J Med Syst (2015) 39: 78
DOI 10.1007/s10916-015-0258-7

PATIENT FACING SYSTEMS

A Secure Three-Factor User Authentication and Key Agreement Protocol for TMIS With User Anonymity

Ruhul Amin[1] · **G. P. Biswas**[1]

Received: 25 January 2015 / Accepted: 2 June 2015 / Published online: 26 June 2015

Abstract Telecare medical information system *(TMIS)* makes an efficient and convenient connection between patient(s)/user(s) and doctor(s) over the insecure internet. Therefore, data security, privacy and user authentication are enormously important for accessing important medical data over insecure communication. Recently, many user authentication protocols for *TMIS* have been proposed in the literature and it has been observed that most of the protocols cannot achieve complete security requirements. In this paper, we have scrutinized two (Mishra et al., Xu et al.) remote user authentication protocols using smart card and explained that both the protocols are suffering against several security weaknesses. We have then presented three-factor user authentication and key agreement protocol usable for *TMIS*, which fix the security pitfalls of the above mentioned schemes. The informal cryptanalysis makes certain that the proposed protocol provides well security protection on the relevant security attacks. Furthermore, the simulator *AVISPA* tool confirms that the protocol is secure against active and passive attacks including replay and man-in-the-middle attacks. The security functionalities and performance comparison analysis confirm that our protocol not only provide strong protection on security attacks, but it also achieves better complexities along with efficient login and password change phase as well as session key verification property.

Keywords Authentication · *AVISPA* Tool · Elliptic curve · Security attacks · Smart card

This article is part of the Topical Collection on *Patient Facing Systems*

✉ Ruhul Amin
amin_ruhul@live.com
ruhulamin@cse.ism.ac.in

G. P. Biswas
gpbiswas@gmail.com

[1] Department of Computer Science & Engineering, Indian School of Mines, Dhanbad 826004, India

Introduction

In *TMIS*, medical server generally maintains the electronic medical records of the registered patients and provides various services like health educators, physicians, hospitals, care-givers, public health organizations and home-care service. User-friendly, omnipresence and the low cost of internet technology, facilitates online medical services, in which a registered user/patient can access the remote service at any instant from anywhere. When a registered user wants to get medical services, s/he uses smart card to the smart devices and transmits data to the medical server through public channel. The attacker/adversary may have full control over the public channel. Therefore, s/he can eavesdrop, intercept, record, modify, delete, and replay the message broadcasting via public channel. As, the data is transmitted over the public channel, maintaining user authentication, data privacy, data integrity and confidentiality of the data are very much essential. In order to design authentication protocol, many researchers employ several cryptographic algorithms like non-invertible one-way hash function, Chaotic maps, *ECC-RSA* cryptosystem and some others operation like X-OR and concatenate etc. Though *ECC* and *RSA* both cryptosytem provides same level of security, *ECC* is more suitable than *RSA* [15], because *ECC* [16, 19, 21] uses only point multiplication operation and the key length is 160 bits, whereas *RSA* uses exponentiation operation, which takes very much

longer computation than point multiplication and the key length of the *RSA* is 1024 bits, which is larger than *ECC*. On the other hand, hash-function and chaotic maps [17] both plays a crucial role for designing user authentication protocol and provides same level of security, but the computation of hash function is less than the chaotic maps operation.

The main security threats for the password based authenticated key exchange protocols are password guessing attack. As the adversary has maximum capabilities over the communication network, he/she may trap all the communicating messages and guess the user's password in off-line mode. Therefore, password based authentication system may not be fully secured. To the best of our knowledge, the password based authentication system has some problems such as a long and random password cannot be used, because it is very difficult to remember for a use. If the user stores his/her password somewhere, there is possibility for stealing the password. Moreover, the user may be shared the long and random password with other people, but there is no way to identify who is the legal user. To overcome the above problems, most popular and secure biometric technology (fingerprint, iris, retina etc.) is extensively used to authenticate the legal user because of its uniqueness property and does not need to remember. The several advantages of using the biometric technology are as follows:

1. Biometric key cannot be lost or forgotten and very difficult to copy or share.
2. Biometric key is extremely hard to forge or distribute.
3. Guessing biometric key is dreadfully hard.

Attacker model

As the authentication protocol are executed over the insecure communication, the attacker has several advantages or capabilities over the authentication protocol. The several valid assumptions regarding authentication protocol are presented below.

1. An attacker is able to extract the smart card information by monitoring the power consumption [27, 43]. For example if an attacker gets the smart card of the valid user, s/he then may get al.l the stored information of the smart card.
2. An attacker may eavesdrop all the communication between the entities involved of the protocol over the public channel. It is also assume that an attacker cannot intercept the message over the secure channel.
3. According to reference [48], an attacker can guess low entropy password and identity individually easily but guessing two secret parameters (e.g. password, identity) is computationally infeasible in polynomial time.
4. An attacker can modify, delete and resend, reroute the eavesdrops message.
5. An attacker may be a legitimate user or vice versa.
6. If we assume that the length of the user's identity and password is n character, then the probability of guessing approximately composed of n character is $\frac{1}{2^{6n}}$ as pointed out by [10].

Literature review

To ensure security and privacy during information transmission via public channel, the smart card based anonymous remote user authentication schemes are generally adopted. Last few years, many password or biometric template based remote user authentication and key agreement protocols [1, 3–5, 12, 18, 20, 22, 25, 26, 28–34, 38] have been proposed in the literature for different application systems. But, none of them is completely free from security attacks. In 2010, Wu et al. [52] proposed an efficient authentication scheme for *TMIS* and adding a pre-computing phase for low computational cost. But, Debiao He [13] demonstrated that Wu et al. [52] protocol fails to resist impersonation attack and insider attack and presents an enhance scheme of Wu et al. protocol and claimed that the enhance scheme is completely free from security attacks and takes low computational cost. In 2012, Wei et al. [51] identified that both Wu et al. and Debiao He protocols are inefficient to meet two-factor authentication and also proposed a scheme, which is efficient and achieves two-factor authentication. Thereafter, Zhu [56] described that Wei et al. protocol is vulnerable to off-line password guessing attack and also proposed an improved scheme for *TMIS*. Then, Lee and Liu [35] demonstrated that Zhu's scheme cannot resist parallel session attack and presented a improved scheme and declared that their protocol is efficient in terms of security and applicable for *TMIS* systems.

In 2012, dynamic-ID based authentication and key agreement protocol is presented by Chen et al. [11] But, Lin [39] demonstrated that Chen et al.'s protocol suffers from user anonymity problem and password can be derived from the stolen smart card. Later, Cao and Zhai [9] demonstrated that Chen et al. protocol is vulnerable to off-line identity guessing attack, off-line password guessing attack and un-detectable online password guessing attack when the user's smart card is lost. They also presented an improved scheme for *TMIS*. Thereafter, Xie et al. [53] described that Chen et al.'s protocol suffers from security weaknesses and proposed an improved scheme. In 2013, Tan et al.'s [49] proposed a biometric based remote user authentication scheme for telecare medical information system and declared that their protocol achieves mutual authentication property and session key agreement between the user and the server.

In 2013, Awasthi-Srivastava [8] proposed three-factor based user authentication and key agreement protocol for *TMIS* and declared that the same protocol should be

application in real-time application. However, Mishra et al.'s [46] pointed out that the protocol [8] is not secure against undetectable password guessing attack along with inefficient password change phase. Additionally, they proposed three-factor based authentication protocol for enhancing security of the scheme [8]. Recently, Lu et al.'s [41] also showed that Arshad et al's [7] cannot provide complete security requirements and proposed an improved scheme over scheme [7]. Furthermore, Zhang et al.'s [40] introduced three-factor based user authentication protocol for enhancing security and performance. In the same year, Yan et al.'s [55] reviews the proposed protocol presented by Tan et al.'s and declares that the scheme is vulnerable to denial-of-service attack. To eliminate the drawbacks of Tan et al.'s protocol, Yan et al.'s proposed an improved scheme for better security protection and performance. In 2014, Mishra et al. [45] demonstrated that Yan et al. protocol suffers from user anonymity problem, password guessing attack, inefficient login phase, inefficient password and biometric update phase and three factor authentication problem. They also proposed and improved scheme for better security and performance. Recently, Mrudula et al.'s [47] pointed out that the Mishra et al.'s [45] protocol is insecure against off-line password guessing attack and user impersonation attack. We further described that the Mishra et al.'s protocol cannot withstand server impersonation attack, session key computation attack and smart card theft attack. In 2014, Xu et al.'s [54] proposed a *ECC* based remote user authentication and key agreement scheme for telecare medical information system and claimed that their protocol achieves higher security along with better performance. After that, Mishra [44] showed that the same protocol has security vulnerability on the login and authentication phase and then Islam-Khan [23] further demonstrated that the same protocol [54] is not secure against replay attack and cannot provide efficient authentication phase. However, we have carefully scrutinized the Xu et al.'s protocol and further explained that the protocol proposed by Xu et al.'s cannot provide efficient password change phase. In order to fix the limitations of Mishra et al. [45] and Xu et al. [54], this proposed *ECC* based user authentication and key agreement protocol using smart card for *TMIS* and also analyzed the security as well as performance evaluation of the proposed scheme.

Motivation and contributions

In the literature review section, we observed that most of the protocols suffer from security weaknesses, which ensures that the protocols are not suitable for practical implementation. Therefore, we are motivated to develop biometric based user authentication protocol using smart card usable for telecare medical information system. In this paper, we achieve the following contributions:

(1) We have shown that the Mishra et al.'s protocol has security weaknesses such as server impersonation attack, smart card theft attack and session key discloser attack. Moreover, we have also demonstrated that the Xu et al.'s protocol has security weaknesses in the password update phase.
(2) In order to overcome the weaknesses, we have proposed an efficient user authentication protocol using smart card whose performance analysis ensures that the computation and communication costs are relatively better than the existing related protocols. Moreover, the proposed scheme provides efficient login and password update phase along with session key verification property.
(3) We have simulated our proposed protocol using widely accepted *AVISPA* simulator tool which ensures that the protocol is *SAFE* under *OFMC* and *CL-AtSe* models.

Road map of the paper

In "Preliminaries" section, we discussed the concept and the property of cryptographic one-way hash function, bio-hashing techniques, elliptic curve cryptosystem and some computational problems as preliminaries of our works. In "Brief review of Mishra et al. scheme" section, we briefly review Mishra et al. recently published protocol and the security analysis of [45] scheme is given in "Security flaws of the Scheme proposed by Mishra et al." section. "Brief review of Xu et al. scheme" section addresses Xu et al. recently published protocol and the security weaknesses appears in "Security flaws of the scheme proposed by Xu et al" section. "Proposed key agreement protocol for TMIS system" section presents our proposed protocol for *TMIS* and the security attack protection discussion is presented in "Cryptanalysis of the proposed scheme" section. The performance comparison is also made and given in "Performance evaluation" section. Finally, we conclude the paper in " Conclusion and future work" section and completes the paper with references.

Preliminaries

In this section, We briefly review the basic concepts of cryptographic one-way hash function, bio-hashing, *ECC* cryptosystem along with some hardness problems are introduced.

Cryptographic one-way hash function

A cryptographic one-way hash function maps a string of arbitrary length to a string of fixed length called the hashed value. It can be symbolized as: $h : X \rightarrow Y$, where

$X = \{0,1\}^*$, and $Y = \{0,1\}^n$. X is binary string of arbitrary length and Y is a binary string of fixed length n. It is used in many cryptographic applications such as digital signature, random sequence generators in key agreement, authentication protocols and so on. Cryptographic one-way hash function satisfies the following properties:

1. *Easiness:* Given $m \in X$, it can be easily compute y such that $y = h(m)$.
2. *Preimage Resistant:* It is hard to find m from given y, where $h(m) = y$.
3. *Second-Preimage Resistant:* It is hard to find input $m' \in X$ such that $h(m) = h(m')$ for given input $m \in X$ and $m' \neq m$.
4. *Collision Resistant:* It is hard to find a pair $(m, m') \in X \times X$ such that $h(m) = h(m')$, where $m \neq m'$.
5. *Mixing-Transformation:* On any input $m \in X$, the hashed value $y = h(m)$ is computationally indistinguishable from a uniform binary string in the interval $\{0, 2^n\}$, where n is the output length of hash $h(\cdot)$.

It is our assumption that this paper uses *SHA-2* hash function for achieving top security whose message digest is 160 bits.

Bio-hashing

The biometric technology has the great importance for providing genuine user authentication in any authentication system. Generally, imprint biometric characteristics (face, fingerprint, palmprint etc.) may not be exactly same at each time. Therefore, high false rejection of registered users resulting low false acceptation, is often occurs in the evaluation of biometric systems. In order to resolve the high false rejection rate, Jina et al. [24] proposed a two-factor authenticator on iterated inner products between tokenised pseudo-random number and the user specific fingerprint features, which produces a set of user specific compact code that coined as Bio-Hashing. Later, Lumini and Nanni [42] proposed the improvement of Bio-Hashing. As pointed out by [10], Bio-Hashing is used to map a user/patients biometric feature onto user specific random vectors in order to generate a code, called biocode and then discritizes the projection coefficients into zero and one. Bio-Hashing is always one-way function and secure as cryptographic one-way hash function.

ECC-cryptosystem

The elliptic curve cryptosystem was initially proposed by Koblitz (1987) and Miller (1985) to design public key cryptosystem and presently it is widely used in several cryptographic schemes to provide desired level of security and computational efficiency. Let $E_p(a, b)$ be a set of elliptic curve points over the prime field F_p defined by the non-singular elliptic curve equation: $y^2 mod\ p = x^3 + ax + b\ mod\ p$ with $(a, b) \in F_p$ and $4a^3 + 27b^2\ mod\ p \neq 0$. The additive *ECC* group defined as $G_p = \{(x, y) : x, y \in F_p$ and $x, y \in E_p(a, b)\} \bigcup \{O\}$, where the point O is known as "point at infinity". The scalar multiplication on the cyclic group G_p defined as: k.P = (P + P + P ..... k times) that means k times addition of point P. There are several computational problem based on *ECC* which are presented below:

Definition 1 The elliptic curve discrete logarithm problem *(ECDLP)* is defined as: Given $Q, R \in G_p$, find an integer $k \in [1, n-1]$ such that $R = k.Q$.

Definition 2 The computational Diffie?Hellman problem *(CDHP)* is defined as: Given (P, aP, bP) for any $a, b \in [1, n-1]$ computation of abP is very hard to the group G_p.

Definition 3 The elliptic curve factorization problem *(ECFP)* is defined as: Given $P, Q \in G_p$, where $Q = sP + tP$ and $(s, t) \in [1, n-1]$, then computation of sP and tP is impossible.

Definition 4 The decisional Diffie?Hellman problem is defined as: Given (P, aP, bP, cP) for any $(a, b, c) \in [1, n-1]$, decide whether or not $cP = abP$ i.e. $c = ab\ mod\ p$ or not.

Definition 5 The weak Diffie?Hellman problem *(WDHP)* is defined as: For $Q \in G_p$ and some $k \in [1, n-1]$ from the given triplet (P, Q, kP) computation of kQ is hard.

Brief review of Mishra et al. scheme

In this section, we briefly review Mishra et al. protocol, which is the improvement of Yan et al. protocol. In Table 1, we have presented all the notations used throughout this paper. Like Yan et al. protocol, Mishra et al. protocol has mainly four phases such as registration phase, login phase, authentication phase and password change phase. All the mentioned phases are presented below. In Fig. 1, we have demonstrated all the phases of the Mishra et al. scheme.

Registration phase

For the new user registration, this phase executes all the steps which are discussed below:

Step 1: User U_i chooses identity ID_i, Password PW_i, biometric template B_i, a random number N_i and computes $W = h(ID_i \parallel PW_i \parallel N_i)$ and then submits

Table 1 List of notations used

Symbol	Description
U_i	i-th User/patients
S	Medical server
PW_i	Password of the user U_i
ID_i	Identity of the user U_i
B_i	Biometric of the user U_i
p, q	Two large prime numbers
F_p	Finite field
$E(F_p)$	elliptic curve over F_p defined by the equation $y^2 = x^3 + ax + b$, where a,b $\in F_p$ $4a^3 + 27b^2 \neq 0$
G	cyclic additive group consisting of points on $E(F_p)$ that has a specific point called the infinite point;
P	Generator point of G with the order q;
aP	a times addition of point P
x	Secret key of the server (S) (1024 bits)
Z_p^*	The multiplicative group of Z_p
$h(\cdot)$	secure One-way hash function: $(0, 1)^* \rightarrow Z_p^*$,
$h_1(\cdot)$	secure One-way hash function: $(G_p, G_p) \rightarrow Z_p^*$,
$H(\cdot)$	Bio-hashing function:
$\oplus$	Bit-wise Xor operation
$\|$	Concatenation operation

registration message $\langle ID_i, W\rangle$ to the server S through secure channel.

Step 2: S computes $X_i = h(ID_i \parallel x)$, $Y_i = X_i \oplus W$, where x is the server's secret key. Then, S generates a random nonce R and uses symmetric key cryptosystem such as *AES-256* to compute $NID = Sym.Enc_{(x)}(ID_i \parallel R)$ and issues a smart card after storing secret parameters $\langle NID, Y_i, h(\cdot)\rangle$ into the memory of smart card. It may be noted that Mishra et al.'s protocol does not store biohashing function $H()$ in the memory of smart card. So, the computation of $H(B_i)$ is not feasible in the login phase. Hence, it is required to store $H()$ into memory of smart card.

Step 3: After getting the smart card, U_i stores $N = N_i \oplus H(B_i)$ and $V_i = h(ID_i \parallel PW_i \parallel N_i)$ into the smart card.

Login phase

This phase executes when U_i wishes to access server's resources. All the steps of this phases are presented below:

Step 1: U_i provides $\langle ID_i, PW_i\rangle$ to the smart devices and B_i at the sensor device.

Step 2: Smart card then computes $N_i = N \oplus H(B_i)$, $V_i^* = h(ID_i \parallel PW_i \parallel N_i)$ and matches V_i^* with the stored V_i. If it matches, continues next steps; otherwise, terminates the session.

Step 3: Smart card further computes $W = h(ID_i \parallel PW_i \parallel N_i)$, $X_i = Y_i \oplus W$ and then generates a random nonce r_i and again computes $a_i = h(ID_i \parallel X_i \parallel r_i)$. Then, the smart card sends $\langle NID, a_i, r_i\rangle$ to S through public channel.

Authenticated key agreement phase

In order to accomplish mutual authentication and session key agreement, the U_i and S execute all the steps which are presented below.

Step 1: Server first retrieves ID_i by decrypting NID parameter and computes $X_i = h(ID_i \parallel x)$, $a_i^* = h(ID_i \parallel X_i \parallel r_i)$. Then, S matches the value of a_i^* with a_i, if it matches, continues next steps; otherwise, terminates the session.

Step 2: In this steps, S generates random numbers $\langle r_s, R'\rangle$ and computes $sk = h(ID_i \parallel X_i \parallel r_i \parallel r_s)$, $NID' = SymEnc_{(x)}(ID_i \parallel R')$, $b_i = h(ID_i \parallel NID \parallel sk \parallel NID')$. Server then sends a message $\langle r_s, b_i, h(sk \parallel ID_i) \oplus NID'\rangle$ to the user.

Step 3: Upon receiving the message, the smart card computes the session key $sk = h(ID_i \parallel X_i \parallel r_i \parallel r_s)$ and retrieves $NID' = h(sk \parallel ID_i) \oplus NID' \oplus h(sk \parallel ID_i)$ and further computes $b_i^* = h(ID_i \parallel NID \parallel sk \parallel NID')$. Then, S matches computed b_i^* with the received b_i. If it does not match, terminates the connection; otherwise, continues the next steps.

Step 4: Smart card computes $c_i = h(ID_i \parallel NID' \parallel sk)$ and sends the verification message $\langle c_i\rangle$ to the server S through public channel.

Step 5: Upon receiving the verification message, server computes $c_i^* = h(ID_i \parallel NID' \parallel sk)$ and matches it with the received c_i. If it matches, session key is verified and user is authenticated; otherwise, stops the session.

Password and biometric update phase

As described earlier in the login phase, Steps 1-2 are same. Therefore. we discuss rest of the steps which are described below.

Step 3: U_i inputs password PW_i', biometric template B_i' and random number N_i'.

Step 4: After inputting, smart card computes the following parameters: $W = h(ID_i \parallel PW_i \parallel N_i)$, $W_{new} = h(ID_i \parallel PW_i' \parallel N_i')$, $Y_{new} = Y_i \oplus W \oplus W_{new}$, $V_{new} = h(ID_i \parallel PW_i' \parallel N_i')$, $N_{new} = N_i' \oplus H(B_i')$. Then, the smart card replaces $\langle Y_i, N, V_i\rangle$ with the new values $\langle Y_{new}, N_{new}, V_{new}\rangle$ respectively and completes password and biometric update phase.

User U_i/Smartcard		Server S
	Registration Phase	
User U_i Choose $\langle ID_i, PW_i, B_i \rangle$ $W = h(ID_i \parallel PW_i \parallel N_i)$		
	$\longrightarrow \langle ID_i, W \rangle$	S computes $X_i = h(ID_i \parallel x)$, $Y_i = X_i \oplus W$ $NID = Sym.ENC_x(ID_i \parallel R)$ Stores $\langle NID, Y_i, h() \rangle$ into SC
	$\Longleftarrow$ delivers SC to U_i	
U_i stores $N = N_i \oplus H(B_i)$ $V_i = h(ID_i \parallel PW_i \parallel N_i)$ and completes this phase.		
	Login and Authentication Phase	
Insert smart card and inputs $\langle ID_i, PW_i, B_i \rangle$ computes $N_i = N \oplus H(B_i)$ $V_i^* = h(ID_i \parallel PW_i \parallel N_i)$ Check $V_i^= ? V_i$ If holds, generates number r_i $W = h(ID_i \parallel PW_i \parallel N_i)$ $X_i = Y_i \oplus W$ $a_i = h(ID_i \parallel X_i \parallel r_i)$		
	$\longrightarrow \langle NID, a_i, r_i \rangle$ to S	
		S extract ID_i from NID_i and computes $X_i = h(ID_i \parallel x)$ $a_i^* = h(ID_i \parallel PW_i \parallel r_i)$ checks $a_i^* = ? a_i$ If holds, generates r_s, $R^{'}$ and computes $sk = h(ID_i \parallel X_i \parallel r_i \parallel r_s)$ $NID^{'} = Sym.Enc_x(ID_i \parallel R^{'})$ $b_i = h(ID_i \parallel NID \parallel sk \parallel NID^{'})$
	$\longleftarrow \langle r_s, b_i, h(sk \parallel ID_i) \oplus NID^{'} \rangle$	
$sk = h(ID_i \parallel X_i \parallel r_i \parallel r_s)$ $NID^{'} = h(sk \parallel ID_i) \oplus NID^{'} \oplus$ $h(sk \parallel ID_i)$ $b_i^* = h(ID_i \parallel NID \parallel sk \parallel NID^{'})$ Checks $b_i^* = ? b_i$		
$c_i = h(ID_i \parallel NID^{'} \parallel sk)$	$\longrightarrow \langle c_i \rangle$	
		S computes $c_i^* = h(ID_i \parallel NID^{'} \parallel sk)$ Checks $c_i^* = ? c_i$ *(sk verification)*
	PW_i and B_i Change Phase	
U_i inputs new $\langle PW_i^{'}, B_i^{'}, N_i^{'} \rangle$ $W = h(ID_i \parallel PW_i \parallel N_i)$ $W_{new} = h(ID_i \parallel PW_i^{'} \parallel N_i^{'})$ $Y_{new} = Y_i \oplus W \oplus W_{new}$ $V_{new} = h(ID_i \parallel PW_i^{'} \parallel N_i^{'})$ $N_{new} = N_i^{'} \oplus H(B_i^{'})$ and Replace $\langle Y_i, N_i, V_i \rangle$ with new $\langle Y_{new}, N_{new}, V_{new} \rangle$		

Fig. 1 Mishra et al.'s scheme [45]

Security flaws of the Scheme proposed by Mishra et al.

This section presents several security flaws of the scheme proposed by Mishra et al. They have mentioned that if the attacker gets the smart card of the user U_i, s/he can extract all the information stored in the smart card memory by monitoring the power consumption [27, 43]. Based on assumption mentioned in the attacker model, this section discusses several security attacks such as server impersonation attack, smart card theft attack and session key discloser attack. As mentioned in [47], the attacker knows valid ID_i from the Mishra et al.'s protocol description.

Server impersonation attack

In this attack, the attacker can impersonate as a valid server after intercepting the reply message of the Mishra et al. protocol. The detail description of this attack is presented below:

Step 1: The attacker first intercepts login-reply messages of the protocol and computes $sk^a = h(ID_g \parallel X_i^a \parallel r_i \parallel r_s)$ and then retrieves $NID' = h(sk \parallel ID_i) \oplus NID' \oplus h(sk^a \parallel ID_g)$.

Step 2: Attacker generates a random nonce R_s and computes $sk^{**} = h(ID_g \parallel X_i^a \parallel r_i \parallel R_s)$, $b_i^{**} = h(ID_g \parallel NID \parallel sk^{**} \parallel NID')$, $c_i^{**} = h(sk^{**} \parallel ID_g) \oplus NID'$ and sends $\langle R_s, b_i^{**}, c_i^{**} \rangle$ to the smart card user U_i.

Step 3: After receiving message from the attacker, U_i computes $sk = h(ID_i \parallel X_i \parallel r_i \parallel R_s)$ and then retrieves $NID' = c_i^{**} \oplus h(sk \parallel ID_i)$. Further the U_i computes $b_i = h(ID_i \parallel NID \parallel sk \parallel NID')$. Finally, the U_i checks whether the computed b_i matches with the received b_i^{**}. If it matches, user believes that the server is authentic, but it is not true.

The above description clearly states that the protocol proposed by Mishra et al. is insecure against server impersonation attack.

Smart card theft attack

The scheme [47] described that the attacker knows valid ID_i from the scheme [45]. In this attack, we will describe that the attacker can act as a valid user after getting the legal user's smart card by some means. This is clear from the following descriptions:

Step 1: The attacker chooses a password PW_i^a and utilizes his/her biometric template B_i like fingerprint and then computes $W = h(ID_i \parallel PW_i^a \parallel N_i^a)$, where N_i^a is the random number chosen by the attacker.

Step 2: From the smart card parameters, The attacker computes $X_i = Y_i \oplus V_i = h(ID_i \parallel x)$, $Y_i^a = X_i \oplus W$ and then replaces Y_i with the new value Y_i^a into the memory of smart card, where the value of NID is remain unchanged. Attacker further computes $N = N_i^a \oplus H(B_i)$ and $V_i^a = h(ID_i \parallel PW_i^a \parallel N_i^a)$ and stores $\langle N, V_i^a \rangle$ into the memory of smart card.

Step 3: It is noticeable that the new computed smart card parameters are valid. Therefore, the smart device and the server cannot detect the attacker that resulting the attacker easily can access medical server at anytime. This is a very serious attack on Mishra et al.'s protocol.

Session key computation attack

The authenticated session key is used for secure communication between the entities involved, and an attacker upon disclosure of the key can decrypt the secret information. So, the secrecy of session key is the mandatory property of any key agreement protocol. However, Mishra et al.'s protocol is insecure against the session key disclosure attack. The description of the above attack is presented below:

Step 1: The attacker knows valid ID_i of a legal U_i using off-line identity guessing attack.

Step 2: The attacker knows $X_i = h(ID_i \parallel x)$ by computing $Y_i \oplus V$.

Step 3: The attacker also knows $\langle r_c, r_s \rangle$ from the login-reply messages.

Step 4: As the session key of the Mishra et al.'s protocol relies on the $\langle ID_i, X_i, r_i, r_s \rangle$, the attacker can easily compute $SK = (h(ID_i \parallel X_i \parallel r_i \parallel r_s))$ and decrypt the cipher messages exchanged between the user and the medical server.

Brief review of Xu et al. scheme

This section briefly introduces Xu et al. protocol, which is based on the elliptic curve cryptosystem. The Xu et al.'s protocol consists mainly four phases namely registration phase, login phase, authentication phase and password change phase. All the phases of Xu et al.'s protocol are presented below and also demonstrates in Fig. 2.

Registration phase

This phase consists of the following steps:

Step 1: User U_i chooses his/her identity ID_i, Password PW_i and computes $A = h(PW_i \parallel r)$. Then U_i sends $\langle ID_i, A \rangle$ to the telecare server S through secure channel.

Step 2: After getting the registration message, S computes $M = h(x \oplus ID_i)$ and $B = M \oplus A$. Then, S issues a smart card after storing $\langle E_p, P, Y, B, h(), h_1() \rangle$ into memory of smart card through secure channel for each U_i.

Step 3: After receiving the smart card, U_i stores r into the smart card and uses it properly in the future.

User U_i/Smartcard		Server S
	Registration Phase	
User U_i 1. Choose $\langle ID_i, PW_i \rangle$ computes $A_i = h(PW_i \parallel r)$		
	$\Longrightarrow \langle ID_i, A_i \rangle$	1. S computes $M = h(x \oplus ID_i)$, $B = M \oplus A$ Stores $\langle E_p, P, Y, B, h(), h_1() \rangle$ into SC
	$\Longleftarrow$ delivers SC to U_i	
2. The U_i stores r into SC.		
	Login and Authentication Phase	
1. Insert the smart card and 2. inputs $\langle ID_i, PW_i \rangle$ 3. SC computes $A = h(PW_i \parallel r)$ $M = B \oplus A$, $C_1 = a \cdot P$ $C_2 = a \cdot Y$, $CID = ID_i \oplus h_1(C_2)$ $F = h(ID_i \parallel M \parallel T_1)$, a = random number	$\longrightarrow \langle C_1, CID, F, T_1 \rangle$ to S	
		S checks timestamp validity. computes $C_2^{'} = x \cdot C_1$ $ID_i^{'} = CID \oplus h_1(C_2^{'})$, $M^{'} = h(x \oplus ID_i^{'})$ $F^{'} = h(ID_i^{'} \parallel M^{'} \parallel T_1)$ checks $F_i^{'} = ?F_i$ If holds, generates c and computes $D_1 = c \cdot P$, $D_2 = c \cdot C_1$ $sk = h(ID_i^{'} \parallel h_1(D_2) \parallel M^{'})$ $G = h(sk \parallel M^{'} \parallel T_2)$
	$\longleftarrow \langle D_1, G, T_2 \rangle$	
U_i checks timestamp validity. computes $D_2^{'} = a \cdot D_1$ $sk^{'} = h(ID_i \parallel h_1(D_2^{'} \parallel M))$ $G^{'} = h(sk^{'} \parallel M \parallel T_2)$ Checks $G_i^{'} = ?G_i$		
	$sk = h(ID_i \parallel h_1(D_2^{'} \parallel M))$	
	PW_i ***Change Phase***	
U_i inputs $\langle ID_i, PW_i \rangle$ SC computes $A = h(PW_i \parallel r)$ $B = M \oplus A$, U_i input new PW_{new} $A_{new} = h(PW_{new} \parallel r)$ $B_{new} = A_{new} \oplus M$ Replaces $\langle B \rangle$ with new $\langle B_{new} \rangle$		
	$\Longrightarrow$: Secure channel	$\longrightarrow$: Insecure channel

Fig. 2 Xu et al.'s scheme [54]

Login phase

When U_i wants to get services, s/he needs to send login message to the server. All the steps of this phase are presented below:

Step 1: U_i first inserts his/her smart card to the smart devices and inputs $\langle ID_i, PW_i \rangle$. Smart card then performs the following operation:

$$
\begin{aligned}
A &= h(PW_i \parallel r) \\
M &= B \oplus A \\
C_1 &= a \cdot P \\
C_2 &= a \cdot Y \\
CID &= ID_i \oplus h_1(C_2) \\
F &= h(ID_i \parallel M \parallel T_1)
\end{aligned}
$$

where a is the random nonce chosen by the U_i and T_1 is the current timestamp.

Step 2: Smart card then sends the login message $\langle C_1, CID, F, T_1 \rangle$ to the server S through public channel.

Authentication phase

This phase performs mutual authentication and session key agreement between the U_i and the S. All the steps of this phase are presented below:

Step 1: After receiving the login message, S checks the validity of T_1. If it is not true, terminates the session; otherwise, S computes the following operation:

$$C_2' = x \cdot C_1$$

$$ID_i' = CID \oplus h_1(C_2')$$
$$M' = h(x \oplus ID_i')$$
$$F' = h(ID_i' \parallel M' \parallel T_1)$$

Then, S checks the condition $F' = F$. If the condition holds, S believes that U_i is a legal user and continues next steps; otherwise, terminates the session.

Step 2: In this step, S generates a random nonce c and takes current timestamp T_2 and then computes the following operations:

$$D_1 = c \cdot P, D_2 = c \cdot C_1$$
$$sk = h(ID_i' \parallel h_1(D_2) \parallel M')$$
$$G = h(sk \parallel M' \parallel T_2)$$

Then, S sends reply message $\langle D_1, G, T_2 \rangle$ to the U_i through public channel.

Step 3: After receiving the reply message, U_i checks whether T_2 is valid or not. If it is not valid, terminates the connection; otherwise, computes $D_2' = a \cdot D_1$, $sk' = h(ID_i \parallel h_1(D_2') \parallel M)$, $G' = h(sk' \parallel M \parallel T_2)$ and matches the computed G' with the received G. If it matches, the U_i believes that the S is authentic. After that, both parties S and U_i share a common session key $sk = h(ID_i \parallel h_1(D_2) \parallel M)$ for subsequent communications.

Password update phase

This phase works when a U_i wants to change his/her password. It needs to perform the following steps:

Step 1: U_i firstly enters $\langle ID_i, PW_i \rangle$ and then smart card computes $A = h(PW \parallel r)$, $B \oplus A = M$

Step 2: U_i then inputs new password PW_{new} and smart card computes $A_{new} = h(PW_{new} \parallel r)$, $B_{new} = A_{new} \oplus M$. Then, the smart card replaces B with B_{new} and updates the new password successfully.

Security flaws of the scheme proposed by Xu et al.

Xu et al. proposed an efficient protocol in terms of security for the *TMIS* system. However, we have pointed out that their protocol has security weaknesses in the password change phase and presented below.

Design flaws in the password update phase

In the password update phase of Xu et al. protocol, U_i first provides ID_i and password PW_i to the smart devices. It then computes some parameters and asks to input new password to the U_i. Then, U_i provides new password PW_i^{new} to the smart device and updates some information(s) of the smart card. It may be noted that the smart devices never verify the old password before updating the new password into the smart card. In this regard, there may arise two difficulties such as 1) Password change after smart card lost, 2) lacks of properly password update in the password update phase of the Xu et al. protocol.

Password change after smart card lost

We assume that an attacker/non-registered user has stolen U_i's smart card by some means and inserted it into the smart device and entered identity $ID_i^a \neq ID_i$ and password $PW_i^a \neq PW_i$. Since, the smart device never verifies old user password or identity, it straightforwardly computes $A' = h(PW_i^a \parallel r)$, $M' = B \oplus A' \neq M$ and then asks for a new password PW_i^{new} to the U_i. After that, the smart devices computes $A^{new} = h(PW^{new} \parallel r)$, $B^{new} = A^{new} \oplus M$ and updates the smart card with the new parameters $\langle A^{new}, B^{new} \rangle$ and delivers it to the U_i by some means. This case is applicable where the smart card may be returned to the original user U_i by some means. the original user U_i will be failed to access the server's resources for the subsequent transaction, because of inefficient password change phase.

Lacks of properly password update

As mention earlier, smart devices never verify old password in the password update phase. In order to update password, we supposed that U_i inputs correct identity ID_i and wrong password PW_i^w by mistake in password update phase. Then, the smart device straightforwardly computes $A^* = h(PW_i^w \parallel r)$, $M^* = B \oplus A^*$. It may be noted that, $A^* \neq A$ and $M^* \neq M$ as $PW_i^w \neq PW_i$.

Now, smart device asks to input new password to the U_i and s/he inputs new password PW_i^{new} and then smart device computes $A^{new} = h(PW^{new} \parallel r)$, $B^{new} = A^{new} \oplus M^*$. Finally, the smart device replaces $\langle B \rangle$ with the $\langle B^{new} \rangle$. There may arise several difficulties which are discussed below.

A. The value of M^* should be $h(x \oplus ID_i)$ but actually $M^* = h(x \oplus ID_i) \oplus h(PW_i \parallel r) \oplus h(PW_i^w \parallel r)$, which is $\neq h(x \oplus ID_i)$, since $PW_i \neq PW_i^w$.

B. It is also noticeable that the value of B^{new} is dependent on the wrong entered password PW_i^w and the new password PW^{new}.

C. Valid user U_i thinks that the password is updated successfully, so s/he uses new password PW_i^{new} for the next transaction and so on. Since, smart device never verifies password in the login phase of the Xu et al.

protocol, so each and every time login message will be created and forwarded to the S. After checking the condition, S rejects the connection due to invalid login message, as the login message is created by the wrong entered password PW_i^w and U_i then believes that the server acts as a fraud but it is wrong. This problem can be avoided, if the smart devices check user's password at the beginning of the login phase. Thus, it is clear that Xu et al. protocol has design flaws in the password update phase.

Proposed key agreement protocol for *TMIS* system

In this section, we proposed a user authentication and key agreement protocol based on the cryptographic one-way hash function and *ECC* using smart card applicable for *TMIS*. In our scheme, there are several phases like user registration phase, login phase, key agreement and mutual authentication phase and password change phase. All these phases of our proposed protocol are presented below.

User U_i/Smartcard		**Server S**
	Registration Phase	
User U_i Choose $\langle ID_i, PW_i, T_i \rangle$ $A_i = h(ID \parallel PW_i)$ $F_i = H(T_i)$		
	$\Longrightarrow \langle ID_i, A_i, F_i \rangle$	computes $W = h(ID_s \parallel x \parallel ID_i)$, $B_i = h(ID_i \parallel A_i) \oplus W$ $CID_i = ENC_x(ID_i \parallel Ran)$ Embeds $\langle F_i, CID_i, A_i, B_i, h(), H() \rangle$ in SC
	$\Longleftarrow$ delivers SC to U_i	
	Login and Authentication Phase	
Insert the smart card and inputs $\langle ID_i, PW_i, T_i \rangle$ computes $F_i^* = H(T_i) = ?F_i$ $A_i^* = h(ID_i \parallel PW_i) = ?A_i$ generates random number r_i $C_1 = r_i \cdot P$ $W = B_i \oplus h(ID_i \parallel A_i)$ $C_2 = r_i \oplus W$ $C_4 = h(ID_i \parallel r_i \parallel W)$	$\longrightarrow \langle C_2, CID_i, C_4 \rangle$ to S	
		S extracts ID_i from CID_i S computes $W = h(ID_s \parallel x \parallel ID_i)$ $r_i^* = C_2 \oplus W$, $C_1^* = r_i^* \cdot P$ $C_4^* = h(ID_i^* \parallel r_i^* \parallel W)$ Checks $C_4^* = ?C_4$ Generates random number r_j $D_1 = r_j \cdot P$, $SK = r_j \cdot C_1^*$ $G_1 = D_1 + C_1^*$, $L_i = h(ID_i^* \parallel h_1(D_1) \parallel W)$ $CID_i^{'} = ENC_x(ID_i \parallel Ran^{'})$
	$\longleftarrow \langle L_i, G_1, CID_i^{'} \rangle$	
U_i computes $D_1^* = G_1 - C_1^*$ $L_i^* = h(ID_i \parallel h_1 D_1^* \parallel W)$ $SK = r_i \cdot D_1^* = r_i \cdot r_j \cdot P$ Checks $L_i^* = ?L_i$ Computes $Z_i = h(ID_i \parallel SK)$	$\longrightarrow \langle Z_i \rangle$	
		S computes $Z_i^* = h(ID_i \parallel SK)$ checks $Z_i^* = Z_i$
	PW_i Change Phase	
U_i inputs $\langle ID_i, PW_i \rangle$ SC computes $F_i^* = H(T_i) = ?F_i$ $A_i^* = h(ID_i \parallel PW_i) = ?A_i$ Input new PW_i^{new} $A_i^{new} = h(ID_i \parallel PW_i^{new})$ $B_i^{new} = h(ID_i \parallel A_i^{new}) \oplus W$ Replaces $\langle A_i, B_i \rangle$ with $\langle A_i^{new}, B_i^{new} \rangle$		
	$\Longrightarrow$: Secure channel	$\longrightarrow$: Insecure channel

Fig. 3 Description of the proposed scheme

In Fig. 3, we have explained all the phases of the proposed protocol.

User registration phase

It is the initial phase for accessing the medical services and any user can register with the medical server. The user chooses his/her desired identity ID_i, password PW_i, biometric template like fingerprint T_i and sends $\langle ID_i, A_i, F_i \rangle$ to the medical server through secure channel or in person after computing $A_i = h(ID_i \parallel PW_i)$ and $F_i = H(T_i)$. After receiving the registration request, medical server S computes $A_i = h(ID_i \parallel PW_i)$, $W = h(ID_s \parallel x \parallel ID_i)$, $B_i = h(ID_i \parallel A_i) \oplus W$, $CID_i = ENC_x(ID_i \parallel R_{ran})$ and issues a smart card for the U_i after storing $\langle F_i, A_i, B_i, CID_i, h(), H() \rangle$ into the memory of smart card through secure channel and completes the registration process, where ID_s is the identity of medical server and R_{ran} is the random number. It is our assumption that an user chooses low entropy $\langle ID_i, PW_i \rangle$ which are guessable individually in polynomial time.

Login phase

After completing registration procedure successfully, the U_i can access the medical server at anytime from anywhere through a card reader or terminal device which is connected to the medical server. All the steps of this phase are presented below:

Step 1: The U_i primarily inserts his/her smart card into the card reader device and inputs biometric template T_i to the specific sensor device. The card reader computes $F_i^* = H(T_i)$ and matches it with the stored F_i. If it matches, biometric verification passes successfully and asks to input $\langle ID_i, PW_i \rangle$ to the U_i; otherwise, aborts the connection.

Step 2: The card reader computes $A_i^* = h(ID_i \parallel PW_i)$ and matches it with the stored A_i. The matching result ensures whether the U_i has provided valid $\langle ID_i, PW_i \rangle$ or not. If it matches, continues the next step; otherwise, aborts the connection.

Step 3: In this step, the terminal generates a random nonce r_i and computes $C_1 = r_i \cdot P$, $W = B_i \oplus h(ID_i \parallel A_i^*)$, $C_2 = r_i \oplus W$, $C_4 = h(ID_i \parallel r_i \parallel W)$ and transmits $\langle C_2, C_4, CID_i \rangle$ to the medical server as a login message through public/open channel.

Authentication and key agreement phase

The main aim of this phase is to achieve mutual authentication and to share session key agreement between the U_i and the medical server. All the steps of this phase are presented below:

Step 1: Based on the received login message, the medical server first decrypts CID_i using server's secret key x and gets $(ID_i^* \parallel R_{ran}) = DEC_x(CID_i)$, $W = h(ID_s \parallel x \parallel ID_i^*)$, $r_i^* = C_2 \oplus W$, $C_1^* = r_i^* \cdot P$, $C_4^* = h(ID_i \parallel r_i^* \parallel W)$ and matches C_4^* with the received C_4. If it matches, the medical server believes the authenticity of the U_i.

Step 2: The medical server generates a random nonce r_j and computes $D_1 = r_j \cdot P$, $SK = r_j \cdot C_1^* = r_j \cdot r_i^* \cdot P$, $G_1 = D_1 + C_1^*$, $L_i = h(ID_i^* \parallel h_1(D_1) \parallel W)$, $CID_i' = ENC_x(ID_i^* \parallel R_{ran}')$ and sends reply message $\langle L_i, G_1, CID_i' \rangle$ to the U_i through public channel, where R_{ran}' is the random number generated by the medical server.

Step 3: Based on the received reply message, the U_i computes $D_1^* = G_1 - C_1^*$, $L_i^* = h(ID_i \parallel h_1(D_1^*) \parallel W)$, $SK = r_i \cdot D_1^* = r_i \cdot r_j \cdot P$ and matches L_i^* with the received L_i. If it matches, the U_i believes the authenticity of the medical server and the protocol achieves the mutual authentication property and shares a common secret session key SK. After performing mutual authentication, the U_i replaces old $\langle CID_i \rangle$ with the new $\langle CID_i' \rangle$ into the memory of smart card. Finally, the U_i computes $Z_i = h(ID_i \parallel SK)$ and forwards it to the medical server through public channel.

Step 4: After receiving it, the server computes $Z_i^* = h(ID_i^* \parallel SK)$ and matches it with the received Z_i. If it matches, both parties start secure communication.

Password change phase

In any password based user authentication scheme, it is a good property for designing password change phase to provide to change the password facility efficiently without the help of the medical server. Initially, the U_i inserts the smart card to the card reader and executes step-1,2 of the login phase for the authenticity of the U_i. After that, the card reader executes the following steps for changing the password successfully.

Step 1: After user authentication, the card reader asks to input new password PW_i^{new} to the U_i and after entering it, the card reader computes $A_i^{new} = h(ID_i \parallel PW_i^{new})$, $B_i^{new} = h(ID_i \parallel A_i^{new}) \oplus W$, where W is the old parameter and then the card reader replaces $\langle A_i, B_i \rangle$ with the new values $\langle A_i^{new}, B_i^{new} \rangle$ respectively and completes the password change phase successfully.

Cryptanalysis of the proposed scheme

This section cryptanalyses the proposed authentication scheme based on the assumptions mentioned in the attacker model. Firstly, we have demonstrated that the proposed protocol provides well security protection on all the confidential information of the user and server presented in Theorem 1. Secondly, the informal cryptanalysis ensures that all the relevant secure attacks are well protected. Thirdly, the *AVISPA* simulator tool confirms that the proposed authentication protocol is *SAFE* against active and passive attacks including replay and man-in-the-middle attacks.

Theorem 1 *Under the assumptions that the attacker knows smart card parameters* $\langle F_i, B_i, A_i, CID_i, h(), H() \rangle$*, login message* $\langle C_2, C4, CID_i \rangle$ *and reply* $\langle L_i, G_1, CID_i' \rangle$ *from the proposed scheme description. The proposed scheme is secure against the attacker for deriving or guessing* $\langle ID_i, PW_i, T_i \rangle$ *of a legal user, secret key* $\langle x \rangle$ *of the medical server and the session key* $\langle SK \rangle$ *between user and medical server.*

Proof In this proof, we assume that the attacker is able to derive or guess secret parameters $\langle ID_i, PW_i, T_i, x \rangle$ including the session key $\langle SK \rangle$ from the proposed authentication scheme. The justifications of the proof are presented below:

(1) The attacker primarily tries to retrieve confidential information $\langle ID_i, PW_i, T_i, x \rangle$ from the smart card knowledge $\langle F_i, B_i, A_i, CID_i, h(), H() \rangle$, where $F_i = H(T_i)$, $A_i = h(ID_i \parallel PW_i)$, $B_i = h(ID_i \parallel A_i) \oplus W$, $W = h(ID_s \parallel x \parallel ID_i)$ and $CID_i = ENC_x(ID_i \parallel R_{ran})$. The parameter F_i is protected by the bio-hashing operation which produces a bio-code. It should be noted that the bio-hashing operation is one-way and secure as cryptographic one-way hash function. Therefore, the attacker cannot gain any sensitive information from the bio-code. As the biometric template is very high entropy parameter, the adversary cannot guess it in polynomial time. The another smart card parameter A_i relies on the $\langle ID_i, PW_i \rangle$ and protected by the non-invertible cryptographic one-way hash function. Therefore, the attacker cannot retrieve $\langle ID_i, PW_i \rangle$ from A_i. In order to guess the correct $\langle ID_i, PW_i \rangle$ from A_i at a time, the probability is approximately $\frac{1}{2^{12n}}$, which is very negligible. Similarly, the parameter B_i is protected by the one-way hash function and relies on the secret values $W = h(ID_s \parallel x \parallel ID_i)$. The attacker only knows $\langle A_i, F_i, ID_s \rangle$ from the $B_i = h(ID_i \parallel A_i) \oplus W$ parameter. The attacker cannot extract $\langle ID_i, x \rangle$ from the B_i because of non-invertible one-way hash function. To guess the secret parameters $\langle ID_i, x \rangle$, the probability is approximately $\frac{1}{2^{6n+1024}}$, which is negligible.

(2) In each authentication cycle, the login parameters $\langle C_2, C_4, CID_i \rangle$ is transmitted through insecure channel, where $C_2 = r_i \oplus W$, $C_4 = h(ID_i \parallel r_i \parallel W)$ and $CID_i = ENC_x(ID_i \parallel R_{ran})$. As mentioned above, the parameter W is secure against the attacker. Therefore, the attacker cannot extract random nonce r_i or W from C_2. Similarly, the attacker cannot extract ID_i from C_4, as the parameter C_4 is protected by the non-invertible one-way hash function. Since the parameter C_4 relies on the $\langle r_i, W \rangle$, the attacker cannot verify the guessed identity using C_4. If the attacker wants to guess ID_i from C_4, the probability is approximately $\frac{1}{2^{6n+160+1024}}$, which is very negligible.

(3) In each execution of the proposed protocol, the reply message $\langle L_i, G_1, CID_i' \rangle$ is transmitted over the insecure channel, where $L_i = h(ID_i^* \parallel h_1(D_1) \parallel W)$, $G_1 = D_1 + C_1^*$ and $CID_i = ENC_x(ID_i \parallel R_{ran}')$. The computation of L_i parameter relies on the confidential parameter $\langle ID_i, D_1, W \rangle$ and also all these parameter are protected by the cryptographic one-way hash function. Therefore, the attacker cannot extract ID_i from L_i and also the guessing probability would be very negligible.

(4) The session key of the proposed protocol $SK = r_i \cdot r_j \cdot P$ relies on the random nonces $\langle r_i, r_j \rangle$ which are secure throughout our scheme. Additionally, the SK is protected by the elliptic curve discrete logarithm problem *(ECDLP)*. Therefore, the attacker cannot compute SK without the knowledge of $\langle r_i, r_j \rangle$. In order to guess the session key, the probability is approximately $\frac{1}{2^{320}}$, which is very negligible and infeasible in polynomial time.

The above explanations clearly state that the attacker has no way to extract the confidential parameters of the user and server including session key of the proposed protocol. Additionally, it also shows that the guessing probability is negligible. Therefore, our assumption is wrong and proof the Theorem 1. □

Off-line identity-password guessing attack

In our proposed scheme, we have assumed that each users use very low entropy ID_i and PW_i which can be guessed by the attacker in polynomial time. However, the attacker cannot successfully verify the guessed ID_i or PW_i in off-line mode by using the extracted parameters $\langle F_i, B_i, A_i, CID_i, h(), H() \rangle$ from the stolen smart card and intercepted parameters $\langle C_2, C4, CID_i, L_i, G_1 \rangle$ from the login-reply messages. We have presented below clear

justifications for resilience of off-line identity-password guessing attacks.

(1) In order to verify the guessed parameters ID_i^* and PW_i* by using the parameter $A_i = h(ID_i \parallel PW_i)$, the attacker has to guess correctly two unknown parameters $\langle ID_i, PW_i \rangle$ at a time, which is infeasible in polynomial time.
(2) To Verify the guessed identity ID_i^* by using the condition $B_i^* = h(ID_i^* \parallel A_i) \oplus h(ID_s \parallel x \parallel ID_i) = B_i$, the attacker needs to know the secret key x of the server. It is noted that the secret key x is only known to the attacker.
(3) Similarly, the attacker needs to know x parameters to verify the guessed identity ID_i^* by using CID_i, where $CID_i = ENC_x(ID_i \parallel R_{ran})$.
(4) The attacker also cannot verify the guessed identity ID_i^* by using the parameters $\langle C_4, L_i \rangle$ because of secret parameters $\langle r_i, W, D_1 \rangle$.

The above justifications clearly state that the attacker cannot verify the guessed ID_i^* and PW_i^*. Therefore, the proposed scheme is secure against off-line password guessing attack.

User anonymity and untraceability attack

We have explained in off-line identity-password guessing attacks that the attacker cannot extract or guess user identity from protocol description and known parameters of the attacker. Therefore, we can claim that the proposed protocol preserves user anonymity property. During execution of the proposed protocol, the parameter $CID_i = ENC_x(ID_i \parallel R_{ran})$ is transmitted as a login message. After receiving CID_i, the medical server extracts ID_i after decrypting CID_i with the help of the server's secret key x. After performing mutual authentication, the server sends $CID_i' = ENC_x(ID_i \parallel R_{ran}')$ to the U_i, and after getting it, the smart card updates CID_i' in each authentication cycle. As the parameter CID_i is not static, it is our argue that the attacker cannot trace the U_i from the login message.

Privileged insider attack

Most of the today's security system does not guarantee high reliability due to insider attack. It is practical that most of the users use identical password for accessing a set of application servers. If an insider of the system such as system manager or administrator leaks the user's confidential information i.e. password (PW_i) to the attacker, he/she may use that password to the others accounts of the others servers. Therefore, the confidentiality on the user's password from the server is very necessary, though the server is trusted. At the time of registration, the proposed protocol sends $A_i = h(ID_i \parallel PW_i)$ instead of plaintext password PW_i to the server. Therefore, the insider person cannot extract PW_i from A_i due to non-invertible one-way hash function.

User impersonation attack

In this attack, we suppose that the attacker eavesdrops the login message of the proposed scheme $\langle C_2, C_4, CID_i \rangle$ and tries to impersonate as a legitimate user. However, the attacker cannot impersonate as a legitimate user due to the following reasons:

(1) It is clear that the attacker can generate a random number r_i^* and compute $C_1 = r_i^* \cdot P$, as the parameter P is public.
(2) To compute valid C_2, where $C_2 = r_i \oplus W$, the attacker needs valid $W = h(ID_s \parallel x \parallel ID_i)$ which is not feasible.
(3) The parameter $C_4 = h(ID_i \parallel h(C_1) \parallel W)$ relies on the secret parameter W and the identity ID_i. Therefore, the attacker cannot compute valid C_4 parameter.

The above explanations show that the proposed scheme resists user impersonation attack i.e. the attacker cannot impersonate as a legitimate user.

Server impersonation attack

Like user impersonation attack, the attacker may also try to impersonate as a legitimate server after generating valid reply message $\langle L_i^*, G_1^*, CID_i' \rangle$ of the proposed scheme. However, the proposed scheme provides strong security protection against server impersonation attack due to following reasons:

(1) The parameter G_1 is defined as $G_1 = D_1 + C_1^*$, where $D_1 = r_j \cdot P$ and $C_1^* = r_i^* \cdot P$.
(2) It is confirmed that the attacker can compute $D_1^* = r_j^* \cdot P$ after generating a random number r_j^*.
(3) To compute valid G_1, the attacker needs to compute valid C_1^* which is dependent on the random number r_i. As the attacker cannot compute valid r_i from C_2 parameter due to secret value W, he/she is unable to compute valid G_1.
(4) Similarly, the attacker cannot compute valid $L_i = h(ID_i \parallel h_1(D_1) \parallel W)$ due to unknown secret parameters $\langle ID_i, W \rangle$.

The above explanations show that the proposed scheme is secure against server impersonation attack.

Smart card theft attack

It is a very critical attack on any smart card based user authentication and key agreement protocol. If the attacker

can compute the valid registration parameters without altering server's information and identity of the user, s/he may introduce a new smart card by using his/her own biometric template and password. However, the attacker cannot launch smart card theft attack which is justified below:

(1) The smart card holds $\langle F_i, B_i, A_i, CID_i, h(), H()\rangle$ of a legal user where, $F_i = H(T_i)$, $A_i = h(ID_i \parallel PW_i)$, $B_i = h(ID_i \parallel A_i) \oplus W$, $W = h(ID_s \parallel x \parallel ID_i)$, $CID_i = ENC_x(ID_i \parallel R_{ran})$.
(2) It is confirmed that the attacker can compute $F_i^a = H(T_i^a)$ by utilizing his/her biometric template T_i^a and can embed into the smart card.
(3) As the valid user password PW_i is hashed with the valid ID_i in A_i, the attacker cannot embed attacker's password PW_i^a without the knowledge of valid ID_i. Therefore, the computation $A_i^a = h(ID_i \parallel PW_i^a)$ is not feasible.
(4) Similarly, the attacker cannot compute valid B_i without the knowledge of secret parameters $\langle ID_i, W\rangle$.

The above descriptions confirm that the proposed authentication scheme is secure against smart card theft attack.

Smart card stolen attack

We supposed that the attacker has got the legal user smart card by some means and extracted smart card information $\langle F_i, A_i, B_i, CID_i, h(), H()\rangle$, where $F_i = H(T_i)$, $A_i = h(ID_i \parallel PW_i)$, $B_i = h(ID_i \parallel A_i) \oplus W$, $W = h(ID_s \parallel x \parallel ID_i)$ and $CID_i = ENC_x(ID_i \parallel R_{ran})$. It is noted that the bio-hashing is secure as cryptographic one-way hash function and non-invertible. Therefore, the attacker cannot extract T_i from F_i. Moreover, s/he cannot guess it in polynomial time due to high entropy property. The parameter A_i is protected by the one-way hash function and hence cannot extract $\langle ID_i, PW_i\rangle$ from A_i. Additionally, they (attacker) cannot guess low entropy $\langle ID_i, PW_i\rangle$ from A_i as suggested in [48]. The attacker is not able to guess or extract $\langle ID_i\rangle$ from B_i, CID_i due to unknown parameter $\langle x, R_{ran}\rangle$. Therefore, the proposed authentication scheme provides strong security protection on smart card stolen attack.

Session key computation attack

The security of the session key $SK = r_i \cdot r_j \cdot P$ of our proposed protocol relies on the difficulty of elliptic curve discrete logarithm problem. We have shown earlier that the attacker has no way to extract $\langle r_i, r_j\rangle$ parameters from the protocol description. As the computation of the session key depends upon the two secret number $\langle r_i, r_j\rangle$, the attacker is unable to compute it. Therefore, the proposed scheme provides strong security protection on the session key.

Session key verification

In the step 3-4 of the authentication phase, the user sends $Z_i = h(ID_i \parallel SK)$ to the medical server and upon receiving it, the server checks the verification whether $Z_i^* = h(ID_i^* \parallel SK) =? Z_i$. If it is correct, it ensures that the session key SK is verified. Therefore, the proposed scheme provides session key verification property.

Efficient login and password change phase

In the login phase of our scheme, the smart card generates and transmits the login message after verifying the user's biometric template and password every time. Therefore, the login phase reduces extra computation as well as network congestion. Similarly, the smart card verifies the authenticity of the user before updating the new password. It may be noted that the user can choose and update the password at his will without the help of the medical server which reduces communication cost as well as computation cost.

Message freshness

Timestamp method is the another way for resisting replay attack. However, this method may sometimes suffer from clock synchronization problem. To overcome it, the authentication scheme should maintain global clock time i.e. the

```
role alice (Ui, Sj : agent,
SKuisj : symmetric_key,
% H is hash function
H,Add,Sub,Mul : hash_func, Snd, Rcv: channel(dy))
played_by Ui
def=
local State : nat,
IDi, IDs, PWi, Ti, Bi, X, Fi, CIDi, CIDin, R, Ai, Ri,Rs, P: text,
W, C2, C4, Li, G1, D1, Ki, M1, Zi, SK: message,
Inc : hash_func
const alice_bob_ri, bob_alice_rs,
subs1, subs2, subs3 : protocol_id
init State :=0
transition
1. State = 0 /\ Rcv(start) =|>
State' := 1 /\ Ai' := H(IDi.PWi)
/\ Snd({IDi.Ai'.Ti}_SKuisj)
/\ secret({PWi, Ti}, subs2, Ui)
/\ secret({IDi}, subs3, {Ui,Sj})
2. State = 1 /\ Rcv({Fi'. H(IDi.PWi). Bi'. CIDi'}_SKuisj) =|>
State' := 2 /\ Ri' := new()
/\ W' := xor(Bi', H(IDi.X))
/\ C2' := xor(Ri',W')
/\ C4' := H(IDi.Ri'.W')
/\ Snd(C2',C4',CIDi')
/\ witness(Ui, Sj, alice_bob_ri, Ri')
3. State = 2 /\ Rcv(Li',G1'. CIDin') =|>
State' := 3 /\ Ri' := new()
/\ M1' := Sub(G1',Mul(Ri'.P))
/\ SK' := Mul(Ri'. M1')
/\ Zi' := H(IDi.SK')
/\ Snd(Zi')
/\ request(Sj, Ui, bob_alice_rs, M1')
end role
```

Fig. 4 Role specification for the user *(Ui)* of the proposed scheme in *HLPSL*

user and the medical server should maintain same time, which requires extra cost of the protocol. For avoiding this problem, our proposed protocol uses random nonces instead of timestamp to verify the freshness of the message.

Simulation for formal security verification using *AVISPA* tool

This section discusses regarding the simulation of our proposed scheme for the formal security verification using the widely-accepted *AVISPA* [2, 6, 12] (Automated Validation of Internet Security Protocols and Applications) tool for proving the proposed protocol is secure against passive and active attacks including the replay and man-in-the-middle attacks.

Specifying the proposed protocol

In this section, we discuss briefly the specification of the proposed scheme using *HLPSL* language for the roles of the user, server, session and the environment. In Fig. 4, we have implemented the role for the U_i. During the registration phase, the U_i initially transmits $\langle ID_i, A_i, F_i \rangle$ to the medical server through secure channel with the help of the *Snd()* operation and symmetric key. The type declaration channel(dy) means that the channel is for the Dolev-Yao [14] threat model. The declaration *secret(PWi, Ti, subs2, Ui)* indicates that the parameters $\langle PW_i, T_i \rangle$ are only known to U_i and similarly *secret(IDi, subs3, Ui,S)* tells that the ID_i is kept secret permanently to both U_i and S. U_i then receives a smart card information

```
role bob (Ui, Sj : agent,
SKuisj : symmetric_key,
H,Add,Sub,Mul : hash_func,
Snd, Rcv: channel(dy) )
played_by Sj
def=
local State : nat,
IDi, IDs, PWi, Ti, Bi, X, Fi, CIDi, CIDin, Rn, R, Ai, Ri, Rs,P: text,
W, C2, C4, Li, G1, M1, Ki, D1, Zi, SK: message,
Inc : hash_func
const alice_bob_ri, bob_alice_rs,
subs1, subs2, subs3 : protocol_id
init State := 0
transition
1. State = 0 /\ Rcv({IDi.H(IDi.PWi).Ti}_SKuisj) =|>
State' := 1 /\ R' := new()
/\ secret({X}, subs1, Sj)
/\ secret({PWi, Ti}, subs2, Ui)
/\ secret({IDi}, subs3, {Ui,Sj})
/\ Fi' := H(Ti)
/\ W' := H(IDs.X. IDi)
/\ Bi':= xor(H(IDi.Ai),W')
/\ CIDi' := {IDi.R'}_X
/\ Snd({Fi'. H(IDi.PWi). Bi'. CIDi'}_SKuisj)
2. State = 1 /\ Rcv(C2'.C4'.CIDi') =|>
State' := 2 /\ Rs' := new()
/\ Rn' := new()
/\ IDi' := {CIDi'}_X
/\ W' := H(IDs.X.IDi')
/\ D1' := Mul(Rs',P)
/\ Ki' := xor(C2', W')
/\ G1' := Add(D1', Mul(Ki'.P))
/\ Li' := H(IDi'.H(D1').W')
/\ CIDin' := {IDi'.Rn'}_X
/\ Snd (Li'.G1'.CIDin')
/\ witness(Sj, Ui, bob_alice_rs, Rs')
3. State = 2 /\ Rcv(Zi') =|>
State' := 3 /\ Ri' := new()
/\ request(Ui, Sj, alice_bob_ri, Ri')
end role
```

Fig. 5 Role specification for the bob (S) of the proposed scheme in *HLPSL*

```
role session(Ui, Sj: agent,
SKuisj : symmetric_key,
H,Add,Sub,Mul : hash_func)
def=
local SI, SJ, RI, RJ: channel (dy)
composition
alice(Ui, Sj, SKuisj, H,Add,Sub,Mul, SI, RI)
/\ bob(Ui, Sj, SKuisj, H, Add, Sub, Mul, SJ, RJ)
end role
role environment()
def=
const ui, sj: agent,
skuisj : symmetric_key,
h,add,sub,mul : hash_func,
idi, ids, pwi, ti, bi, x, fi, ai, ri, rs, p : text,
alice_bob_ri, bob_alice_rs,
subs1, subs2, subs3: protocol_id
intruder_knowledge = {ui, sj, h, add, sub, mul,
Fi, Ai, CIDi, Bi, C2, C4, Li,G1}
composition
session(ui, sj, skuisj, h, add, sub, mul)
/\ session(ui, sj, skuisj, h, add, sub,mul)
end role
goal
secrecy_of subs1
secrecy_of subs2
secrecy_of subs3
authentication_on alice_bob_ri
authentication_on bob_alice_rs
end goal
environment()
```

Fig. 6 Role specification for session, goal and environment of the proposed scheme in *HLPSL*

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfilenktK5E.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.14s
  visitedNodes: 13 nodes
  depth: 4 plies
```

Fig. 7 Simulation result of the proposed scheme for the *OFMC* backend

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfilenktK5E.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed  : 0 states
  Reachable  : 0 states
  Translation: 0.03 seconds
  Computation: 0.00 seconds
```

Fig. 8 Simulation result of the proposed scheme for the *CL-AtSe* back-end

$\langle F_i, A_i, B_i, CID_i h(), H() \rangle$ with the help of the *Rcv()* operation. Thereafter, the U_i generates a random number with the help of the *new()* operation and sends the login message $\langle C_2, C_4, CID_i \rangle$ to the S through public channel. The declaration $witness(Ui, S, alice\_bob\_ri, Ri')$ indicates that the U_i has generated freshly R_i' for the S in the login phase. In the authentication and key agreement phase, the U_i receives reply message $\langle L_i', G_i', CIDin, \rangle$ with the help of the *Rcv()* operation and finally sends $\langle Z_i \rangle$ to the S over public channel for session key verification. The declaration $request(S, Ui, bob\_alice\_rs, Rs')$ means that the U_i authenticates the S.

In Fig. 5, we have implemented the role for the medical server S in *HLPSL* language. After receiving the registration message, the S transmits a smart card with the parameters $\langle F_i, A_i, B_i, CID_i, h(), H() \rangle$ with the help of the $Snd()$ operation through secure channel. The declaration *secret(X, subs1, S)* states that the secret key is kept secret permanently to the medical server. In login phase, the server receives $\langle C_2, C_4, CID_i \rangle$ from the U_i through insecure channel. After that, the S generates a random nonce with the help of the *new()* operation and transmits a reply message $\langle L_i', G_i', CIDin \rangle$ to the U_i over a public channel. Finally, the S receives a message $\langle Z_i \rangle$ for session key verification with the help of the $Rcv()$ operation. The declaration $request\ (Ui, S, alice\_bob\_ri, Ri')$ states that the S authenticates U_i.

In Fig. 6, we have provided the specification for the roles of session, goal and environment in *HLPSL*. In the session segment, all the basic roles including the roles for the U_i and the S are instanced with concrete arguments. The environment section contains the global constant and composition of one or more session and the intruder knowledge is also given. The current version (2006/02/2013) of *HLPSL* supports the standard authentication and secrecy goals. In our implementation, the following three secrecy goals and two authentications are verified.

- secrecy_of subs 1: It represents the secret key X_s is kept secret to medical server only.
- secrecy_of subs 2: It represents the secret parameters $\langle PW_i, T_i \rangle$ is kept secret only to U_i.
- secrecy_of subs 3: It represents the secret parameter ID_i is kept secret only to U_i and S
- authentication_on bob_alice_ri: It means that the U_i generates a random nonce r_i where r_i is only known to U_i and if S receives it through message securely, S then authenticates the U_i.
- authentication_on bob_alice_rs: It means that the S generates a random nonce r_s where r_s is only known to S

Table 2 Security functionality comparison of the proposed scheme with related schemes

Schemes ⇒	[10]	[12]	[51]	[56]	[11]	[9]	[39]	[53]	[45]	[54]	proposed
A1	×	×	×	×	√	√	√	√	×	√	√
A2	×	×	√	√	√	√	√	√	×	√	√
A3	×	×	×	√	√	×	√	×	√	√	√
A4	×	×	√	√	√	√	√	√	√	√	√
A5	√	√	√	√	√	×	√	√	√	√	√
A6	×	√	√	−	−	√	×	×	√	×	√
A7	×	×	√	√	√	√	√	√	×	√	√
A8	*NA*	√	√	×	√	√	√	√	√	√	√
A9	×	√	×	×	√	×	×	×	√	×	√
A10	×	√	×	×	√	×	×	×	√	×	√

A1: Resist user anonymity, A2: Smart card stolen attack, A3: Resist off-line password guessing attack A4: Resist insider attack, A5: Resist replay attack, A6: Session key verification, A7: Mutual authentication A8: Session key agreement A9: Efficient login phase, A10: Efficient and user friendly password change phase ×: Cannot resist corresponding attack or cannot achieve corresponding features, √: Resists corresponding attack or achieves corresponding property, NA: Not Applicable feature

Table 3 Computation, communication and storage cost comparison of the proposed scheme with related schemes

Schemes ⇒	Cheng [10]	Das [12]	Tan [49]	Yan [55]	Mishra [45]	Xu [54]	Proposed
SCSC	960	1280	640	640	800	1120	960
CC	960	1280	800	960	1120	1120	1120
CCLP	$3T_h$	$5T_h$	$3T_h + 1T_S$	$3T_h$	$4T_h$	$3T_h + 2\ T_{pm}$	$4T_h + 1\ T_{pm}$
CCAP	$7T_h$	$13T_h$	$8T_h + 1\ T_S$	$8T_h$	$10T_h + 2\ T_S$	$11T_h + 4T_{pm}$	$7T_h + 2T_s + 4\ T_{pm}$
TC in sec.	0.005	0.009	0.0299	0.0055	0.0244	0.38545	0.1099

SCSC: Smart card storage cost, *CC*: Communication cost, *CCLP*: Computation cost for login phase, *CCAP*: Computation cost for authentication phase, *TC* in sec: Total cost in second

and if U_i receives it through message securely, U_i then authenticates the S.

Simulation results

In this section, we specify simulation results of our proposed scheme based on the widely-accepted two back-ends such as *OFMC* and *CL-AtSe* using the *AVISPA* web tool [50]. The Figs. 7 and 8 confirm that the proposed protocol is *SAFE* under two back-ends *OFMC* and *CL-AtSe* respectively. Moreover, the simulation results using *AVISPA* clearly demonstrates that the proposed scheme is secure against active and passive attacks including replay and man-in-the-middle attacks.

Performance evaluation

The computation and communication cost complexities are the most important issues to measure the performance of any user authentication and key agreement protocol and it should be as minimum as possible than the existing related schemes for achieving better performance. This section evaluates the performance comparison of the proposed protocol with some other existing related protocols. In Table 2, we have presented security functionality comparison of the proposed protocol with other existing related protocols and it has been observed that none of the protocols are completely free from security weaknesses. However, the proposed protocol not only protects security attacks described in the Table 2, but also protects user-server impersonation attack, smart card theft attack, smart card stolen and achieves efficient login and password update phase and session key verification property. In order to measure the computation cost, this paper mainly uses symmetric key encryption/decryption operation T_s, cryptographic hash function (T_h), point multiplication (T_{pm}) operation, xor ($\oplus$) and concatenation ($\|$). As the operations $\langle \oplus, \| \rangle$ take very negligible computation cost than $\langle T_h, T_{pm,T_s} \rangle$, we avoid it in our comparison. According to [36, 37], the computation cost complexity can be roughly expressed as ($T_{pm} >> T_s > T_h$). As suggested in [36, 37], we have assumed that the computation cost for one-way hash function, symmetric key encryption decryption algorithm and elliptic curve scalar point multiplication operation take *0.0005*, *0.0087* and *0.063075* second respectively. The Table 3 presents the computation and communication cost comparison of the proposed protocol with some other related existing protocols. The Table 3 clearly indicates that the execution time of the proposed protocol is better than scheme [54]. However, the proposed scheme takes relatively more computation than the schemes [10, 12, 45, 49, 55], because these schemes are based on the light wight cryptographic hash function. Although the proposed protocol provides high security protection on the relevant security attacks mentioned in Table 2, the Table 3 ensures that the protocol is relatively better than existing related protocols in terms of smart card storage and computation cost.

Conclusion and future work

In this paper, we have analyzed that both (Mishra et al. and Xu et al.) protocols suffer from several security weaknesses. Thereafter, we have proposed a more efficient and secure authentication protocol to fix the Mishra and Xu et al.'s security weaknesses. The proposed scheme satisfies all the desirable security attributes which are presented in the security analysis section of this paper through both formal and informal security analysis. We have simulated our proposed scheme for the formal security verification using the widely-accepted *AVISPA* tool and shown that the proposed protocol is secure against passive and active attacks including the replay and man-in-the-middle attacks. The performance analysis confirms that the proposed protocol is efficient as compared to other related existing schemes in terms of computation and smart card storage overhead. The proposed scheme supports efficient login and

authentication phase, password change phase and achieves mutual authentication as well as session key agreement and verification between the user and the medical server. Considering the security and efficiency provided by the proposed scheme, we conclude that the proposed scheme is more appropriate for practical application for telecare medical information system. Further, we aim to reduce the complexities of the proposed protocol without compromising security.

References

1. Amin, R., Cryptanalysis and an efficient secure id-based remote user authentication using smart card. *Int. J. Comput. Appl.* 75(13):43–48, 2013.
2. Amin, R., and Biswas, G., A novel user authentication and key agreement protocol for accessing multi-medical server usable in tmis. *J. Med. Syst.* 39(3):33, 2015. doi:10.1007/s10916-015-0217-3.
3. Amin, R., and Biswas, G., Remote access control mechanism using rabin public key cryptosystem. In: Information Systems Design and Intelligent Applications, Advances in Intelligent Systems and Computing, Vol. 339, pp. 525–533. India: Springer, 2015. doi:10.1007/978-81-322-2250-7_52.
4. Amin, R., and Biswas, G.P., Anonymity preserving secure hash function based authentication scheme for consumer usb mass storage device. In: Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on, pp. 1–6, 2015. doi:10.1109/C3IT.2015.7060190.
5. Amin, R., Maitra, T., Rana, S.P., An improvement of wang. et. al.'s remote user authentication scheme against smart card security breach. *Int. J. Comput. Appl.* 75(13):37–42, 2013.
6. Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuellar, J., Drielsma, P., Hem, P., Kouchnarenko, O., Mantovani, J., Mdersheim, S., von Oheimb, D., Rusinowitch, M., Santiago, J., Turuani, M., Vigan, L., Vigneron, L., The avispa tool for the automated validation of internet security protocols and applications. In: Computer Aided Verification, Lecture Notes in Computer Science, Vol. 3576, pp. 281–285, 2005.
7. Arshad, H., and Nikooghadam, M., Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. *J. Med. Syst.* 38(12):1–12, 2014. doi:10.1007/s10916-014-0136-8.
8. Awasthi, A., and Srivastava, K., A biometric authentication scheme for telecare medicine information systems with nonce. *J. Med. Syst.* 37(5):9964, 2013. doi:10.1007/s10916-013-9964-1.
9. Cao, T., and Zhai, J., Improved dynamic id-based authentication scheme for telecare medical information systems. *J. Med. Syst.* 37(2):9912, 2013. doi:10.1007/s10916-012-9912-5.
10. Chang, Y.F., Yu, S.H., Shiao, D.R., A uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care. *J. Med. Syst.* 37(2):9902, 2013. doi:10.1007/s10916-012-9902-7.
11. Chen, H.M., Lo, J.W., Yeh, C.K., An efficient and secure dynamic id-based authentication scheme for telecare medical information systems. *J. Med. Syst.* 36(6):3907–3915, 2012.
12. Das, A.K., and Goswami, A., A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care. *J. Med. Syst.* 37(3):9948, 2013. doi:10.1007/s10916-013-9948-1.
13. Debiao, H., Jianhua, C., Rui, Z., A more secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(3):1989–1995, 2012.
14. Dolev, D., and Yao, A.C., On the security of public key protocols. *IEEE Trans. Inf. Theory* 29(2):198–208, 1983.
15. Giri, D., Maitra, T., Amin, R., Srivastava, P., An efficient and robust rsa-based remote user authentication for telecare medical information systems. *J. Med. Syst.* 39(1):145, 2014. doi:10.1007/s10916-014-0145-7.
16. Hafizul Islam, S., and Biswas, G., Dynamic id-based remote user mutual authentication scheme with smartcard using elliptic curve cryptography. *J. Electron. (China)* 31(5):473–488, 2014. doi:10.1007/s11767-014-4002-0.
17. Islam, S., Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps. *Nonlinear Dyn.* 78(3):2261–2276, 2014. doi:10.1007/s11071-014-1584-x.
18. Islam, S.H., Design and analysis of an improved smartcard-based remote user password authentication scheme. *Int. J. Commun. Syst.*, 2014. doi:10.1002/dac.2793.
19. Islam, S.H., and Biswas, G., A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks. *Ann. Telecommun.* 67(11-12):547–558, 2012.
20. Islam, S.H., and Biswas, G.P., A more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *J. Syst. Softw.* 84(11):1892–1898, 2011.
21. Islam, S.H., and Biswas, G.P., Design of improved password authentication and update scheme based on elliptic curve cryptography. *Math. Comput. Model.* 57(11-12):2703–2717, 2013.
22. Islam, S.H., and Biswas, G.P., Design of two-party authenticated key agreement protocol based on ecc and self-certified public keys. *Wirel. Pers. Commun.* 1–24, 2015. doi:10.1007/s11277-015-2375-5.
23. Islam, S.H., and Khan, M.K., Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems. *J. Med. Syst.* 38(10):135, 2014. doi:10.1007/s10916-014-0135-9.
24. Jina, A.T.B., Ling, D.N.C., Goh, A., Biohashing: Two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recogn.* 37(11):2245–2255, 2004.
25. Khan, M.K., and Kumari, S., Cryptanalysis and improvement of an efficient and secure dynamic id-based authentication scheme for telecare medical information systems. *Secur. Commun. Netw.* 7(2):399–408, 2014. doi:10.1002/sec.791.
26. Khan, M.K., Kumari, S., Gupta, M., More efficient key-hash based fingerprint remote authentication scheme using mobile device. *Computing* 96(9):793–816, 2014. doi:10.1007/s00607-013-0308-2.
27. Kocher, P., Jaffe, J., Jun, B., Differential power analysis. In: Advances in Cryptology CRYPTO 99, Lecture Notes in Computer Science, Vol. 1666, pp. 388–397, 1999.
28. Kumar, M., Gupta, M.K., Kumari, S., An improved efficient remote password authentication scheme with smart card over insecure networks. *Int. J. Netw. Secur.* 13(3):167–177, 2011.
29. Kumari, S., Gupta, M.K., Khan, M.K., Li, X., An improved timestamp-based password authentication scheme: comments, cryptanalysis, and improvement. *Secur. Commun. Netw.* 7:1921–1932, 2014. doi:10.1002/sec.906.
30. Kumari, S., and Khan, M.K., More secure smart card based remote user password authentication scheme with user anonymity. *Secur. Commun. Netw.* 7:2039–2053, 2013. doi:10.1002/sec.916.
31. Kumari, S., and Khan, M.K., Cryptanalysis and improvement of 'a robust smart-card-based remote user password authentication scheme. *Int. J. Commun. Syst.* 27:3939–3955, 2014. doi:10.1002/dac.2590.

32. Kumari, S., Khan, M.K., Kumar, R., Cryptanalysis and improvement of a privacy enhanced scheme for telecare medical information systems. *J. Med. Syst.* 37(4):9952, 2013. doi:10.1007/s10916-013-9952-5.
33. Kumari, S., Khan, M.K., Li, X., An improved remote user authentication scheme with key agreement. *Comput. Electr. Eng.* 40(6):1997–2012, 2014. doi:10.1016/j.compeleceng.2014.05.007.
34. Kumari, S., Khan, M.K., Li, X., Wu, F., Design of a user anonymous password authentication scheme without smart card. *Int. J. Commun. Syst.* 27(10):609–618, 2014. doi:10.1002/dac.2853.
35. Lee, T.F., Chang, I.P., Lin, T.H., Wang, C.C., A secure and efficient password- based user authentication scheme using smart cards for the integrated epr information system. *J. Med. Syst.* 37(3):1–7, 2013.
36. Li, C.T., Hwang, M.S., Chu, Y.P., A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks. *Comput. Commun.* 31(12):2803–2814, 2008.
37. Li, W., Wen, Q., Su, Q., Jin, Z., An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network. *Comput. Commun.* 35(2):188–195, 2012.
38. Li, X., Niu, J.W., Ma, J., Wang, W.D., Liu, C.L., Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. *J. Netw. Comput. Appl.* 34(1):73–79, 2011.
39. Lin, H.Y., On the security of a dynamic id-based authentication scheme for telecare medical information systems. *J. Med. Syst.* 37(2):1–5, 2013.
40. Liping, Z., and Shaohui, Z., Robust ecc-based authenticated key agreement scheme with privacy protection for telecare medicine information systems. *J. Med. Syst.* 39(5), 2015. doi:10.1007/s10916-015-0233-3.
41. Lu, Y., Li, L., Peng, H., Yang, Y., An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. *J. Med. Syst.* 39(3):32, 2015. doi:10.1007/s10916-015-0221-7.
42. Lumini, A., and Nanni, L., Biohashing: Two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recogn.* 40(3):1057–1065, 2007.
43. Messerges, T.S., Dabbish, E.A., Sloan, R.H., Examining smartcard security under the threat of power analysis attacks. *IEEE Trans. Comput.* 51(5):541–552, 2002.
44. Mishra, D. *A study on id-based authentication schemes for telecare medical information system*: CoRR, 2013. arXiv:1311.0151.
45. Mishra, D., Mukhopadhyay, S., Chaturvedi, A., Kumari, S., Khan, M., Cryptanalysis and improvement of Yan et al.'s biometric-based authentication scheme for telecare medicine information systems. *J. Med. Syst.* 38(6):24, 2014. doi:10.1007/s10916-014-0024-2.
46. Mishra, D., Mukhopadhyay, S., Kumari, S., Khan, M., Chaturvedi, A., Security enhancement of a biometric based authentication scheme for telecare medicine information systems with nonce. *J. Med. Syst.* 38(5):41, 2014. doi:10.1007/s10916-014-0041-1.
47. Sarvabhatla, M., Giri, M., Vorugunti, C.S. *Cryptanalysis of cryptanalysis and improvement of Yan et al. biometric-based authentication scheme for TMIS*: CoRR, 2014. arXiv:1406.3943.
48. Sood, S.K., Sarje, A.K., Singh, K., A secure dynamic identity based authentication protocol for multi-server architecture. *J. Netw. Comput. Appl.* 34(2):609–618, 2011. Efficient and Robust Security and Services of Wireless Mesh Networks.
49. Tan, Z., An efficient biometrics-based authentication scheme for telecare medicine information systems. *Network* 2(3):200–204, 2013.
50. Tool, A.W.: http://www.avispa-project.org/web-interface/, 2015.
51. Wei, J., Hu, X., Liu, W., An improved authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(6):3597–3604, 2012.
52. Wu, Z.Y., Lee, Y.C., Lai, F., Lee, H.C., Chung, Y., A secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(3):1529–1535, 2012.
53. Xie, Q., Zhang, J., Dong, N., Robust anonymous authentication scheme for telecare medical information systems. *J. Med. Syst.* 37(2):9911, 2013.
54. Xu, X., Zhu, P., Wen, Q., Jin, Z., Zhang, H., He, L., A secure and efficient authentication and key agreement scheme based on ecc for telecare medicine information systems. *J. Med. Syst.* 38(6):24, 2014. doi:10.1007/s10916-013-9994-8.
55. Yan, X., Li, W., Li, P., Wang, J., Hao, X., Gong, P., A secure biometrics-based authentication scheme for telecare medicine information systems. *J. Med. Syst.* 37(5):1–6, 2013.
56. Zhu, Z., An efficient authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(6):3833–3838, 2012.