

6.1.1. Anonymity and privacy

In our devised protocol, $\mathcal{U}_p$'s ID_p is not sent in a clear text. Instead, PID_p is computed using $PID_p = N \oplus (ID \| C_p)$. Whenever PID_p arrives on the server's end, the server $\mathcal{S}_q$ firstly decrypts $N\ ID$ to obtain N from $(N \| r_q) = Dec_{k_q}(N\ ID)$. As it requires $\mathcal{S}_q$'s private key for decryption, only the legitimate server $\mathcal{S}_q$ can decrypt $N\ ID$ to determine N. Furthermore, $\mathcal{S}_q$ computes ID_p from $(I\acute{D}_p \| \acute{C}_p) = N \oplus PID_p$. Moreover, $C_p = a_p P$ has random variables which are specific in each session and resists $\mathcal{A}$ to anticipate whether the two distinct sessions have or have not been commenced by the same user. In this way, our devised protocol offers anonymity and privacy for the user.

6.1.2. Mutual authentication

Whenever $\mathcal{U}_p$ sends a request message $\{N\ ID, PID_p, X_p\}$ to $\mathcal{S}_q$, $\mathcal{S}_q$ verifies $\mathcal{U}_p$ by confirming $\acute{X}_p \stackrel{?}{=} X_p$. In the same way, upon getting a challenge message $\{Y_q, Z_q, R_q\}$ from $\mathcal{S}_q$, $\mathcal{U}_p$ authenticates $\mathcal{S}_q$ by verifying $\acute{Z}_p \stackrel{?}{=} Z_p$. Since $\mathcal{S}_q$ authenticates $\mathcal{U}_p$ and $\mathcal{U}_p$ authenticates $\mathcal{S}_q$, our devised protocol offers mutual authentication.

6.1.3. User impersonation attack

In our devised protocol, $\mathcal{A}$ can not generate a valid login message $\{N\ ID, PID_p, X_p\}$ because the computation of this message requires $\mathcal{U}_p$'s identity and password, which are only known to the legitimate $\mathcal{U}_p$. Another condition can be considered where $\mathcal{A}$ steals $\mathcal{U}_p$'s smart card $\mathcal{SC}$ and extracts datum $\{M_p, O_p, N\ ID, n_0, r_p\}$ to generate $\{N\ ID, PID_p, X_p\}$. Even in that case, $\mathcal{A}$ will remain unable to impersonate $\mathcal{U}_p$ because there is no hint for $\mathcal{A}$ through which he can recognize ID_p and PW_p of $\mathcal{U}_p$. Since ID_p and PW_p are not available to $\mathcal{A}$, our devised protocol is resilient against the $\mathcal{U}_p$ impersonation attack.

6.1.4. Server impersonation attack

In order to breach the security of $\mathcal{S}_q$, $\mathcal{A}$ must have the knowledge of $\mathcal{S}_q$'s secret key k_q to produce a real response message $\{Y_q, Z_q, R_q\}$, where $Y_q = c_q.P$, $Z_q = h(SK \| Y_q \| N \| \acute{C}_p)$ and $R_q = Enc_{k_q}(N \| r_q^{new}) \oplus \acute{C}_p$, since $\mathcal{A}$ needs to know k_q for computing $\{Y_q, Z_q, R_q\}$, which is only known to $\mathcal{S}_q$. Thus, our devised protocol is resilient to the server impersonation attack.

6.1.5. Replay attack

Both the user and the server generate their own session-specific random numbers a_p and c_q. If $\mathcal{A}$ intercepts the request message, he can not replay it later on, because both the request and challenge messages $\{N\ ID, PID_p, X_p\}$ and $\{Y_q, Z_q, R_q\}$ involve a_p and c_q. Thus, due to the dynamic nature of messages generated in each session, it is not feasible for $\mathcal{A}$ to mount a replay attack.

6.1.6. Perfect forward secrecy

The U_p and S_q compute the session key enclosed by C_p and Y_q from either side, respectively. Since the computation of the section key is secured under the hard problem of ECC, even if the long term private key to any participant is brought out by $\mathcal{A}$, the preceding session keys will not be easily derived by $\mathcal{A}$ in the polynomial time. Hence, perfect forward secrecy is offered by our proposed protocol.

Table 2
Execution time of cryptographic operations.

Operation	Execution Time (ms)
T_{th}	0.00097
T_{tpm}	0.0035
T_{Enc}/T_{Dec}	0.109/0.0036
T_{pa}	0.0028

Table 3
System specifications.

Item	Specifications
System	DELL
Generation	Core i7
RAM	16 GB
Processor	3.60 GH_Z
Language	Python
OS	Linux Ubunto
Library	PyCrypto

6.1.7. Insider and stolen verifier attack

Our introduced scheme does not hold any verifier table. Similarly, there is no database maintained by the server. Moreover, the $\mathcal{U}_p$ does not send the password in a plain text. Thus, the insider attack and the stolen verifier attack are not feasible in our introduced protocol.

6.1.8. Efficient smart card revocation

Unlike Qiu et al.'s protocol, a registered $\mathcal{U}_p$ can recover his stolen/lost $\mathcal{SC}$. For this purpose, $\mathcal{U}_p$ needs to forward a request message to $\mathcal{S}_q$ and authenticate his registration parameters (e.g., identity and password). After that, $\mathcal{S}_q$ will issue a new smart card to $\mathcal{U}_p$. It is worth noting that in our designed protocol, $\mathcal{S}_q$ will deactivate $\mathcal{U}_p$'s old $\mathcal{SC}$ on his request after verifying the registration parameters (e.g., identity and password). $\mathcal{S}_q$ will also deactivate $\mathcal{SC}$ of $\mathcal{U}_o$ if the time for entering the wrong password exceeds the threshold value (such as 5) when it perceives that $\mathcal{A}$ has stolen $\mathcal{SC}$ of $\mathcal{U}_p$. Hence, smart card revocation is efficient in our designed protocol.

6.1.9. Smart card stolen attack

If $\mathcal{A}$ happens to steal $\mathcal{U}_p$'s smart card and recovers datum $\{M_p, O_p, P, N\ ID, n_o, h()\}$ stored in it, $\mathcal{A}$ may try to generate a valid login message $N\ ID, PID_p, X_p$ to deceive $\mathcal{S}_q$. However, any attempt to compute the valid login message will fail because the identity and password of $\mathcal{U}_p$ are required for computing the request message. Since identity and password are unavailable to $\mathcal{A}$ and there is not any hint to which $\mathcal{A}$ can get access, our protocol has the potential to resist the stolen smart card attack.

6.1.10. No clock synchronization

Timestamps are not used in the proposed scheme. Instead, random numbers from both U_p and S_q for each session are generated. So, no clock synchronization is required.

6.2. Formal security analysis

We adopt the well known Random Oracle Model (ROM) mentioned in Ref. [35] to formally validate our scheme. For this purpose, the following oracle is characterized:

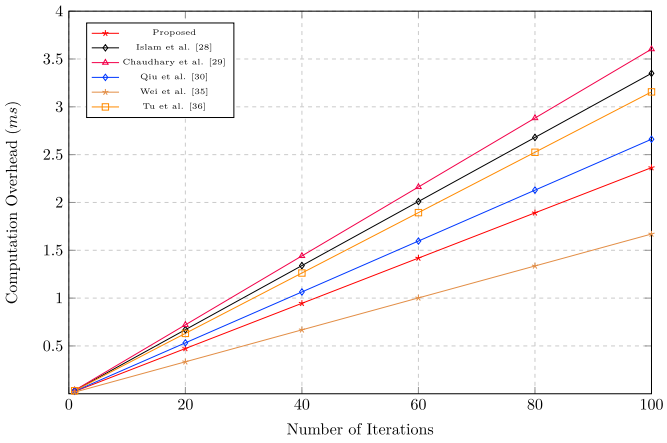

Fig. 9. Analysis of computational cost between proposed and related protocols.

Table 4
Performance comparison.

Scheme:	Total Computation Cost (ms)	Communication Cost (bits)	Storage Cost (bits)
Proposed	$10T_h + 4T_{pm} + 2T_{Enc} + 1T_{Dec} + 5T_{\oplus} + 33T_{\|} \approx 0.02364$	2240	1216
Islam et al. [28]	$10T_h + 6T_{pm} + 1T_{pa} + 2T_{\oplus} + 28T_{\|} \approx 0.0335$	3040	1376
Chaudhry et al. [29]	$9T_h + 7T_{pm} + 1T_{pa} + 1T_{mi} + 11T_{\oplus} + 17T_{\|} \approx 0.03603$	2240	1440
Qiu et al. [30]	$13T_h + 4T_{pm} + 5T_{\oplus} + 32T_{\|} \approx 0.02661$	1664	1568
Wei et al. [35]	$10T_h + 2T_{me} + 1T_{mi} + 0T_{\oplus} + 15T_{\|} \approx 0.0167$	2755	1152
Tu et al. [36]	$8T_h + 6T_{pm} + 1T_{pa} + 0T_{\oplus} + 24T_{\|} \approx 0.03156$	1920	416

Table 5
Comparison of security features.

Scheme:	Proposed	Islam et al. [28]	Chaudhry et al. [29]	Qiu et al. [30]	Wei et al. [35]	Tu et al. [36]
Offer anonymity and privacy	✓	✓	✓	✓	×	–
Prevents privilege insider	✓	✓	✓	✓	✓	✓
Prevents offline password-guessattack	✓	×	×	×	×	✓
Prevents user masquerade attack	✓	×	×	×	×	✓
Prevents server masquerade attack	✓	×	×	✓	×	–
Prevents replay attack	✓	✓	✓	✓	×	✓
Prevents man-in-middle attack	✓	×	×	✓	×	–
Offers mutual authentication	✓	×	✓	×	×	✓
Offers perfect-forward secrecy	✓	×	✓	✓	×	✓
Offers no clock synchronization	✓	×	×	×	×	–
Efficient Smart Card Revocation	✓	×	×	×	×	×
Smart Card Stolen Attack	✓	×	×	×	×	×

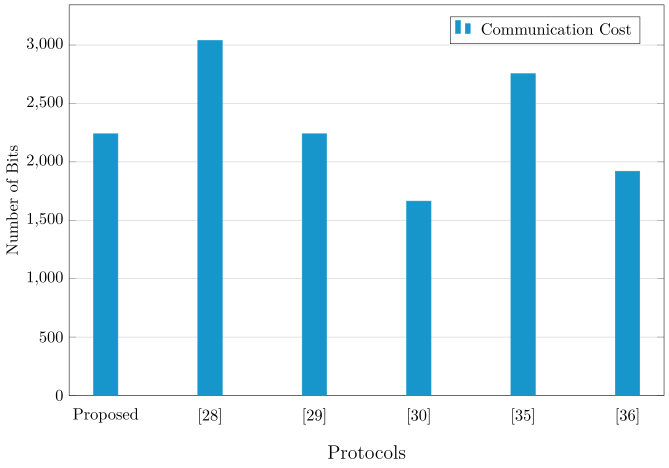

Fig. 10. Analysis of communication cost between proposed and related protocols.

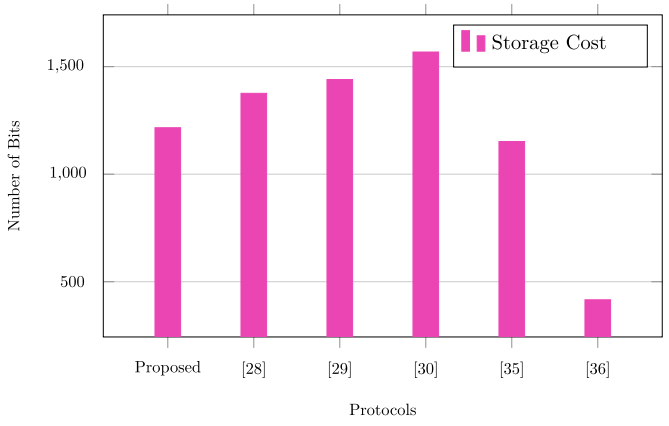

Fig. 11. Analysis of storage cost between proposed and related protocols.

Algorithm 1. $EXP1^{HASH,ECDLP}_{PRU,AAS}$

1: Intrude the login communication $\{N\ ID,PID_p,X_p\}$, Where $X_p = h(ID_p\|N\|C_p)$, $PID_p = N \oplus (ID_p\|C_p)$
2: Request Reveal oracle on X_p and obtain $h(ID_p'\|N\|C_p') \leftarrow Reveal(X_p)$
3: Request Reveal oracle on $(ID_p\|N)$ and obtain $(ID_p'\|N) \leftarrow Reveal(ID_p'\|N)$
4: Compute $(ID_p'\|C_p') = N \oplus PID_p$
5: **if** $(X_p = X_p')$ **then**
6: Take ID_p'
7: Request Extract oracle on PID_p' to obtain $(ID_p'\|C_p') \leftarrow Extract(ID_p'\|C_p')$
8: Request Reveal oracle on and obtain $(ID_p'\|N) \leftarrow Reveal(ID_p'\|N)$
9: Eavesdrop the challenge message $\{Y_q, Z_q, R_q\}$, Where
$Y_q = c_qP,\ Z_q = h(SK\|Y_q\|N\|C_p'), R_q = Enc_{k_q}(N\|r_q^{new}) \oplus C_p'$
10: Compute $SK' = h(ID_p\|N\|a_pY_q\|C_p\|Y_q)$
11: Compute $Z_q' = h(SK'\|Y_q\|N\|C_p')$
12: **if** $(Z_q' \stackrel{?}{=} Z_q)$ **then**
13: Accept SK
14: Compute session key $SK = h(ID_p\|N\|a_pY_q\|C_p\|Y_q)$
15: **else**
16: **return** Fail
17: **end if**
18: **else**
19: **return** Fail
20: **end if**

Theorem. *The proposed protocol is secure against adversary $\mathcal{A}$ for the verdict that $\mathcal{U}_p$'s identity ID_p, k_q and SK are computed between U_p and S_q under the firmness supposition of the elliptic curve discrete logarithm problem and the protected one way hash function.*

Proof. *consider a malicious user as an adversary $\mathcal{A}$ with the competence to extract U_p's ID_p, S_q's secrete key k_q and the session key SK. $\mathcal{A}$ simulates the algorithmic experiment $EXP^{\mathcal{HASH},\mathcal{ECDLP}}_{\mathcal{PRU\ AS},\mathcal{A}}$ over the design protocol $\mathcal{PRUAS}$ by executing both Reveal and Extract oracle's. Here we can denote the probability of success for the aforementioned experiment as: $succe_1 = |Prb[EXP^{\mathcal{HASH},\mathcal{ECDLP}}_{\mathcal{PRU\ AS},\mathcal{A}} = 1] - 1|$. Subsequently, the advantage of $\mathcal{A}$ can be characterized as: $Advt^{ECDLP,HASH}_{TFBAMS,\mathcal{A}}(q_{ex},t_e,q_{rv}) = max\__A(succe_1)$, where q_{ex} can be made by $\mathcal{A}$ and the reveal queries q_{rv}. SK, k_q and ID_p can be computed by $\mathcal{A}$ according to the trail if he can:*

- *Interrupt the ECDLP*
- *Reverse the protected hash function*

Likewise, breaking the ECDLP is not computably feasible by the definition of point 1. *Therefore, according to the definition of point* 2, *reversing the protected hash is not possible as well. Hence, $Advt^{ECDLP,HASH}_{TFBAMS,\mathcal{A}}(q_{ex}, t_e, q_{rv}) = max\__A(succe_1)\varepsilon$. Consequently, our devised protocol is protected against $\mathcal{A}$'s computation of U_p's ID_p, S_q's secrete key k_q and the session key SK.*

- **Extract** The scalar multiply k is yielded by the oracle categorically out of the given elliptic curve point P and $KP = O$.
- **Reveal** The oracle categorically yields a string S from the hash function that is one way $R = h(S)$.

7. Security and performance comparisons

This section compares the security features and performance of the designed protocol and those of the related protocols, such as schemes developed Qiu et al. [30], Wei et al. [35], Chaudhry et al. [29], Islam et al. [28] and Tu et al. [36]. We ignore the string concatenation and exclusive-OR operation when comparing the computational complexity of our protocol with those of the related protocols because their computational cost is trivial. The operations considered during this comparison are described below:

T_{pa}: The estimated execution time required for elliptic curve point addition.
T_{pm}: The estimated execution time required for point multiplication.
T_{me}: The execution time for modeler exponentiation.
T_{mi}: The execution time of modeler inversion.
T_{th}: The execution time of one way hash function.
$T_{E_{nc}/D_{ec}}$:The execution time of encryption/decryption.
$T_{\oplus}$: The time of Xor operation.
$T_{\|}$: The execution time of concatenation.

We utilize the experimental results of T_{th}, T_{pm}, T_{Enc}, T_{Dec} and T_{pa}, which are 0.00097 ms, 0.0035 ms, 0.109 ms, 0.0036 ms and 0.0028 ms, respectively. The cryptographic operations and their execution time are illustrated in Table 2. These results are taken after implementing the concerned protocols over a specific system with specifications as follows: Core i7 processor with clock speed 3.60 GHz, RAM 16 GB and *Linux Ubunto* Operating System. Moreover, the specification of the specific system is also illustrated in Table 3. The relative analysis is presented in two segments: (i) Table 4 shows the computational complexity comparison; (ii) Table 5 depicts the security features comparison. Table 4 presents a comparison of the computational cost of the proposed protocol with those of the related protocols [28–30, 35, 36]. Table 5 shows a comparison of the security parameters of the proposed protocol with those of the related protocols in which ✓ means the availability of a specific security feature, × means the absence of that particular security

features mentioned on the left side of the table and shows that the security feature is not applicable.

Fig. 9 presents the comparison of the computational costs of our introduced protocol and those of related protocols. The list of the introduced scheme and the related schemes is given on the X-axis, whereas the computation cost(in ms = milliseconds) is given on the Y-axis. Fig. 9 clearly shows that our protocol is more cost-effective than [28–30,36], but more cost-effective than [35].

For the communication and storage cost comparison, the subsequent suppositions are made: for literals like random numbers, timestamps, identities and passwords, 160 bits are reserved for each; the number of bits for the one-way-hash function is 256 bits; and for symmetric encryption and decryption, 512 bits are assumed. In order to present the storage and communication costs of our introduced scheme and the related scheme, Table 4 specifies the calculation on the basis of the above assumptions.

Fig. 10 shows the comparison of our introduced scheme and related schemes in terms of communication cost. The list of the introduced and related schemes is given on the X-axis, whereas the cost of computation (in bits) is given on the Y-axis. Our proposed protocol performs better as compared to Refs. [29,30,36], but its efficiency in terms of communication cost is compromised against [28,35]. However, our protocol promises to offer better security features.

Fig. 11 shows the storage costs of our introduced scheme and related schemes. A list of the introduced and related schemes is marked on the X-axis whereas the cost of storage (in bits) is given on the Y-axis. Although the storage cost of our proposed scheme is less than those of the related schemes, it offers better security features than the related schemes.

Finally, we can conclude after observing Tables 4 and 5 that although the computational and communication costs of our scheme are slightly higher than those of some of the related schemes, the introduced scheme offers better security features whereas other algorithms are vulnerable to one or many serious security attacks.

8. Conclusion

In this paper, Qiu et al.'s cryptanalysis is presented and described. We find that their scheme has an incorrect notion of perfect user anonymity and is vulnerable to the user impersonation attack. Therefore, we introduce an improved scheme that can resist all the known security attacks against Qiu et al.'s scheme. Our introduced protocol inherits all the security features of Chaudhry et al.'s, Islam et al.'s and Qiu et al.'s. schemes. The performance evaluation unveils that our scheme has slightly higher computational and communication costs than some of the related schemes but offers better security features than related schemes [28–30, 35,36], which are vulnerable to one or many serious security attacks.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.dcan.2021.07.002.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] John Franks, Phillip Hallam-Baker, Jeffrey Hostetler, Lawrence Scott, Paul Leach, Ari Luotonen, Lawrence Stewart, Basic and Digest Access Authentication, Http authentication:, Technical report, 1999 (accessed 12 March 2019).

[2] Chou-Chen Yang, Ren-Chiun Wang, Wei-Ting Liu, Secure authentication scheme for session initiation protocol, Comput. Secur. 24 (5) (2005) 381–386.

[3] Hui-Feng Huang, A new efficient authentication scheme for session initiation protocol, in: 9th Joint International Conference on Information Sciences (JCIS-06), Atlantis Press, 2006, pp.8-11.

[4] Aytunc Durlanik, Ibrahim Sogukpinar, Sip authentication scheme using ecdh, World Enformatika Soc Trans Eng Comput Technol 8 (2005) 350–353.

[5] Razi Arshad, Nassar Ikram, Elliptic curve cryptography based mutual authentication scheme for session initiation protocol, Multimed. Tool. Appl. 66 (2) (2013) 165–178.

[6] Tien-ho Chen, Hsiu-lien Yeh, Pin-chuan Liu, Han-chen Hsiang, Wei-kuan Shih, A secured authentication protocol for sip using elliptic curves cryptography, in: International Conference on Future Generation Communication and Networking, Springer, 2010, pp. 46–55.

[7] Mohammad Sabzinejad Farash, Mahmoud Ahmadian Attari, An enhanced authenticated key agreement for session initiation protocol, Inf. Technol. Contr. 42 (4) (2013) 333–342.

[8] Fuwen Liu, Hartmut Koenig, Cryptanalysis of a sip authentication scheme, in: IFIP International Conference on Communications and Multimedia Security, Springer, 2011, pp. 134–143.

[9] Jia Lun Tsai, Efficient nonce-based authentication scheme for session initiation protocol, IJ Network Security 9 (1) (2009) 12–16.

[10] Hongbin Tang, Xinsong Liu, Cryptanalysis of arshad et al. ecc-based mutual authentication scheme for session initiation protocol, Multimed. Tool. Appl. 65 (3) (2013) 321–333.

[11] Saru Kumari, Li Xiong, Fan Wu, Ashok Kumar Das, Kim-Kwang Raymond Choo, Jian Shen, Design of a provably secure biometrics-based multi-cloud-server authentication scheme, Future Generat. Comput. Syst. 68 (2017) 320–330.

[12] Saru Kumari, Muhammad Khurram Khan, Mohammed Atiquzzaman, User authentication schemes for wireless sensor networks: a review, Ad Hoc Netw. 27 (2015) 159–194.

[13] Saru Kumari, Li Xiong, Fan Wu, Ashok Kumar Das, Hamed Arshad, Muhammad Khurram Khan, A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps, Future Generat. Comput. Syst. 63 (2016) 56–75.

[14] Khalid Mahmood, Husnain Naqvi, Bander A. Alzahrani, Zahid Mehmood, Azeem Irshad, Shehzad Ashraf Chaudhry, An ameliorated two-factor anonymous key exchange authentication protocol for mobile client-server environment, Int. J. Commun. Syst. 31 (18) (2018), e3814.

[15] Saru Kumari, Pradeep Chaudhary, Chien-Ming Chen, Muhammad Khurram Khan, Questioning key compromise attack on ostad-sharif et al. authentication and session key generation scheme for healthcare applications, IEEE Access 7 (2019) 39717–39720.

[16] Saru Kumari, Design flaws of an anonymous two-factor authenticated key agreement scheme for session initiation protocol using elliptic curve cryptography, Multimed. Tool. Appl. 76 (11) (2017) 13581–13583.

[17] Eun-Jun Yoon, Kee-Young Yoo, Cryptanalysis of ds-sip authentication scheme using ecdh, in: 2009 International Conference on New Trends in Information and Service Science, IEEE, 2009, pp. 642–647.

[18] Eun-Jun Yoon, Yong-Nyuo Shin, Il-Soo Jeon, Kee-Young Yoo, Robust mutual authentication with a key agreement scheme for the session initiation protocol, IETE Tech. Rev. 27 (3) (2010) 203–213.

[19] Shaheena Khatoon, Sk Md Mizanur Rahman, Majed Alrubaian, Atif Alamri, Privacy-preserved, provable secure, mutually authenticated key agreement protocol for healthcare in a smart city environment, IEEE Access 7 (2019) 47962–47971.

[20] Hui Qiao, Xuewen Dong, Yulong Shen, Authenticated key agreement scheme with strong anonymity for multi-server environment in tmis, J. Med. Syst. 43 (11) (2019) 321.

[21] Xiaoxue Liu, Wenping Ma, Hao Cao, Mbpa: a medibchain-based privacy-preserving mutual authentication in tmis for mobile medical cloud architecture, IEEE Access 7 (2019) 149282–149298.

[22] Xiaoxue Liu, Wenping Ma, Hao Cao, Npma: a novel privacy-preserving mutual authentication in tmis for mobile edge-cloud architecture, J. Med. Syst. 43 (10) (2019) 318.

[23] Husnain Naqvi, Shehzad Chaudhry, Khalid Mahmood, An improved authentication protocol for sip-based voip, in: International Conference on Recent Advances in Computer Systems, Atlantis Press, 2015, pp. 7–12.

[24] Mohammad Heydari, Seyed Mohammad Sajad Sadough, Shehzad Ashraf Chaudhry, Mohammad Sabzinejad Farash, Khalid Mahmood, An improved one-to-many authentication scheme based on bilinear pairings with provable security for mobile pay-tv systems, Multimed. Tool. Appl. 76 (12) (2017) 14225–14245.

[25] Shehzad Ashraf Chaudhry, Taeshik Shon, Fadi Al-Turjman, Mohammed H. Alsharif, Correcting design flaws: an improved and cloud assisted key agreement scheme in cyber physical systems, Comput. Commun. 153 (2020) 527–537.

[26] Bander A. Alzahrani, Shehzad Ashraf Chaudhry, Barnawi Ahmed, Abdullah Al-Barakati, Mohammed H. Alsharif, A privacy preserving authentication scheme for roaming in iot-based wireless mobile networks, Symmetry 12 (2) (2020) 287.

[27] Xin Xu, Zheng Ping Jin, Hua Zhang, Ping Zhu, A dynamic id-based authentication scheme based on ecc for telecare medicine information systems, in: Applied Mechanics and Materials, vol. 457, Trans Tech Publ, 2014, pp. 861–866.

[28] Sk Hafizul Islam, Muhammad Khurram Khan, Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems, J. Med. Syst. 38 (10) (2014) 135.

[29] Shehzad Ashraf Chaudhry, Husnain Naqvi, Taeshik Shon, Muhammad Sher, Mohammad Sabzinejad Farash, Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems, J. Med. Syst. 39 (6) (2016) 66. https://doi.org/10.1007/s10916-015-0244-0.

[30] Shuming Qiu, Guoai Xu, Haseeb Ahmad, Licheng Wang, A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems, IEEE access 6 (2018) 7452–7463.
[31] Ding Wang, Debiao He, Ping Wang, Chao-Hsien Chu, Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment, IEEE Trans. Dependable Secure Comput. 12 (4) (2015) 428–442.
[32] Ding Wang, Ping Wang, Two birds with one stone: two-factor authentication with security beyond conventional bound, IEEE Trans. Dependable Secure Comput. 15 (4) (2018) 708–722.
[33] Zahid Ghaffar, Shafiq Ahmed, Khalid Mahmood, Sk Hafizul Islam, Mohammad Mehedi Hassan, Giancarlo Fortino, An improved authentication scheme for remote data access and sharing over cloud storage in cyber-physical-social-systems, IEEE Access 8 (2020) 47144–47160.
[34] Salman Shamshad, Khalid Mahmood, Saru Kumari, Chien-Ming Chen, et al., A secure blockchain-based e-health records storage and sharing scheme, Journal of Information Security and Applications 55 (2020) 102590.
[35] Jianghong Wei, Xuexian Hu, Wenfen Liu, An improved authentication scheme for telecare medicine information systems, J. Med. Syst. 36 (6) (2012) 3597–3604.
[36] Hang Tu, Neeraj Kumar, Naveen Chilamkurti, Seungmin Rho, An improved authentication protocol for session initiation protocol using smart card, Peer-to-Peer Networking and Applications 8 (5) (2015) 903–910.