J Med Syst (2015) 39: 30
DOI 10.1007/s10916-015-0218-2

TRANSACTIONAL PROCESSING SYSTEMS

A Secure User Anonymity-Preserving Three-Factor Remote User Authentication Scheme for the Telecare Medicine Information Systems

Ashok Kumar Das

Received: 4 December 2014 / Accepted: 26 January 2015 / Published online: 13 February 2015

Abstract Recent advanced technology enables the telecare medicine information system (TMIS) for the patients to gain the health monitoring facility at home and also to access medical services over the Internet of mobile networks. Several remote user authentication schemes have been proposed in the literature for TMIS. However, most of them are either insecure against various known attacks or they are inefficient. Recently, Tan proposed an efficient user anonymity preserving three-factor authentication scheme for TMIS. In this paper, we show that though Tan's scheme is efficient, it has several security drawbacks such as (1) it fails to provide proper authentication during the login phase, (2) it fails to provide correct updation of password and biometric of a user during the password and biometric update phase, and (3) it fails to protect against replay attack. In addition, Tan's scheme lacks the formal security analysis and verification. Later, Arshad and Nikooghadam also pointed out some security flaws in Tan's scheme and then presented an improvement on Tan's s scheme. However, we show that Arshad and Nikooghadam's scheme is still insecure against the privileged-insider attack through the stolen smart-card attack, and it also lacks the formal security analysis and verification. In order to withstand those security loopholes found in both Tan's scheme, and Arshad and Nikooghadam's scheme, we aim to propose an effective and more secure three-factor remote user authentication scheme for TMIS. Our scheme provides the user anonymity property. Through the rigorous informal and formal security analysis using random oracle models and the widely-accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) tool, we show that our scheme is secure against various known attacks, including the replay and man-in-the-middle attacks. Furthermore, our scheme is also efficient as compared to other related schemes.

Keywords Telecare medicine information systems · Fuzzy extractor · Biometrics · Password · User anonymity · AVISPA · Security

Introduction

The rapid development of modern information and communication technologies make people's daily lives much easier worldwide. This has also led to the new circumstances at the all levels of the social environment [59]. Consider a healthcare system, where sensors and datalinks offer potential for constant monitoring of a patient's symptoms and needs. It enables the doctors, nurses and other medical staffs to diagnose and monitor health problems for the patient in real-time, where a patient is either at home or outdoors [36, 45, 57].

In a telecare medical information system (TMIS), patients can send health related information or use portals for health monitoring and healthcare-related services over the Internet or mobile networks. If a patient travels to a hospital, it is desirable that the expense of the patients such as travel cost and the hospitalization time is much. In order to reduce significantly these factors, the patients can easily apply TMISs to access the healthcare delivery services. Since the telecare server keeps the electronic medical

This article is part of the Topical Collection on *Transactional Processing Systems*

A. K. Das
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology,
Hyderabad 500 032, India
e-mail: iitkgp.akdas@gmail.com;ashok.das@iiit.ac.in

records of all registered users in the hospitals, TMIS can help the physicians to make more comprehensive decision via the cooperation of some physicians in different places. Wireless mobile telecommunications of TMIS usually work in the open environments. As a result, the security issue becomes a significant concern in TMIS. Thus, an idle secure authentication scheme is required to guarantee that only the authorized (legal) users will have the ability to access the services from TMIS or the network.

There are the following major advantages of using biometric keys (for example, fingerprints, faces, irises, hand geometry and palm-prints, etc.) as described in [33]:

- Biometric keys can not be lost or forgotten.
- Biometric keys are very difficult to copy or share.
- Biometric keys are extremely hard to forge or distribute.
- Biometric keys can not be guessed easily.
- Someone's biometrics is not easy to break than others.

In 2010, Li and Hwang in [33] proposed an efficient biometric-based remote user authentication scheme using smart cards. In their scheme, the biometric verification is performed using the hash value of the user's personal biometrics. However, it was pointed out in [12, 34] that Li-Hwang's scheme may cause a legal user unable to pass biometric verification during the login and password change phases. In the registration phase of Li-Hwang's scheme, the registration center computes $f_i = h(B_i)$, where B_i is the user's personal biometrics, and f_i is then stored in the smart card. Note that the biometric patterns belonging to the same person may vary slightly from time to time, for example fingerprint and voiceprint. Thereupon, when the user enters next time his/her personal biometric, say B_i^*, which may differ slightly from the biometric B_i given during the registration phase, the verification condition $h\left(B_i^*\right) = f_i$ may never succeed due to sensitive property of the one-way hash function $h(\cdot)$. Li et al. [34] showed that Li-Hwang's scheme is insecure against man-in-the-middle attack and does not provide proper authentication. They provided an efficient solution to Li-Hwang's scheme. Das [12] also pointed out that Li-Hwang's scheme has some design flaws. To withstand the security weaknesses found in Li-Hwang's scheme, Das proposed an efficient solution based on the same assumption of Li-Hwang's scheme that storing the information in a tamper-resistant smart card is secure as passwords. In 2012, An [1] showed that Das's scheme [12] is insecure when the secret information stored in the smart card are revealed to an attacker. To withstand those security flaws, An proposed an enhanced efficient scheme. However, Das and Goswami [16] analyzed the security of An's scheme and showed that An's scheme has three serious security flaws in the design of the scheme: (i) flaw in user's biometric verification during the login phase, (ii) flaw in user's password verification during the login and authentication phases, and (iii) flaw in user's password change locally at any time by the user. Due to these security flaws, An's scheme does not support mutual authentication between a user and the server. In addition, it was shown that An's scheme cannot prevent insider attack. In order to remedy the security weaknesses found in An's scheme, they proposed a new robust and secure anonymous biometric-based remote user authentication scheme using smart cards [16]. Lee and Hsu [30] proposed a biometric based remote user authentication scheme using the extended chaotic map. However, their scheme does not protect insider attack. Further, their scheme does not provide formal security analysis.

In recent years, several user authentication schemes have been proposed [5, 10, 21, 22, 25, 26, 29, 32, 35, 38–41, 47, 50–52, 54–56, 60]. Most of them are either insecure against different attacks or inefficient. Wu et al. [55] proposed a password based authentication scheme for TMIS. Later, He et al. [22] pointed out that Wu et al.'s scheme is insecure against impersonation and insider attacks, and they proposed an efficient solution to overcome these security weaknesses found in Wu et al.'s scheme. However, Wei et al. [54] showed that both Wu et al.'s scheme and He et al.'s scheme fail to provide two-factor security. Again, Zhu [60] showed that Wei et al.'s scheme has some weaknesses. Tan [50] proposed a biometric based user authentication scheme for TMIS. But this scheme does not protect against replay attack. Also, this scheme does not preserve user anonymity and it does not provide any formal security analysis. Mishra et al. [39] showed that Yan et al.'s scheme [58] is vulnerable to the off-line password guessing attack and it does not provide the user anonymity property. Moreover, they pointed out that the login and password change phases are inefficient in Yan et al.'s scheme. In order to remove these weaknesses, they proposed an improved scheme for TIMS. Mishra et al. [40] further proposed an enhanced and efficient biometric-based authentication scheme for TIMS using the nonces. Awasthi and Srivastava [5] proposed a three-factor authentication scheme for TMIS. In 2014, Tan [51] analyzed the security of Awasthi-Srivastava's scheme [5] and showed that their scheme is insecure against reflection attack. In addition, their scheme fails to provide three-factor security and the user anonymity. Tan proposed an efficient user anonymity preserving authentication scheme for TMIS. Further, Arshad and Nikooghadam [2] enhanced the security of Tan's scheme and proposed an improvement.

In this paper, we analyze both Tan's scheme and Arshad and Nikooghadam's scheme [2] for the security. Unfortunately, we have seen that their schemes have still several security drawbacks. Tan [51] extended the security requirements of two-factor authentication schemes to three-factor authentication schemes, which are given below:

- *Mutual authentication.* After run of the protocol, the server should believe that the remote user is a legitimate registered client. The user also believes that the communicating party is the server which the user intended to login to.
- *Server not knowing password and biometric.* The registration center (server) should not have any information about the registered user's password and personal biometrics. This is extremely required because several users may apply the same password to access different servers in the real applications. As a result, if a privileged insider of the registration center knows the password or biometrics of a user U_i, he/she may impersonate U_i for accessing the services from other servers.
- *Freedom of password and biometric update.* A user should be allowed to change/update freely his/her password as well as biometric template without contacting the server. The server must be totally unaware of the change of the user's password and biometric template.
- *Three-factor security.* In the security model for three-factor authentication schemes, an adversary can have full control over the communication channel between the users and the server during the login phase and the authentication and key agreement phase. In the three-factor security adversary model, the adversary is modeled to have at most two of the following three abilities, but it is not allowed to have all the three abilities. The adversary can use the techniques in [28, 37] to extract the information from the smart card, obtain the password, or access the biometric template.

Threat model

We make use of the Dolev-Yao threat model [20] in which two communicating parties communicate over an insecure channel. Any adversary (attacker or intruder) can eavesdrop the transmitted messages over the public insecure channel and he/she has the ability to modify, delete or change the contents of the transmitted messages. Usually the smart card of a user is equipped with the tamper-resistant hardware. However, if a user's smart card is stolen or lost, an attacker can still know all the sensitive information stored in its memory by monitoring the power consumption of the smart card [28, 37].

Our contributions

We list the following contributions made in this paper:

- We have revisited the recently proposed Tan's scheme and then identified that Tan's scheme has the loopholes: (i) it fails to provide proper authentication during the login phase, (ii) it fails to provide correct updation of password and biometric of a user during the password and biometric update phase, (iii) it fails to protect against replay attack, and (iv) it lacks the formal security analysis and verification.
- We have further shown that Arshad and Nikooghadam's scheme fails to protect privileged-insider attack and their scheme also lacks the formal security analysis and verification.
- In order to withstand the security drawbacks found in both Tan's scheme, and Arshad and Nikooghadam's scheme, we have proposed a more efficient and secure three-factor user authentication scheme in TMIS.
- Our scheme is shown to be secure against various known attacks through the rigorous informal and formal security analysis and verification using the widely-accepted AVISPA tool.
- Our scheme is also efficient as compared to Tan's scheme and other related schemes.
- High security and computational efficiency make our scheme to be feasible in order to use it for practice in TMIS applications as compared to Tan's scheme and other related schemes.

Organization of the paper

The remainder of this paper is organized as follows. In Section "Mathematical preliminaries", we discuss some basic mathematical preliminaries, which are essential for describing and analyzing Tan's scheme [51], Arshad and Nikooghadam's scheme [2] as well as our proposed scheme. In Section "Review and cryptanalysis of Tan's scheme, and Arshad and Nikooghadam's scheme", we give an overview of the recently proposed Tan's scheme. In Section "The proposed scheme", we present the various phases of our scheme. In Section "Security analysis of the proposed scheme", we show that our scheme is secure against various known attacks. In next section, we simulate our scheme for the formal security verification using the widely-accepted AVISPA tool in order to show that our scheme is secure. We compare the performance of our scheme with other related schemes in Section "Simulation for formal security verification of our scheme using AVISPA tool". In next section, we conclude the paper.

Mathematical preliminaries

In this section, we briefly describe some mathematical preliminaries, which are essential for describing and analyzing Tan's scheme [51], Arshad and Nikooghadam's scheme [2] as well as our proposed scheme.

Collision-resistant one-way hash function

We define the formal definition of a one-way collision-resistant hash function as follows ([15, 46, 49]).

Definition 1 (Formal definition of one-way collision resistant hash function) A collision-resistant one-way hash function $h : A \to B$, where $A = \{0,1\}^*$ and $B = \{0,1\}^n$, is a deterministic algorithm that takes an input as an arbitrary length binary string $x \in A$ and produces an output $y \in B$ as a binary string of fixed-length, n. Let $Adv_{\mathcal{A}}^{HASH}(t_1)$ denote an adversary (attacker) $\mathcal{A}$'s advantage in finding collision. Then, we have, $Adv_{\mathcal{A}}^{HASH}(t_1) = Pr\left[(x, x') \Leftarrow_R \mathcal{A} : x \neq x' \text{ and } h(x) = h(x')\right]$, where $Pr[E]$ denotes the probability of a random event E, and $\left(x, x'\right) \Leftarrow_R \mathcal{A}$ denotes the pair $\left(x, x'\right)$ is selected randomly by $\mathcal{A}$. In this case, the adversary $\mathcal{A}$ is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by the adversary $\mathcal{A}$ with the execution time t_1. The hash function $h(\cdot)$ is then called collision-resistant, if $Adv_{\mathcal{A}}^{HASH}(t_1) \leq \epsilon_1$, for any sufficiently small $\epsilon_1 > 0$.

Key data extraction process from biometric template

We briefly describe the extraction process of key data from the given biometric of a user using a fuzzy extractor method. The output of a conventional hash function $h(\cdot)$ is sensitive and it may also return completely different outputs even if there is a little variation in inputs. The biometric information is thus prone to various noises during data acquisition, and as a result, the reproduction of actual biometric is hard in common practice. In order to avoid such problem, a fuzzy extractor [7, 19, 23] is used, which has the ability to extract a uniformly random string b and a public information par from the biometric template B with the error tolerance t. In the reproduction process, the fuzzy extractor then recovers the original biometric key data b for a noisy biometric B' using par and t. Let $\mathcal{M} = \{0,1\}^v$ be a finite v-dimensional metric space of biometric data points, $d : \mathcal{M} \times \mathcal{M} \to \mathbb{Z}^+$ a distance function, which can be used to calculate the distance between two points based on the metric chosen, l the number of bits of the output string b_i, and t the error tolerance, where $\mathbb{Z}^+$ represents the set of all positive integers.

Definition 2 The fuzzy extractor is a tuple $(\mathcal{M}, l, t)$, which is defined by the following two algorithms, called Gen and Rep:

- **Gen:** This probabilistic algorithm takes a biometric information $B_i \in \mathcal{M}$ as input and outputs a secret key data $b_i \in \{0,1\}^l$ and a public reproduction parameter par_i, where $Gen(B_i) = \{b_i, par_i\}$.
- **Rep:** This deterministic algorithm takes a noisy biometric information $B_i' \in \mathcal{M}$ and a public parameter par_i related to B_i, and then it reproduces (recovers) the biometric key data b_i. In other words, $Rep\left(B_i', par_i\right) = b_i$ provided that the condition $d\left(B_i, B_i'\right) \leq t$ is satisfied.

For more detailed description of the fuzzy extractor and the extraction procedure, one can refer to [7, 19].

Elliptic curve over a prime field

Let a and $b \in Z_p$, where $Z_p = \{0, 1, \ldots, p-1\}$ and $p > 3$ be a prime, such that $4a^3 + 27b^2 \neq 0 \pmod p$. A non-singular elliptic curve $y^2 = x^3 + ax + b$ over the finite field $GF(p)$ is considered as the set $E_p(a, b)$ of all solutions $(x, y) \in Z_p \times Z_p$ to the congruence: $y^2 = x^3 + ax + b \pmod p$, where a and $b \in Z_p$ are constants such that $4a^3 + 27b^2 \neq 0 \pmod p$, together with a special point $\mathcal{O}$ called the point at infinity or the zero point. If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two points in $E_p(a, b)$, then $P + Q = \mathcal{O}$ implies that $x_Q = x_P$ and $y_Q = -y_P$ and $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E_p(a, b)$. In addition, $E_p(a, b)$ forms an abelian group or commutative group under addition modulo p operation.

Let G be the base point on $E_p(a, b)$ whose order be n, that is, $nG = G + G + \ldots + G(n \text{ times }) = \mathcal{O}$. If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two points on elliptic curve $y^2 = x^3 + ax + b \pmod p$, with $P \neq -Q$, then $R = (x_R, y_R) = P + Q$ is computed as follows ([27, 48]): $x_R = \left(\gamma^2 - x_P - x_Q\right) \pmod p$ and $y_R = (\gamma(x_P - x_R) - y_P) \pmod p$, where $\gamma = \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} \pmod p, \text{ if } P \neq Q \\ \frac{3x_P{}^2 + a}{2y_P} \pmod p, \text{ if } P = Q. \end{cases}$
Moreover, in elliptic curve cryptography, scalar multiplication is defined as repeated additions. For example, if $P \in E_p(a, b)$, then $5P$ is computed as $5P = P + P + P + P + P \pmod p$.

The elliptic curve discrete logarithm problem (ECDLP) is formally defined as in [18] as follows.

Definition 3 (Formal definition of ECDLP) Let $E_p(a, b)$ be an elliptic curve modulo a prime p, and $P \in E_p(a, b)$ and $Q = kP \in E_p(a, b)$ be two points, where $k \in_R Z_p$ (We use the notation $x \Leftarrow_R T$ to denote that the number x is chosen randomly from the set T).
Instance: (P, Q, r) for some $k, r \in_R Z_p$.
Output: **yes**, if $Q = rP$, that is, $k = r$, and output **no**, otherwise.
Consider the following two distributions:

$$\begin{aligned} D_{real} &= \{k \Leftarrow_R Z_p, A = P, B = Q(= kP), \\ &\quad C = k : (A, B, C)\}, \\ D_{rand} &= \{k, r \Leftarrow_R Z_p, A = P, B = Q(= kP), \\ &\quad C = r : (A, B, C)\}. \end{aligned}$$

The advantage of any probabilistic, polynomial-time, 0/1-valued distinguisher $\mathcal{D}$ in solving ECDLP on $E_p(a, b)$ is defined as $Adv^{ECDLP}_{\mathcal{D}, E_p(a,b)} = |Pr[(A, B, C) \leftarrow D_{real} : \mathcal{D}(A, B, C) = 1] - Pr[(A, B, C) \leftarrow D_{rand} : \mathcal{D}(A, B, C) = 1]|$, where the probability $Pr[\cdot]$ is taken over the random choices of k and r. $\mathcal{D}$ said to be a (t_2, ϵ_2)-ECDLP distinguisher for $E_p(a, b)$ if $\mathcal{D}$ runs at most in time t_2 such that $Adv^{ECDLP}_{\mathcal{D}, E_p(a,b)}(t_2) \geq \epsilon_2$.
ECDLP assumption: There exists no (t_2, ϵ_2)-ECDLP distinguisher for $E_p(a, b)$. In other words, for every probabilistic, polynomial-time 0/1-valued distinguisher $\mathcal{D}$, we have $Adv^{ECDLP}_{\mathcal{D}, E_p(a,b)}(t_2) \leq \epsilon_2$, for any sufficiently small $\epsilon_2 > 0$.

Review and cryptanalysis of Tan's scheme, and Arshad and Nikooghadam's scheme

In this section, we review in brief the recently proposed Tan's scheme [51]. We use the notations given in Table 1 for describing and analyzing Tan's scheme. We also point out the security flaw found in Arshad and Nikooghadam's scheme [2], which is an improvement over Tan's scheme. We omit the review of Arshad and Nikooghadam's scheme in this paper to reduce the space of the paper. For this purpose, one can refer the detailed description of Arshad and Nikooghadam's scheme in [2].

Tan's scheme consists of the four phases, namely the registration phase, the login phase, the authentication and key agreement phase, and the password and biometric update phase. At first, the telecare medicine information system server, S_j selects a master key $X_s \in Z_q^*$ and a secure collision-resistant chaotic one-way hash function $h : \{0, 1\}^* \rightarrow Z_q^*$. S_j then computes the system's public key $Y = X_s P$ and declares it as public.

Table 1 Notations used in this paper

Symbol	Description
S_j	Telecare medicine information system server
U_i	i^{th} user
ID_i	Identity of user U_i
PW_i	Password of user U_i
B_i	Biometric information of U_i
K	1024-bit secret number only known to U_i
$h(\cdot)$	Collision-free one-way hash function
X_s	1024-bit secret master key of S_j
S_j	Public key of S_j
p	A large prime number or $p = 2^m$, for some large integer $m > 0$
$E_p(a, b)$	An elliptic curve defined over finite field $GF(p)$ with parameters a and b such that $4a^3 + 27b^2 \neq 0 \pmod p$
$C \oplus D$	Bitwise XORed of data C with data D
$C\|\|D$	Data C concatenates with data D

Description of Tan's scheme

Tan's scheme consists of the following phases.

Registration phase

This phase consists of the following steps:

Step 1. The user U_i first selects an identity ID_i, a chosen password PW_i, a random secret number N, and imprints the biometric information B_i at a sensor. U_i then computes $d = h(PW_i||B_i) \oplus N$ and sends the message $\langle ID_i, d\rangle$ to the server S_j via a secure channel.

Step 2. When the server S_j receives the message in Step 1, it computes $c = h(ID_i||X_s) \oplus d$ and issues a smart card containing the information $\{c, P, q, Y, h(\cdot)\}$ to the user U_i via a secure channel.

Step 3. After receiving the the smart card, the user U_i computes $d_1 = c \oplus N$ and $d_2 = bh(PW_i||B_i||ID_i)$, and then replace c with (d_1, d_2) into the memory of the smart card. It is noted that N is not stored in the smart card.

Login phase

This phase has the following steps:

Step 1. The user U_i first inserts his/her smart card into a card reader, and then provides his/her identity ID_i, password PW_i and imprints the biometric information B_i at the sensor. The smart card computes $d_2^* = h(PW_i||B_i||ID_i)$ and checks the condition $d_2^* = d_2$. If it holds, the smart card continues the next step. Otherwise, the smart card terminates the phase.

Step 2. The smart card chooses a random number $r_i \in Z_q^*$ and then computes $x_i = d_1 \oplus h(PW_i \oplus B_i)$, $R_1 = r_i P$, $R_2 = r_i Y$, $v_i = ID_i \oplus h(R_1||R_2)$, and $z_i = h(ID_i||v_i||R_1||R_2||x_i)$. Finally, the smart card sends the message $\langle R_1, v_i, z_i \rangle$ to the server S_j.

Authentication and key agreement phase

This phase consists of the following steps:

Step 1. After receiving the login message $\langle R_1, v_i, z_i \rangle$, the server S_j computes $R_2^* = X_s R_1$, $ID_i^* = v_i \oplus h\left(R_1||R_2^*\right)$, $x_i^* = h\left(ID_i^*||X_s\right)$, and $z_i^* = h\left(ID_i^*||v_i||R_1||R_2^*||x_i^*\right)$. After that S_j checks if $z_i^* = z_i$ holds or not. If this verification passes, the server S_j authenticates the user U_i. Otherwise, S_j refuses the login request and the phase is terminated immediately.

Step 2. S_j then chooses a random number $r \in Z_q^*$, computes $R = rP$, $z = h\left(rR_1||R_2^*||R||x_i^*\right)$, and sends the message $\langle R, z \rangle$ to the user U_i. S_j also computes the session key shared with the user U_i as $sk = h\left(rR_1||ID_i||R||x_i^*\right)$.

Step 3. After receiving the message in Step 2, the user U_i computes $z^* = h\left(r_i R||R_2||R||x_i\right)$ and checks the condition $z^* = z$. If they match, U_i authenticates the server S_j and computes the same session key shared with the server S_j as $sk = h\left(r_i R||ID_i||R||x_i\right)$.

Password and biometric update phase

In this phase, a user U_i can change his/her old password and biometric information locally without contacting the server S_j using the following steps:

Step 1. U_i first inserts his/her smart card into the card reader, and then provides his/her identity ID_i, old password PW_i and imprints the biometric information B_i at the sensor, and issues an update request to the smart card. The smart cards then computes computes $d_2^* = h\left(PW_i||B_i||ID_i\right)$ and checks the condition $d_2^* = d_2$. If it holds, the smart card continues this phase. Otherwise, the smart card refuses the update request.

Step 2. The smart card instructs the user U_i to choose his/her new password PW_i^{new} and imprint his/her new biometric template B_i^{new}. The smart card then computes $d_1^{new} = d_1 \oplus h(PW_i \oplus B_i) \oplus h\left(PW_i^{new} \oplus B_i^{new}\right)$ and replaces the pair (d_1, d_2) with the computed pair $\left(d_1^{new}, d_2^{new}\right)$.

Drawbacks of Tan's scheme

In this section, we show that Tan's scheme has the following security loopholes.

Fails to provide proper authentication during the login phase

It is known that the input biometric characteristic of the same person can be slightly different every time [12, 23, 34]. The output of a one-way hash function including the chaotic one-way hash function is sensitive, and it may return completely a different output even if there is a little variation in input. Biometric information B_i is prone to various noises during the data acquisition and thus, the production of actual biometric is hard in common practice. Suppose a user U_i enters his/her identity ID_i, correct password PW_i, and imprints the biometric B_i^*, where we assume that B_i^* is slightly different from B_i at that time. After that the smart card computes $d_2^* = h(PW_i||B_i^*||ID_i) \neq h(PW_i||B_i||ID_i)$, since $B_i^* \neq B_i$. The smart card then checks the condition $d_2^* = d_2$. Since it is not valid, the user's biometric and password validations fail, and it terminates the session. As a result, this may cause that the legal user is unable to pass biometric and password verification at the login phase. Thus, Tan's scheme fails to provide proper authentication during the login phase.

Fails to provide correct updation during the password and biometric update phase

This analysis is similar to the above analysis. Assume that the user U_i enters ID_i, correct old password PW_i, and imprints his/her biometric template B_i', which is slightly different from B_i at the time of registration due to nature of biometric template. When the smart card computes $d_2' = h(PW_i||B_i'||ID_i)$ and checks the condition $d_2' = d_2$, this condition will fail, since $B_i' \neq B_i$. As a result, the user U_i may never be successful in passing password and biometric verification due to application of chaotic hash function $h(\cdot)$. Thus, the smart card will refuse the update request, and hence, Tan's scheme also fails to provide proper authentication during the password and biometric update phase.

Fails to protect against replay attack

Suppose an adversary intercepts the login request $\langle R_1, v_i, z_i\rangle$ during the login phase and sends the message $\langle R_1^*, v_i^*, z_i^*\rangle = \langle R_1, v_i, z_i\rangle$ to the server S_j after some time. After receiving this message, S_j computes $R_2^* = X_s R_1^*$, $ID_i^* = v_i^* \oplus h\left(R_1^*||R_2^*\right)$, $x_i^* = h\left(ID_i^*||X_s\right)$, and $z_i^{**} = h\left(ID_i^*||v_i^*||R_1^*||R_2^*||x_i^*\right) = h\left(ID_i||v_i||R_1||R_2^*||x_i^*\right)$. S_j then checks the condition $z_i^{**} = z_i^*$. Since it is valid, S_j authenticates the user U_i, and sends backs the message $\langle R, z\rangle$ to the user U_i, where $R = rP$ and $z = h\left(rR_1^*||R_2^*||R||x_i^*\right)$. Thus, it is clear that the server S_j can not detect whether the message $\langle R_1^*, v_i^*, z_i^*\rangle$ is a replay message or not. Hence, Tan's scheme also fails to protect against replay attack. Note that the approach to address the replay attack is based on the classical methods, such as Needham-Schroeder-based approaches, which can all address this attack.

Lack of formal security analysis and verification

Tan's scheme contains only some informal security analysis and it lacks a rigorous formal security proof and formal security verification using some widely-accepted verification tool such as AVISPA tool [3].

Drawbacks of Arshad and Nikooghadam's scheme

In this section, we show that Arshad and Nikooghadam's scheme [2] has the following security loopholes.

Privileged-insider attack

During the registration phase of Arshad and Nikooghadam's scheme, a user U_i inputs an identity ID_i, a password PWi, and a random number N_C. After that he/she imprints his/her personal biometric B_i at a sensor, and then computes his/her masked password MPW_i as $MPW_i = PW_i \oplus N_C$ and his/her masked biometric MB_i as $MB_i = B_i \oplus N_C$. Finally, U_i sends the registration request message $\langle ID_i, MPW_i, MB_i\rangle$ to the telecare server through a secure channel. At the end of the registration phase, after getting the smart card from the telecare server, U_i stores the random number N_C into his/her smart card.

Assume that the smart card of U_i is lost/stolen and a privileged-insider attacker of the telecare server attains this smart card. According to our threat model (provided in Section "Threat model"), the insider attacker can extract all the sensitive information stored in that smart card using the power analysis attacks [28, 37]. Hence, the attacker now knows N_C, and also the masked password $MPW_i = PW_i \oplus N_C$ and the masked biometric $MB_i = B_i \oplus N_C$ which were provided by the user U_i during the registration phase to the telecare server. Thus, the insider attacker can easily derive the password $PW_i = MPW_i \oplus N_C$ and also the biometric $B_i = MB_i \oplus N_C$. This clearly shows that Arshad and Nikooghadam's scheme is completely insecure against the privileged-insider attack.

Lack of formal security analysis and verification

Arshad and Nikooghadam's scheme contains only some informal security analysis and it lacks a rigorous formal security proof and formal security verification using some widely-accepted verification tool such as AVISPA tool [3].

The proposed scheme

In this section, we describe the various phases of our scheme, which are given in the following subsections. We use the notations provided in Table 1 for describing our scheme.

Setup phase

In this phase, the telecare medicine information system server, S_j executes the following steps:

- Step S1. S_j first selects an elliptic curve $E_q(a, b)$ with parameters: q is a large prime such that the elliptic curve discrete logarithm problem (ECDLP) becomes intractable, and $a, b \in Z_q = \{0, 1, \ldots, q-1\}$ with the condition $4a^3+27b^2 \neq 0 \pmod q$, such that the elliptic curve is non-singular.
- Step S2. S_j then selects a base point $P \in E_q(a, b)$, and a master secret key $X_s \in Z_q^*$, where $Z_q^* = \{a|0 < a < q, \gcd(a, q) = 1\}$, that is, $Z_q^* = \{1, 2, \ldots, q-1\}$.
- Step S3. S_j also selects a secure collision-resistant one-way hash function $h : \{0, 1\}^* \rightarrow Z_q^*$ and the fuzzy extractor functions $Gen(\cdot)$ and $Rep(\cdot)$, and then computes the public key $Y = X_s P$ of the system.
- Step S4. The secret key of S_j is X_s. The public parameters are $\{P, q, Y, h(\cdot), Gen(\cdot), Rep(\cdot)\}$.

Registration phase

The registration phase of our scheme consists of the following steps:

Step R1. The user U_i selects an identity ID_i, and chooses his/her password PW_i.

Step R2. U_i generates a 1024-bit secret number K and computes the masked password $RPW_i = h(ID_i||K||PW_i)$.

Step R3. U_i imprints the biometric information B_i at a sensor and applies the fuzzy extractor to generate secret key b_i and a public parameter par_i as $Gen(B_i) = (b_i, par_i)$, as in [16, 23].

Step R4. U_i computes $f_i = h(RPW_i||b_i)$ and sends the registration request message $\langle ID_i, f_i\rangle$ to the server S_j via a secure channel.

Step R5. After receiving the message in Step R4, the server S_j computes $e_i = h(ID_i||X_s) \oplus f_i$, using its own secret master key X_s, and received information ID_i and f_i. S_j then generates a smart card SC_i for the user U_i containing the information $\{P, q, Y, h(\cdot), Gen(\cdot), Rep(\cdot), f_i, e_i, t, par_i\}$, where t is the error tolerance parameter used in fuzzy extractor, and sends it to the user U_i via a secure channel.

Step R6. After receiving the smart card SC_i from the server S_j, U_i computes $d_i = h(ID_i||b_i) \oplus K$, and stores it into the smart card SC_i. As a result, the smart card SC_i of the user U_i finally contains the information $\{P, q, Y, h(\cdot), Gen(\cdot), Rep(\cdot), f_i, e_i, t, par_i, d_i\}$.

Remark 1 Note that at the end of the registration phase of our scheme, the identity ID_i, password PW_i and biometric information B_i are not directly stored in the smart card SC_i of the user U_i. In addition, our scheme does not reveal the password PW_i and the biometric information B_i of the user U_i to the server S_j also. Thus, the privileged insider attack is completely protected by our scheme due to collision-resistant property of one-way hash function $h(\cdot)$ and difficulty of solving ECDLP. The details are explained in the stolen smart card attack while we analyze later our scheme for security in this paper.

Login phase

In order to login to the server S_j, the user U_i needs to perform the following steps:

Step L1. U_i first inserts his/her smart card SC_i into a card reader. U_i then enters his/her identity ID_i, password PW_i, and imprints the biometric information B_i at the sensor. Note that if the user U_i plans to use a mobile device to login the telecare medicine system, U_i can then use the scan software of the mobile device in order to obtain B_i, and input $\{ID_i, PW_i, B_i\}$ into the login interface of the system as described in Tan's scheme [51].

Step L2. SC_i computes $b_i' = Rep(B_i, par_i)$ using the imprint B_i, and the parameters t and par_i stored in its memory.

Step L3. SC_i computes $K' = d_i \oplus h(ID_i||b_i')$, using the stored information d_i in its memory and computed b_i' in order to obtain the secret number K.

Step L4. SC_i uses K' to compute $RPW_i' = h(ID_i||K'||PW_i)$, and $f_i' = h(RPW_i'||b_i')$. SC_i then checks the condition $f_i' = f_i$. If it holds, it ensures that both information PW_i and B_i entered by U_i are valid, and hence, the user U_i passes both the password and biometric verifications. Otherwise, the phase is terminated immediately.

Step L5. SC_i computes $x_i = e_i \oplus f_i' (= h(ID_i||X_s))$, generates a random number $r_i \in Z_q^*$, and then computes $R_1 = r_i P$, $R_2 = r_i Y$, $v_i = ID_i \oplus h(R_1||R_2) \oplus RN_u$, and $z_i = h(ID_i||v_i||R_1||R_2||x_i||RN_u)$. Here RN_u is a random nonce generated by SC_i on behalf of the user U_i.

Step L6. Finally, the smart card SC_i of the user U_i sends the login request message $\langle R_1, v_i, z_i\rangle$ to the server S_j via a public channel.

Remark 2 The input biometric characteristic of the same person can be slightly different every time [12, 23, 34] and thus, the output of a one-way hash function including the chaotic one-way hash function is sensitive, and it may return completely a different output even if there is a little variation in input. Due to sensitive property of the one-way hash function $h(\cdot)$, Tan's scheme cannot tolerate little variations of biometric feature. On the other hand, even if there is a little variation in biometrics input of a legal user U_i, due to application of fuzzy extractor functions, such as $Gen(\cdot)$ and $Rep(\cdot)$, our scheme has the ability to tolerate little variations of biometric feature as long as the condition $d(B_i, B_i') \le t$ is satisfied (provided in Definition 2), where B_i and B_i' are the biometrics provided by U_i at the registration time and the login time, respectively. Note that a low-entropy or simple password can be guessed using the dictionary attacks [33]. However, as pointed out in [33], as compared to low-entropy passwords, biometric keys can not be lost or forgotten, biometric keys are very difficult to copy or share, biometric keys are extremely hard to forge or distribute, and biometric keys can not be guessed easily. Therefore, it is a very difficult task for an attacker to forge or guess a legal user U_i's biometrics B_i. As a result, that attacker will not have ability to make a little variation of the legal user U_i's biometrics B_i, and he/she can not pass the biometric verification during the login phase.

Authentication and key agreement phase

After receiving the login request message $\langle R_1, v_i, z_i \rangle$ from the user U_i, the server S_j authenticates U_i. In this phase, for mutual authentication purpose, U_i also authenticates S_j. Finally, both U_i and S_j establish a common secret session key SK_{ij} for their future secure communication after successful mutual authentication between them. U_i and S_j perform the following steps:

Step AK1. S_j computes $R_2^* = X_s R_1 = X_s(r_i P) = r_i(X_s P) = r_i Y$ and $RN_u^* = ID_i \oplus v_i \oplus h(R_1||R_2^*)$, $x_i^* = h(ID_i||X_s)$, and $z_i^* = h(ID_i||v_i||R_1||R_2^*||x_i^*||RN_u^*)$. Note that for computing RN_u^*, the server S_j knows ID_i, because it is sent during the registration phase by the user U_i via a secure channel. S_j then compares the computed z_i^* with the received z_i. If there is a mismatch between them, the phase is terminated immediately. Otherwise, S_j authenticates the user U_i as the valid user.

In order to protect the replay and main-in-the-middle attacks, we adopt the similar strategy as in [12, 34]. The server S_j stores (ID_i, RN_u^*) in its database. When the server S_j receives another login request message $\langle R_1', v_i', z_i' \rangle$ from U_i later, it computes $R_2' = X_s R_1'$, $RN_u' = ID_i \oplus v_i' \oplus h(R_1'||R_2')$, $x_i' = h(ID_i||X_s)$ and $z_i'' = h(ID_i||v_i'||R_1'||R_2'||x_i'||RN_u')$. If $z_i'' = z_i$, then S_j makes sure that the login request message is a replay one, and in that case $RN_u' = RN_u^*$. As a result, S_j will reject this login request message. Otherwise, S_j authenticates U_i and updates the pair (ID_i, RN_u^*) by (ID_i, RN_u') in its database since the login request message is treated as a fresh one. Note that S_j can store RN_u^* for a longer time in order to ensure that the same login message will not be replayed be any attacker during the longer time period at least the expiry of the session key between a user U_i and the server S_j. One can also use the timestamp along with the random nonces to protect the replay attack strongly, if the nodes are synchronized with their clocks.

Step AK2. S_j chooses a random number $s_i \in Z_q^*$. S_j then generates a random nonce RN_s, and computes the following: $R_3 = s_i P$, $y_i = x_i^* \oplus RN_s \oplus RN_u^*$, and $z_i^{**} = h(s_i R_1||R_2^*||R_3||y_i||RN_u^*||RN_s||SK_{ij})$, where $SK_{ij} = h(ID_i||x_i^*||RN_u^*||RN_s||R_2^*||R_3)$ is the secret session key to be shared with the user U_i. S_j then sends the authentication request message $\langle R_3, y_i, z_i^{**} \rangle$ to the smart card SC_i (user U_i) via a public channel.

Step AK3. After receiving the message in Step AK2, the smart card SC_i of the user U_i computes the following: $r_i R_3 = r_i(s_i P) = s_i(r_i P) = s_i R_1$, $RN_s^* = y_i \oplus x_i \oplus RN_u$, $SK_{ji} = h(ID_i||x_i||RN_u||RN_s^*||R_2||R_3)$, and $z_i^{***} = h(r_i R_3||R_2||R_3||y_i||RN_u||RN_s^*||SK_{ji})$. SC_i then checks the condition $z_i^{***} = z_i^{**}$. If they are equal, S_j is authenticated by the user U_i. Otherwise, U_i refuses the authentication request.

Step AK4. Finally, U_i stores SK_{ji} and S_j stores SK_{ij} for their future secure communication. Note that $SK_{ij} = SK_{ji}$.

The summary of registration phase, login phase, and authentication and key agreement phase of our scheme is given in Table 2.

Password and biometric update phase

In this phase, the user U_i can update/change his/her password as well as biometric template without contacting further the server S_j. The following steps are essential for this phase:

Step PB1. U_i first inserts his/her smart card into a card reader, and inputs his/her identity ID_i, old password PW_i^{old} and imprints old biometric information B_i^{old} at the sensor.

Step PB2. The smart card SC_i of the user U_i computes $b_i^{old} = Rep(B_i^{old}, par_i)$ and $K^* = d_i \oplus h(ID_i||b_i^{old})$. SC_i then computes $RPW_i^{old} = h(ID_i||K^*||PW_i^{old})$ and $f_i^{old} = h(RPW_i^{old}||b_i^{old})$.

Step PB3. SC_i then checks the condition $f_i^{old} = f_i$. If it holds, both entered PW_i^{old} and B_i^{old} are authenticated by SC_i. Otherwise, SC_i refuses the update request.

Table 2 Summary of exchanged messages during the registration phase, login phase, and authentication and key agreement phase of our scheme

Phase	User (U_i)/Smart Card (SC_i)	Server (S_j)
Registration	$\xrightarrow{\langle ID_i, f_i \rangle}$ (via a secure channel)	
		$\xleftarrow{Smart\,Card(P, q, Y, h(\cdot), Gen(\cdot), Rep(\cdot), f_i, e_i, t, par_i)}$ (via a secure channel)
Login	$\xrightarrow{\langle R_1, v_i, z_i \rangle}$ (via a public channel)	
Authentication and key agreement		$\xleftarrow{\langle R_3, y_i, z_i^{**} \rangle}$ (via a public channel)
	Computes $SK_{ij} = h(ID_i \| \| x_i \| \| RN_u \| \| RN_s^* \| \| R_2 \| \| R_3)$.	Computes $SK_{ij} = h(ID_i \| \| x_i^* \| \| RN_u^* \| \| RN_s \| \| R_2^* \| \| R_3)$.

Step PB4. SC_i asks the user U_i to enter his/her new chosen password PW_i^{new} and imprint new biometric template B_i^{new} at the sensor. SC_i computes $x = e_i \oplus f_i^{old} = h(ID_i||X_s)$ and $RPW_i^{new} = h(ID_i||K^*||PW_i^{new})$.

Step PB5. SC_i then applies the fuzzy extractor function $Gen(\cdot)$ on B_i^{new} to generate secret key b_i^{new} and public parameter par_i^{new} as $Gen(B_i^{new}) = (b_i^{new}, par_i^{new})$. SC_i further computes $f_i^{new} = h(RPW_i^{new}||b_i^{new})$, $e_i^{new} = x \oplus f_i^{new} = h(ID_i||X_s) \oplus f_i^{new}$, and $d_i^{new} = h(ID_i||b_i^{new}) \oplus K^*$.

Step PB6. Finally, the smart card SC_i replaces f_i, e_i, d_i and par_i by f_i^{new}, e_i^{new}, d_i^{new} and par_i^{new}, respectively, into its memory.

Security analysis of the proposed scheme

In this section, we show that our scheme is secure against various known attacks.

Informal security analysis

Through the informal security analysis, we show that our scheme has the ability to defend/provide the following attacks and features.

Reflection attack

Suppose that an attacker (adversary) intercepts a login request message $\langle R_1, v_i, z_i \rangle$. To mount the reflection attack, the attacker needs to replace y_i with v_i and z_i^{**} with z_i as a valid login request message $\langle R_3, v_i, z_i \rangle$ in the authentication request message. Upon receiving this login request message, the server S_j computes $R_2^* = X_s R_3 = s_i Y \neq r_i Y$, $RN_u^* = ID_i \oplus v_i \oplus h(R_3||R_2^*) \neq RN_u$, $x_i^* = h(ID_i||X_s)$, $z_i^* = h(ID_i||v_i||R_3||R_2^*||x_i^*||RN_u^*) \neq h(ID_i||v_i||R_1||R_2||x_i||RN_u)$, since $R_3 \neq R1$. As a result, the verification condition $z_i^* = z_i$ will fail, and the server S_j will terminate this request. Hence, it is clear that as in Tan's scheme, our scheme also protects the reflection attack.

Replay attack

Suppose an attacker intercepts the login request message $\langle R_1, v_i, z_i \rangle$ during the login phase, and sends the message $\langle R_1', v_i', z_i' \rangle = \langle R_1, v_i, z_i \rangle$ to the server S_j again. However, according to the strategy suggested in Step AK1 of our authentication and key agreement phase, this message will be detected as a replay message, since S_j keeps the track of the pair (ID_i, RN_u^*) in its database for a longer time period. Hence, the replay attack is protected in our scheme.

Man-in-the-middle attack

Assume that an attacker intercepts the login request message $\langle R_1, v_i, z_i \rangle$ during the login phase. Note that P and Y are public, whereas X_s is secret to S_j only and ID_i is known to both U_i and S_j only. Let the attacker select a random number $r_i' \in Z_q^*$ and then compute $R_1' = r_i' P$ and $R_2' = r_i' Y$. Furthermore, the attacker generates a random nonce RN_u'. To compute $v_i' = ID_i \oplus h(R_1'||R_2') \oplus RN_u'$, it is clear that the attacker needs to know ID_i. However, ID_i is unknown to the attacker. Thus, the attacker has no way to compute v_i' and also $z_i' = h(ID_i||v_i'||R_1'||R_2'||x_i'||RN_u')$ as computation of $x_i' = h(ID_i||X_s)$ is a computationally infeasible problem and ID_i is unknown to that attacker. Hence, the attacker does not have any ability to modify the message $\langle R_1, v_i, z_i\rangle$ as a valid login request message $\langle R_1', v_i', z_i'\rangle$ in between the communication, and our scheme protects against man-in-the-middle attacks.

Many logged-in users with the same login-id attack

The systems which maintain the password/verifier table in order to verify the user login are usually vulnerable to many logged-in users with the same login-id attack. In our scheme, the server S_j and the user U_i do not maintain any verifier table. To login to the server, a user U_i must have a valid triple $\langle ID_i, PW_i, B_i\rangle$ and a smart card corresponding to these information. Note that our scheme requires on-card computation for password and biometric verification. Further, PW_i and b_i of the user U_i are protected by $h(\cdot)$. Even two users U_i and U_j have the same password PW_i, the hash values $f_i = h(h(ID_i||K_i||PW_i)||b_i)$ and $f_j = h(h(ID_j||K_j||PW_j)||b_j)$ are distinct due to the properties of personal biometrics, random numbers K_i and K_j selected by the users U_i and U_j, respectively, and ID_i and ID_j. Since our scheme requires on-card computation to login in the server, once the smart card is removed from the system, the login session is terminated. As a result, our scheme prevents the many logged-in users with the same login-id attack.

Session key security

Suppose an attacker intercepts the login message $\langle R_1, v_i, z_i\rangle$ during the login phase and the authentication request message $\langle R_3, y_i, z_i^{**}\rangle$ during the authentication and key agreement phase. The secret session key $SK_{ij} = h(ID_i||x_i^*||RN_u^*||RN_s||R_2^*||R_3)$ is embedded in z_i^{**} and also protected by the one-way hash function $h(\cdot)$. In addition, to compute SK_{ij} the attacker needs to know ID_i, x_i^*, RN_u, RN_s and R_2^*. Hence, due to the collision-resistant one-way property of $h(\cdot)$, it is a computationally infeasible problem for the attacker to derive SK_{ij}.

Parallel session attack

When an attacker wants to start another parallel session using the previous session login request message $\langle R_1, v_i, z_i\rangle$ to the server S_j, S_j detects the message as a previous one because the random nonce contained in the message is matched with the stored random nonce in S_j's database corresponding to that user U_i. Further, the attacker does not have any ability to change this message, because the attacker does not know ID_i. The parallel session attack is then completely solved in our scheme.

Protection of user anonymity

Suppose an attacker intercepts the login request message $\langle R_1, v_i, z_i\rangle$ during the login phase and the authentication request message $\langle R_3, y_i, z_i^{**}\rangle$ during the authentication and key agreement phase of our scheme. Note that these values are protected by the one-way collision-resistant hash function $h(\cdot)$ and also determined by two random numbers r_i and s_i, and two random nonces RN_u and RN_s. Due to this, these messages are different in each protocol run and as a result, the attacker can not link two login messages of a particular user U_i. Hence, our scheme preserves the user anonymity property.

Stolen smart card attack

Suppose an attacker obtains a stolen/lost smart card SC_i of a legal user U_i. Then according to our threat model, the attacker can easily extract all the sensitive information $\{P, q, Y, h(\cdot), Gen(\cdot), Rep(\cdot), f_i, e_i, t, par_i, d_i\}$ from the memory of the smart card SC_i by monitoring the power consumption of the smart card [28, 37]. Using f_i and e_i, the attacker can compute $h(ID_i||X_s) = e_i \oplus f_i$. However, both the identity ID_i of the user U_i and the secret master key X_s of the server S_j are unknown to the attacker. Due to the one-way collision-resistant property of $h(\cdot)$, it is computationally infeasible task for the attacker to derive X_s. We have, $f_i = h(RPW_i||b_i) = h(h(ID_i||K||PW_i)||b_i)$ and $d_i = h(ID_i||b_i) \oplus K$. Again, the attacker does not know ID_i, K, b_i and PW_i. To guess PW_i and b_i correctly, the attacker needs to know ID_i and K. Due to secure one-way hash function $h(\cdot)$, the attacker does not have any ability to derive PW_i and b_i. Thus, our scheme is secure against the stolen smart card attacks.

Offline password guessing attack

As in stolen smart card attacks discussed above, the attacker does not have any ability to derive the password PW_i of a legal user U_i even if the attacker obtains the user U_i's stolen/lost smart card. This is because the attacker needs to know ID_i, K and b_i to derive PW_i. As a result, our scheme has the ability to resist the offline password guessing attack.

Online password guessing attack

In this attack, an attacker tries to derive the password PW_i of a user U_i by intercepting all messages during various phases. Note that during the registration phase, the messages are transmitted securely between the user and the server. Suppose an attacker tries to retrieve secret data by intercepting all transmitted messages $\langle R_1, v_i, z_i\rangle$ and $\langle R_3, y_i, z_i^{**}\rangle$ in a previous session. None of these messages involves the user's password PW_i directly or indirectly. As a result, these messages are not helpful for deriving the password PW_i of a user U_i. Hence, our scheme is also secure against online password guessing attack.

Privileged insider attack

During the registration phase, an insider being an attacker at the server S_j may try to know PW_i and b_i of a user U_i. However, in our scheme during the registration phase S_j receives the registration request message $\langle ID_i, f_i\rangle$ from U_i. Note that $f_i = h(RPW_i||b_i) = h(h(ID_i||K||PW_i)||b_i)$, and $Gen(B_i) = (b_i, par_i)$. Since K is not revealed to the server S_j and it is only known to U_i, S_j does not have any ability to determine or guess correctly PW_i and B_i, since PW_i and b_i are protected by $h(\cdot)$. Hence, the insider attack is eliminated from our scheme.

Mutual authentication

In our scheme, after receiving the login request message $\langle R_1, v_i, z_i\rangle$ from the user U_i, the server S_j checks the condition whether $z_i^* = z_i$. If they are equal, S_j authenticates the user U_i as a valid user. On the other hand, after receiving the authentication request message $\langle R_3, y_i, z_i^{**}\rangle$, the smart card SC_i of the user U_i checks the condition $z_i^{**} = z_i^*$. If this condition is valid, U_i authenticates S_j as a valid server. Thus, our scheme provides the mutual authentication between U_i and S_j.

Server not knowing password and biometric

During the registration phase of our scheme, the user U_i sends the registration request message $\langle ID_i, f_i\rangle$ to the server S_j via a secure channel, where $f_i = h(RPW_i||b_i) = h(h(ID_i||K||PW_i)||b_i)$, and $Gen(B_i) = (b_i, par_i)$. Note that S_j does not know K, PW_i and b_i. To know PW_i, the server S_j needs to know K and b_i. Due to the collision-resistant property of $h(\cdot)$, it is a computationally infeasible problem for S_j to derive PW_i and B_i since K is a 1024-bit secret number only known to the user U_i.

Freedom of password and biometric update

In our scheme, before the user U_i updates his/her old password and biometric pair $\{PW_i^{old}, B_i^{old}\}$ by new password and biometric pair $\{PW_i^{new}, B_i^{new}\}$, the smart card SC_i of the user U_i computes $b_i^{old} = Rep(B_i^{old}, par_i)$, $K^* = d_i \oplus h(ID_i||b_i^{old})$, $RPW_i^{old} = h(ID_i||K^*||PW_i^{old})$ and also $f_i^{old} = h(RPW_i^{old}||b_i^{old})$. After that SC_i compares f_i^{old} with the stored f_i. If they match, then only SC_i continues the update phase. Also, it is noted that during the entire duration of the password and biometric update phase, SC_i executes these operations without involving the server S_j. As a result, S_j is totally unaware of the password as well as biometric update.

Three-factor security

In the three-factor security model, the main goals of an attacker are to mount an impersonation attack where the attacker has learned at most two elements of the triple $\{PW_i, SC_i, B_i\}$, in order to obtain the last element or to compromise the user anonymity. As in the analysis of Tan's scheme, it is also clear that our scheme provides the three-factor security.

Formal security analysis

In this section, using the formal security analysis under the random oracle model we show that our scheme is secure. We use the proof of the formal security by the method of contradiction as in [11]. We follow the similar analysis as in [8, 9, 13, 14, 16, 18, 42–44]. Note that one can also prove the formal security in the standard model. However, in this paper, we perform the formal security analysis under the generic group model of cryptography.

In order to use the method of contradiction proof [11] for our formal security analysis, we assume that there exist the following two oracles for an adversary:

- $Reveal1$: This oracle will unconditionally output the input x from the corresponding hash value $y = h(x)$.
- $Reveal2$: Given $P \in E_q(a, b)$ and the public key $Q = kP \in E_q(a, b)$, this oracle will unconditionally output the private key k.

Theorem 1 *Under the elliptic curve discrete logarithm problem (ECDLP) assumption, our proposed scheme is secure against an adversary for deriving the identity ID_i and session key SK_{ij} between a user U_i and the server S_j, if the one-way hash function $h(\cdot)$ closely behaves like a random oracle.*

Proof In this proof, we need to construct an adversary (attacker) $\mathcal{A}$ who will have the ability to derive both ID_i and SK_{ij}. The adversary $\mathcal{A}$ uses the random oracles $Reveal1$ and $Reveal2$ for running the experimental algorithm, say $EXP1_{\mathcal{A},UA}^{HASH,ECDLP}$ provided in Algorithm 1 for our proposed three-factor remote user authentication scheme, say UA. Define the success probability for $EXP1^{HASH,ECDLP}_{\mathcal{A},UA}$ as $Succ1 = 2Pr[EXP1^{HASH,ECDLP}_{\mathcal{A},UA} = 1] - 1$, where $Pr[E]$ denotes the probability of an event E. The advantage function for this experiment becomes $Adv1(et_1, q_{R_1}, q_{R_2}) = \max_{\mathcal{A}}\{Succ1\}$, where the maximum is taken over all $\mathcal{A}$ with execution time et_1, and the number of queries q_{R_1} and q_{R_2} made to the $Reveal1$ and $Reveal2$ oracles, respectively. We call ourscheme is provably secure against an adversary $\mathcal{A}$ for

Algorithm 1 $EXP1^{HASH,ECDLP}_{\mathcal{A},UA}$

Eavesdrop the login request message $\langle R_1, v_i, z_i\rangle$ during the login phase, where $R_1 = r_iP$, $R_2 = r_iY$, $v_i = ID_i \oplus h(R_1||R_2) \oplus RN_u$, and $z_i = h(ID_i||v_i||R_1||R_2||x_i||RN_u)$.
Call $Reveal2$ oracle on input R_1 to retrieve r_i as $r'_i \leftarrow Reveal2(R_1)$.
Compute $R'_2 = r'_iY$ using the public key Y.
Call $Reveal1$ oracle on input z_i to retrieve ID_i, v_i, R_1, R_2, x_i, and RN_u as $(ID'_i, v'_i, R'_1, R''_2, x'_i, RN'_u) \leftarrow Reveal1(z_i)$.
if $(v'_i = v_i)$ and $(R'_2 = R''_2)$ **then**
    Accept ID'_i as the correct ID_i of the user U_i.
    Eavesdrop the authentication request message $\langle R_3, y_i, z^{**}_i\rangle$ during the authentication and key agreement phase.
    Call $Reveal2$ oracle on input R_3 to retrieve s_i as $s'_i \leftarrow Reveal2(R_3)$.
    Compute $RN'_s = y_i \oplus x'_i \oplus RN'_u$.
    Compute $SK'_{ij} = h(ID'_i||x'_i||RN'_u||RN'_s||R'_2||R_3)$.
    Compute $z'_i = h(s'_iR_1||R'_2||R_3||y_i||RN'_u||RN'_s||SK'_{ij})$.
    if $z'_i = z^{**}_i$ **then**
        Accept SK'_{ij} as the correct session key SK_{ij} between U_i and S_j.
        return 1 (Success)
    else
        return 0 (Failure)
    end if
else
    return 0 (Failure)
end if

deriving ID_i and SK_{ij}, if $Adv1(et_1, q_{R_1}, q_{R_2}) \leq \epsilon$, for any sufficiently small $\epsilon > 0$. According to this experiment if the adversary $\mathcal{A}$ has the ability to invert the one-way hash function $h(\cdot)$ and solve ECDLP, he/she can easily derive both ID_i and SK_{ij}, and win the game. However, by Definition 2.1, it is a computationally infeasible problem to invert $h(\cdot)$, that is, $Adv^{HASH}_{\mathcal{A}}(t_1) \leq \epsilon_1$, for any sufficiently small $\epsilon_1 > 0$. Also, by Definition 2.3, it is computationally infeasible to derive k from P and $Q = kP$ in $E_q(a,b)$, that is, $Adv^{ECDLP}_{\mathcal{D},E_p(a,b)}(t_2) \leq \epsilon_2$, for any sufficiently small $\epsilon_2 > 0$. Hence, we have $Adv1(et_1, q_{R_1}, q_{R_2}) \leq \epsilon$, since $Adv1\,(et_1, q_{R_1}, q_{R_2})$ depends on other advantages $Adv^{HASH}_{\mathcal{A}}(t_1)$ and $Adv^{ECDLP}_{\mathcal{D},E_p(a,b)}(t_2)$. □

Theorem 2 *Under the assumption that the one-way hash function $h(\cdot)$ closely behaves like an oracle, our proposed scheme is secure against an adversary for deriving the secret key X_s of the server S_j, and the password PW_i and the biometric key b_i of the user U_i.*

Proof We construct an adversary $\mathcal{A}$ who will have the ability to derive the secret key X_s of the server S_j, and the password PW_i and the biometric key b_i of the user U_i. For this purpose, the adversary $\mathcal{A}$ can run the experiment provided in Algorithm 2 for our proposed three-factor remote user authentication scheme. We define the success probability for this experiment as $Succ2 = Pr[EXP2^{HASH}_{\mathcal{A},UA} = 1] - 1$. The advantage function for this experiment is $Adv2\,(et_2, q_{R_1}) = \max_{\mathcal{A}}\{Succ2\}$, where the maximum is taken over all $\mathcal{A}$ with execution time et_2, and the number of queries q_{R_1} made to the $Reveal1$ oracles. Our scheme is said to be provably secure against an adversary $\mathcal{A}$ for deriving the secret key X_s of the server S_j, and the password PW_i and the biometric key b_i of the user U_i, if $Adv2(et_2, q_{R_1}) \leq \epsilon$, for any sufficiently small $\epsilon > 0$. According to the experiment provided in Algorithm 2, if the adversary $\mathcal{A}$ has the ability to invert the one-way hash function $h(\cdot)$, he/she can easily derive X_s, PW_i and b_i, and win the game. However, by Definition 2.1, it is a computationally infeasible problem to invert $h(\cdot)$, that is, $Adv^{HASH}_{\mathcal{A}}(t_1) \leq \epsilon_1$, for any sufficiently small $\epsilon_1 > 0$. Hence, we have $Adv2\,(et_2, q_{R_1}) \leq \epsilon$, since $Adv2\,(et_2, q_{R_1})$ depends on the advantage $Adv^{HASH}_{\mathcal{A}}(t_1)$. □

Algorithm 2 $EXP2^{HASH}_{\mathcal{A},UA}$

Extract all the information $\{P, q, Y, h(\cdot), Gen(\cdot), Rep(\cdot), f_i, e_i, t, par_i, d_i\}$ from the stolen/lost smart card SC_i of a legal user U_i by monitoring the power consumption of the smart card [28], [37].
Using f_i and e_i, compute $h(ID_i||X_s) = e_i \oplus f_i$.
Call $Reveal1$ oracle on input $h(ID_i||X_s)$ to retrieve the information ID_i and X_s as $(ID'_i, X'_s) \leftarrow Reveal1(h(ID_i||X_s))$.
Call $Reveal1$ oracle on input $f_i = h(RPW_i||b_i)$ to retrieve RPW_i and b_i as $(RPW'_i, b'_i) \leftarrow Reveal1(f_i)$.
Call $Reveal1$ oracle on input RPW'_i in order to retrieve ID_i, K, and PW_i as $(ID''_i, K'', PW''_i) \leftarrow Reveal1(RPW'_i)$.
if $ID''_i = ID'_i$ **then**
    Accept X'_s as the correct secret key X_s of the server S_j.
    Compute $K^* = d_i \oplus h(ID'_i||b'_i)$.
    if $K^* = K''$ **then**
        Accept PW''_i and b'_i as the correct password PW_i and the biometric key b_i of the user U_i.
        return 1 (Success)
    else
        return 0 (Failure)
    end if
else
    return 0 (Failure)
end if

Simulation for formal security verification of our scheme using AVISPA tool

In this section, we simulate our scheme for the formal security verification using the widely-accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) tool in order to show that our scheme is secure. We have further simulate Tan's scheme for the formal security analysis, and show that Tan's scheme is not secure.

AVISPA overview

AVISPA stands for a push-button tool for the automated validation of Internet security-sensitive protocols and applications. It basically provides a modular and expressive formal language for specifying protocols and their security properties, and integrates different back-ends that implement a variety of state-of-the-art automatic analysis techniques [3]. We have used the widely-accepted AVISPA back-end for our formal security verification [9, 13, 14, 17, 24]. AVISPA consists of four back-ends, which are OFMC, CL-AtSe, SATMC and TA4SP. A static analysis needs to perform in order to check the executability of the protocol, and then the protocol and the intruder actions are compiled into an intermediate format (If). If is the start point for the four automated protocol analysis techniques. It is a lower-level language than HLPSL, and is read directly by the back-ends to the AVISPA tool. The detailed descriptions of these back-ends are given in [3].

In AVISPA, the designed protocols need to be specified in HLPSL language [53]. HLPSL is based on roles: the basic roles represent each participant role, and composition roles represent the scenarios of basic roles. Each role is independent from the others, which gets some initial information by parameters, and then communicates with the other roles by channels. In HLPSL, the intruder is always modeled using the Dolev-Yao model [20] (as in the threat model used in this paper) with the possibility for the intruder to assume a legitimate role in a protocol run. The role system defines the number of sessions, and the number of principals and the roles. The output format (OF) of AVISPA is generated by using one of the four back-ends. When the analysis of a protocol has been successful (by finding an attack or not), the output describes precisely what is the result, and under what conditions it has been obtained. The detailed formats of the OF can be found in [53].

Specifying our scheme

We have implemented the registration phase, the login phase and the authentication and key agreement phase of our scheme in HLPSL language. In our implementation, we have two basic roles: *alice* and *bob*, which represent the participants as the user U_i and the telecare medicine information system server S_j, respectively. The specification in HLPSL language for the role of the initiator, that is, the user U_i is shown in Fig. 1. The user U_i first receives the start signal and changes its state from 0 to 1, and then sends the registration request message $\langle ID_i, f_i\rangle$ to the server S_j securely using the $Snd(\,)$ operation. The user U_i is issued with a smart card by the server S_j with the information $(P, q, Y, h(\cdot), Gen(\cdot), Rep(\cdot), f_i, e_i, t, par_i)$ securely with the $Rcv(\,)$ operation. During the login phase, U_i sends the login request message $\langle R_1, v_i, z_i\rangle$ to S_j via a public channel. Finally, U_i receives the authentication request message $\langle R_3, y_i, z_i^{**}\rangle$ from S_j via a public channel. Note that the type declaration *channel* (dy) declares that the channel is for the Dolev-Yao threat model. As a result, the intruder, which is always denoted by i, has the ability to intercept, analyze, and/or modify messages transmitted over the insecure channel. In HLPSL language, witness(A,B,id,E) declares for a (weak) authentication property of A by B on E, declares that agent A is witness for the information E; this goal will be identified by the constant id in the goal section [3]. On the other hand, request(B,A,id,E) is for a strong authentication property of A by B on E, declares that agent B requests a check of the value E; this goal will be identified by the constant id in the goal section [3]. The declaration witness(Ui, Sj, alice_bob_rnu, RNu') tells that U_i has freshly generated the value RN_u for S_j. request(Sj, Ui, bob_alice_rns, RNs') is a declaration to mean that U_i's acceptance of the value RN_s generated for U_i by S_j. In other words, U_i authenticates S_j based on RN_s. The declaration secret(X, id, A) indicates that the information X is kept secret permanently to the agent A, and the label id (of type protocol_id) is used to identify the goal.

In Fig. 1, we have also implemented the specification in HLPSL language for the role of the responder, the server S_j. During the registration phase, after receiving the registration request message $\langle ID_i, f_i\rangle$ securely from the user U_i, the server S_j issues a smart card SC_i and sends it with the information $(P, q, Y, h(\cdot), Gen(\cdot), Rep(\cdot), f_i, e_i, t, par_i)$ securely to U_i. During the authentication and key agreement phase, after receiving the login request message $\langle R_1, v_i, z_i\rangle$ in the login phase via a public channel, the server S_j sends the authentication request message $\langle R_3, y_i, z_i^{**}\rangle$ to U_i via a public channel.

Finally, we have specified The roles for the session, and the goal and environment of our scheme are specified in Fig. 2. In the session role, all the basic roles: alice and bob are considered as the instances with concrete arguments. The top-level role (environment) is always defined in the specification of HLPSL language. The intruder participates in the execution of protocol as a concrete session.

Fig. 1 Role specification in HLPSL for the user U_i and the server S_j of our scheme

```
role alice (Ui, Sj   : agent,
            SKuisj : symmetric_key,
            % H is one−way hash function
            H: hash_func,
            Snd, Rcv: channel(dy))
% Player by the initiator: the user Ui
played_by Ui
def=
 local State  : nat,
       RPWi, PWi, Bi, Xs, K, IDi, Q,
       Fi, Ri, Si, P, RNu, RNs, SKij:  text,
       F, Gen, Rep : hash_func
 const alice_bob_rnu,  bob_alice_rns,
       subs1, subs2, subs3,
       subs4, subs5 : protocol_id
init  State := 0
transition
% Registration phase
 1. State = 0   /\ Rcv(start) =|>
    State' := 1 /\ Fi' := H(H(IDi.K.PWi).Bi)
% Send the registration request message to Sj
               /\ Snd({IDi.Fi'}_SKuisj)
/\ secret({Xs}, subs1, Sj)
               /\ secret({PWi, Bi, K}, subs2, Ui)
               /\ secret({IDi}, subs3, {Ui,Sj})
% Receive the registration acknowledgment message from Sj
 2. State = 1 /\ Rcv({P.Q.F(Xs.P).H(H(IDi.K.PWi).Bi).
                      xor(H(IDi.Xs),H(H(IDi.K.PWi).Bi)).
                      H.Rep}_SKuisj) =|>
% Login phase
    State' := 2 /\ Ri' := new()
               /\ RNu' := new()
               /\ secret(Ri', subs4, Ui)
% Send the login request message to Sj
               /\ Snd(F(Ri'.P).
                  xor(xor(IDi, H(F(Ri'.P).F(Ri'.Xs.P))),RNu').
                  H(IDi.xor(xor(IDi, H(F(Ri'.P).F(Ri'.Xs.P))),RNu').
                  F(Ri'.P).F(Ri'.Xs.P).H(IDi.Xs).RNu'))
% Ui has freshly generated the value RNu for Sj
               /\ witness(Ui, Sj, alice_bob_rnu, RNu')
%Authentication and session key agreement phase
% Receive the authentication request message from Sj
3. State = 2  /\ Rcv(F(Si'.P).xor(xor(H(IDi.Xs),RNs'),RNu').
                     H(F(Si'.Ri'.P).F(Ri'.Xs.P).F(Si'.P).
                     xor(xor(H(IDi.Xs),RNs'),RNu').RNu'.RNs'.
                     H(IDi.H(IDi.Xs).RNu'.RNs'.F(Ri'.Xs.P).
                     F(Si'.P))) ) =|>
% Ui's acceptance of the value RNs generated for Ui by Sj
State' := 3   /\ request(Sj, Ui, bob_alice_rns, RNs')
end role
```

```
role  bob (Ui, Sj   : agent,
            SKuisj : symmetric_key,
            % H is one−way hash function
            H: hash_func,
            Snd, Rcv: channel(dy))
% Player by the responder: the server Sj
played_by Sj
def=
 local State  : nat,
       RPWi, PWi, Bi, Xs, K, IDi, Q,
       Fi, Ri, Si, P, RNu, RNs, SKij:  text,
       F, Gen, Rep : hash_func
 const alice_bob_rnu,  bob_alice_rns,
       subs1, subs2, subs3,
       subs4, subs5 : protocol_id
init  State := 0
 transition
% Registration phase
% Receive the registration request message from Ui
1. State  = 0 /\ Rcv({IDi.H(H(IDi.K.PWi).Bi)}_SKuisj) =|>
  State' := 1 /\ secret({Xs}, subs1, Sj)
              /\ secret({PWi, Bi, K}, subs2, Ui)
              /\ secret({IDi}, subs3, {Ui,Sj})
% Send the registration acknowledgment message to Ui
              /\ Snd({P.Q.F(Xs.P).H(H(IDi.K.PWi).Bi).
                  xor(H(IDi.Xs),H(H(IDi.K.PWi).Bi)).
                  H.Rep}_SKuisj)
% Login phase
% Receive the login request message from Ui
2. State = 1 /\ Rcv(F(Ri'.P).
               xor(xor(IDi, H(F(Ri'.P).F(Ri'.Xs.P))),RNu').
               H(IDi.xor(xor(IDi, H(F(Ri'.P).F(Ri'.Xs.P))),RNu').
               F(Ri'.P).F(Ri'.Xs.P).H(IDi.Xs).RNu')) =|>
% Authentication and session key agreement phase
 State' := 2 /\ Si' := new()
             /\ RNs':= new()
             /\ secret(Si', subs5, Sj)
% Send the authentication request message to Ui
             /\ Snd(F(Si'.P).xor(xor(H(IDi.Xs),RNs'),RNu').
                 H(F(Si'.Ri'.P).F(Ri'.Xs.P).F(Si'.P).
                 xor(xor(H(IDi.Xs),RNs'),RNu').RNu'.RNs'.
                 H(IDi.H(IDi.Xs).RNu'.RNs'.F(Ri'.Xs.P).
                 F(Si'.P))))
% Sj has freshly generated the value RNs for Ui
             /\ witness(Sj, Ui, bob_alice_rns, RNs')
% Sj's acceptance of the value RNu generated for Sj by Ui
            /\ request(Ui, Sj, alice_bob_rnu, RNu')
end role
```

In the HLPSL implementation of our scheme, we have five secrecy goals and two authentication goals. For example, the secrecy goal secrecy_of subs1 tells that X_s is kept secret to the server S_j only, which is indicated by the protocol id subs1. Similarly, we have given other secrecy goals for the protocol ids subs2, subs3, subs4 and subs5. On the other hand, the authentication goal authentication_on alice_bob_rnu presents that $U_i(C_i)$ generates a random nonce RN_u, where RN_u is only known to U_i. When the server S_j receives RN_u from other messages from U_i, the server S_j performs a strong authentication for U_i based on RN_u. Other authentication goal authentication_on bob_alice_rns indicates S_j generates a random nonce RN_s, where RN_s is only known to S_j. If the user U_i receives RN_s from other messages from S_j, the user U_i (the smart card SC_i) performs a strong authentication for S_j based on RN_s.

Simulation results

We have chosen the back-end OFMC for an execution test and a bounded number of sessions model checking [6]. For the replay attack checking, this back-end checks whether the legitimate agents can execute the specified protocol by performing a search of a passive intruder. After that this back-end gives the intruder the knowledge of some normal sessions between the legitimate agents. For the Dolev-Yao model check, this back-end also checks whether there is any man-in-the-middle attack possible by the intruder. We have simulated our scheme for formal security verification using OFMC back-end under the AVISPA web tool [4]. The simulation results for the formal security verification of our scheme using OFMC are shown in Fig. 3. In this figure, the first printed section, called the SUMMARY, indicates

Fig. 2 Role specification in HLPSL for the session, and the goal and environment of our scheme

```
role session(Ui, Sj: agent,
           SKuisj : symmetric_key,
           H : hash_func)
def=
  local  SI, SJ, RI, RJ: channel (dy)
  composition
          alice (Ui, Sj, SKuisj, H, SI, RI)
       /\  bob   (Ui, Sj, SKuisj, H, SJ, RJ)
end role
```

```
role environment()
def=
  const ui, sj: agent,
       skuisj : symmetric_key,
       h   : hash_func,
       pwi, bi, xs, k, idi, rnu, rns:  text,
       alice_bob_rnu,  bob_alice_rns,
       subs1, subs2, subs3, subs4, subs5 : protocol_id
  intruder_knowledge = {ui, sj, h}
  composition
   session(ui, sj, skuisj, h)
/\ session(ui, sj, skuisj, h)
end role
goal
  secrecy_of subs1
  secrecy_of subs2
  secrecy_of subs3
  secrecy_of subs4
  secrecy_of subs5
  authentication_on alice_bob_rnu
  authentication_on bob_alice_rns
end goal
environment()
```

whether the protocol is safe, unsafe, or whether the analysis is inconclusive. It is clear that our scheme is safe from the printed SUMMARY section. DETAILS section explains under what condition the protocol is declared safe, or what conditions have been used for finding an attack, or finally why the analysis was inconclusive. It is also noted that our scheme is declared as safe, and no attack is found in our scheme. As a result, the result in this figure ensures that our scheme is secure against passive and active attacks including the replay and man-in-the-middle attacks.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web−interface−computation/
  ./tempdir/workfileYs0p0j.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.11s
  visitedNodes: 13 nodes
  depth: 4 plies
```

Fig. 3 The result of the analysis using OFMC backend of our scheme

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/avispa/web−interface−computation/
./tempdir/workfilexSEl7J.if
GOAL
  authentication_on_alice_bob_ri
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.23s
  visitedNodes: 22 nodes
  depth: 3 plies
```

Fig. 4 The result of the analysis using OFMC backend of Tan's scheme

Table 3 Notations used for the computational complexity

Symbol	Description
T_h	Time for performing a one-way hashing operation $h(\cdot)$
T_X	Time for performing an XOR operation
T_E	Time for performing a symmetric encryption operation
T_D	Time for performing a symmetric decryption operation
T_{PE}	Time for executing an asymmetric encryption operation
T_{PD}	Time for executing an asymmetric decryption operation
T_C	Time for executing a Chebyshev chaotic map operation
T_M	Time for executing an ECC point multiplication
T_{FE}	Time for executing a fuzzy extractor

Since our scheme is an improved three-factor remote user authentication scheme for TMIS over Tan's scheme, we have further simulated Tan's scheme for the formal security verification using AVISPA tool. We have implemented the roles for user, server, session, goal and environment in HLPSL for Tan's scheme, and then simulated using the OFMC backend. The simulation results for the formal security verification of Tan's scheme are shown in Fig. 4. The results clearly indicate that Tan's scheme is not secure against passive and active attacks including the replay and man-in-the-middle attacks.

Performance comparison with other related schemes

In this section, we compare the functionality features and performance of our scheme with those for other related three-factor authentication schemes [2, 5, 12, 30, 50, 51].

For the performance comparison, we use the notations listed in Table 3. As pointed out in [23], the computational time of a one-way hashing operation $h(\cdot)$, a symmetric encryption/decryption, a modular exponentiation, and an elliptic curve point multiplication are 0.00032 s, 0.0056 s, 0.0192 s, and 0.0171 s, respectively. For asymmetric cryptosystem (for example, RSA), the computational time for executing encryption/decryption is taken as that for a modular exponentiation operation. According to the experiments in [31], the time for executing a Chebyshev chaotic map operation is 0.0322 s. As in [23], we also assume that the time for executing a fuzzy extractor is also same as that for an elliptic curve point multiplication at the most. We have compared the performance of our scheme with other related three-factor schemes in Table 4 for all the phases. Note that the portion of same data presented in Table 4 is taken from [51]. It is assumed that the time for executing an XOR operation is negligible. It is observed that the rough computational costs of our scheme and other schemes [2, 5, 12, 30, 50, 51] are 0.19514 s, 0.01696 s, 0.0048 s, 0.26432 s,

Table 4 Comparison of performance

Phase	Node	[50]	[12]	[30]	[5]	[51]	[2]	Ours
R	U_i	$2T_h$	–	$3T_h+T_X$	$2T_X+T_{PE}$	$2T_h+3T_X$	$2T_X$	$2T_h+T_{FE}$
	S_j	$2T_h+T_X$	$3T_h+3T_X$	$2T_h+2T_X$	$3T_h+4T_X$ $+T_{PD}$	T_h+T_X	$4T_h+7T_X$	$2T_h+2T_X$
L	U_i	$4T_h+2T_X$ $+T_E$	$2T_h+3T_X$	$5T_h+4T_X$ $+2T_c$	$3T_h+3T_X$	$4T_h+3T_X$ $+2T_M$	$3T_h+5T_X$ $+T_M$	$5T_h+4T_X$ $+2T_M+T_{FE}$
	S_j	–	–	–	–	–		–
AK	U_i	$2T_h$	$3T_h+T_X$	$2T_h+2T_C$	T_h+T_X	T_h+2T_M	$5T_h+2T_X$ $+T_M$	$5T_h+4T_X+$ $2T_M+T_{FE}$
	S_j	$3T_h+T_X$ $+T_D$	$5T_h+2T_X$	$5T_h+T_X$ $+4T_C$	$4T_h+4T_X$	$4T_h+T_X$ $+3T_M$	$8T_h+6T_X$ $+2T_M$	$2T_h+2T_X$ $+3T_M$
PB	U_i	$5T_h+2T_X$	$2T_h+T_X$	$4T_h+5T_X$	$2T_h+4T_X$	$4T_h+4T_X$	$4T_h+14T_X$	$6T_h+$ $4T_X+2T_{FE}$
	S_j	–	–	–	–	–		–
	Total	$18T_h+6T_X$ $+T_E+T_D$ ≈ 0.01696 s	$15T_h+9T_X$ ≈ 0.0048 s	$21T_h+13T_X$ $+8T_C$ ≈ 0.26432 s	$13T_h+18T_X$ $+T_{PE}+T_{PD}$ ≈ 0.04256 s	$16T_h+12T_X$ $+7T_M$ ≈ 0.12482 s	$24T_h+36T_X$ $+4T_M$ ≈ 0.07608 s	$22T_h+4T_{FE}+$ $14T_X+7T_M$ ≈ 0.19514 s

Note: R: Registration phase; L: Login phase; AK: Authentication and key agreement phase; PB: Password and biometric update phase

Table 5 Functionality comparison

	[50]	[12]	[30]	[5]	[51]	[2]	Ours
F_1	Yes	Yes	Yes	Yes	Yes	Yes	Yes
F_2	Yes	No	Yes	Yes	Yes	No	Yes
F_3	Yes	No	No	Yes	Yes	No	Yes
F_4	Yes	Yes	Yes	Yes	Yes	Yes	Yes
F_5	No	Yes	Yes	Yes	No	Yes	Yes
F_6	Yes	Yes	Yes	No	Yes	Yes	Yes
F_7	Yes	No	Yes	No	Yes	Yes	Yes
F_8	No	No	Yes	No	Yes	Yes	Yes
F_9	Yes	No	Yes	No	Yes	Yes	Yes
F_{10}	No	No	No	No	No	No	Yes
F_{11}	No	No	No	No	No	No	Yes

Note: F_1 : mutual authentication; F_2 : server not knowing password; F_3 : server not knowing biometrics; F_4 : freedom of password and biometric update; F_5 : replay attack protection; F_6 : reflection attack protection; F_7 : three-factor security; F_8 : user anonymity; F_9 : key agreement; F_{10} : formal security analysis; F_{11} : formal security verification using AVISPA tool

0.04256 s, 0.12482 s, and 0.07608 s respectively. Note that the registration phase is only one time process, and the password and biometric update phase is not executed frequently. Thus, the computational complexity of our scheme for the login phase, and the authentication and key agreement phase becomes 0.12386 s only.

We have compared the functionality analysis in terms of security properties of our scheme with other related schemes in Table 5. It is clear that our scheme is superior than other schemes. Our scheme provides all the functionality such as mutual authentication, server not knowing password and biometric, replay attack protection, reflection attack protection, freedom of password and biometric update, three-factor security, user anonymity, key agreement, formaljj security analysis under random oracle models and formal security verification using the widely-accepted AVISPA tool. In addition, our scheme protects other attacks, which are described in section "Security analysis of the proposed scheme". All other schemes do not provide formal security analysis and verification. The replay attack is not protected in [50, 51]. The user anonymity property is not supported in [5, 12, 50]. Moreover, Arshad and Nikooghadam's scheme [2] fails to protect the privileged-insider attack. As compared to other three-factor schemes, our scheme is suitable for real-life practical applications due to its high security.

Conclusion

We have revisited the recently proposed Tan's three-factor authentication scheme for the telecare medicine information systems and shown that though Tan's scheme is efficient, it has several security drawbacks. After that we have also shown that Arshad and Nikooghadam's scheme, which is an improvement of Tan's scheme, fails to protect the privileged-insider attack. To remedy such weaknesses, we have proposed an efficient scheme. Our scheme is shown to be secure through the rigorous formal and informal security analysis. In addition, we have shown that our scheme is secure under the formal security verification. Though our scheme requires little more computational cost as compared to Tan's scheme, Arshad and Nikooghadam's scheme, and other schemes, our scheme is more suitable for practical applications due to its high security. Furthermore, it is shown that our scheme preserves the user anonymity property and all other features which are required for an idle three-factor authentication scheme for the telecare medicine information systems.

Acknowledgments The author would like to acknowledge the helpful suggestions of the anonymous reviewers and the Editor, which have improved the content and the presentation of this paper.

Conflict of interests The author declares that there is no conflict of interest.

References

1. An, Y., Security Analysis and Enhancements of an Effective Biometric-Based Remote User Authentication Scheme Using Smart Cards. *J. Biomed. Biotechnol.* 2012:1–6, 2012. Article ID 519723.
2. Arshad, H., and Nikooghadam, M., Three-Factor Anonymous Authentication and Key Agreement Scheme for Telecare Medicine Systems Information. *J. Med. Syst.* 38(6):1–12, 2014.
3. AVISPA: Automated Validation of Internet Security Protocols and Applications. Accessed on January 2013. http://www.avispa-project.org/.
4. AVISPA: AVISPA Web Tool. Accessed on April 2014. http://www.avispa-project.org/web-interface/expert.php/.
5. Awasthi, A.K., and Srivastava, K., A Biometric Authentication Scheme for Telecare Medicine Information Systems with Nonce. *J. Med. Syst.* 37(5):1–4, 2013.
6. Basin, D., Modersheim, S., Vigano, L., OFMC: A symbolic model checker for security protocols. *Int. J. Inf. Secur.* 4(3):181–208, 2005.
7. Burnett, A., Byrne, F., Dowling, T., Duffy, A., A Biometric Identity Based Signature Scheme. *Int. J. Netw. Secur.* 5(3):317–326, 2007.
8. Chatterjee, S., and Das, A.K., An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks. *Security and Communication Networks*, 2014. doi:10.1002/sec.1140.
9. Chatterjee, S., Das, A.K., Sing, J.K., An Enhanced Access Control Scheme in Wireless Sensor Networks. *Ad Hoc & Sensor Wireless Networks* 21(1–2):121–149, 2014.
10. Chen, B.-L., Kuo, W.-C., Wuu, L.-C., Robust smart-card-based remote user password authentication scheme. *Int. J. Commun. Syst.* 27(2):377–389, 2014.

11. Chuang, Y.-H., and Tseng, Y.-M., An efficient dynamic group key agreement protocol for imbalanced wireless networks. *Int. J. Netw. Manag.* 20(4):167–180, 2010.
12. Das, A.K., Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards. *IET Inf. Secur.* 5(3):145–151, 2011.
13. Das, A.K., A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications. *Netw. Sci.* 2(1–2):12–27, 2013.
14. Das, A.K., Chatterjee, S., Sing, J.K., A novel efficient access control scheme for large-scale distributed wireless sensor networks. *Int. J. Found. Comput. Sci.* 24(5):625–653, 2013.
15. 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):1–16, 2013.
16. Das, A.K., and Goswami, A., A robust anonymous biometric-based remote user authentication scheme using smart cards.In: *Journal of King Saud University - Computer and Information Sciences (Elsevier)*: In Press, 2014.
17. Das, A.K., Massand, A., Patil, S., A novel proxy signature scheme based on user hierarchical access control policy. *Journal of King Saud University - Comput. Inform. Sci.* 25(2):219–228, 2013.
18. Das, A.K., Paul, N.R., Tripathy, L., Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem. *Inf. Sci.* 209(C):80–92, 2012.
19. Dodis, Y., Reyzin, L., Smith, A., Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Proceedings of the Advances in Cryptology (Eurocrypt'04), LNCS, Vol. 3027, pp. 523–540, 2004.
20. Dolev, D., and Yao, A., On the security of public key protocols. *IEEE Trans. Inf. Theory* 29(2):198–208, 1983.
21. Giri, D., Maitra, T., Amin, R., Srivastava, P.D., An efficient and robust rsa-based remote user authentication for systems telecare medical information. *J. Med. Syst.* 39(1):1–9, 2014.
22. He, D., Chen, J., Zhang, R., A More Secure Authentication Scheme for Telecare Medicine Information Systems. *J. Med. Syst.* 36(3):1989–1995, 2012.
23. He, D., Kumar, N., Lee, J.-H., Sherratt, R.S., Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Trans. Consum. Electron.* 60(1):30–37, 2014.
24. Islam, S.H., and Biswas, G.P., A provably secure identity-based strong designated verifier proxy signature scheme from pairings bilinear. *Journal of King Saud University - Comput. Inform. Sci.* 26(1):55–67, 2014.
25. Islam, S.K.H., and Khan, M.K., Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems. *J. Med. Syst.* 38(10):1–16, 2014.
26. Khan, M.K., and Kumari, S., Cryptanalysis and improvement of an efficient and secure dynamic id-based authentication scheme for telecare medical information systems. *Security and Communication Networks* 7(2):399–408, 2014.
27. Koblitz, N., Elliptic Curves Cryptosystems. *Math. Comput.* 48:203–209, 1987.
28. Kocher, P., Jaffe, J., Jun, B., Differential power analysis. In: Proceedings of Advances in Cryptology - CRYPTO'99, LNCS, Vol. 1666, pp. 388–397, 1999.
29. 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):1–11, 2013.
30. Lee, C.-C., and Hsu, C.-W., A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. *Nonlinear Dyn.* 71(1–2):201–211, 2013.
31. Lee, C.-C., Li, C.-T., Chiu, S.-T., Lai, Y.-M., A new three-party-authenticated key agreement scheme based on chaotic maps without password table. *Nonlinear Dyn.* 1–11, 2014. doi:10.1007/s11071-014-1827-x.
32. Lee, T.-F., and Liu, C.-M., A Secure Smart-Card Based Authentication and Key Agreement Scheme for Telecare Medicine Information Systems. *J. Med. Syst.* 37(3):1–8, 2013.
33. Li, C.-T., and Hwang, M.-S., An efficient biometric-based remote authentication scheme using smart cards. *J. Netw. Comput. Appl.* 33(1):1–5, 2010.
34. 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.
35. Maitra, T., and Giri, D., An efficient biometric and password-based remote user authentication using smart card for telecare medical information systems in multi-server environment. *J. Med. Syst.* 38(12):1–19, 2014.
36. Massey, T., Marfia, G., Stoelting, A., Tomasi, R., Spirito, M.A., Sarrafzadeh, M., Pau, G., Leveraging Social System Networks in Ubiquitous High-Data-Rate Health Systems. *IEEE Trans. Inf. Technol. Biomed.* 15(3):491–498, 2011.
37. Messerges, T.S., Dabbish, E.A., Sloan, R.H., Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* 51(5):541–552, 2002.
38. Mishra, D., On the security flaws in id-based password authentication schemes for telecare medical information systems. *J. Med. Syst.* 39(1):1–16, 2015.
39. Mishra, D., Mukhopadhyay, S., Chaturvedi, A., Kumari, S., Khan, M.K., Cryptanalysis and improvement of Yan et al.'s biometric-based authentication scheme for telecare medicine information systems. *J. Med. Syst.* 38(6):1–12, 2014.
40. Mishra, D., Mukhopadhyay, S., Kumari, S., Khan, M.K., Chaturvedi, A., Security enhancement of a biometric based authentication scheme for telecare medicine information systems with nonce. *J. Med. Syst.* 38(5):1–11, 2014.
41. Mishra, D., Srinivas, J., Mukhopadhyay, S., A secure and efficient chaotic map-based authenticated key agreement scheme for telecare medicine information systems. *J. Med. Syst.* 38(10):1–10, 2014.
42. Odelu, V., Das, A.K., Goswami, A., An Effective and Secure Key-Management Scheme for Hierarchical Access Control in E-Medicine System. *J. Med. Syst.* 37(2):1–18, 2013.
43. Odelu, V., Das, A.K., Goswami, A., A secure effective key management scheme for dynamic access control in a large leaf class hierarchy. *Inf. Sci.* 269(C):270–285, 2014.
44. Odelu, V., Das, A.K., Goswami, A., A secure and efficient ECC-based user anonymity preserving single sign-on scheme for distributed computer networks. *Security and Communication Networks*, 2014. doi:10.1002/sec.1139.
45. Patel, M., and Wang, J., Applications, challenges, and prospective in emerging body area networking technologies. *IEEE Wirel. Commun.* 17(1):80–88, 2010.
46. Sarkar, P., A Simple and Generic Construction of Authenticated Encryption with Associated Data. *ACM Trans. Inf. Syst. Secur.* 13(4):1–16, 2010.
47. Siddiqui, Z., Abdullah, A.H., Khan, M.K., Alghamdi, A., Smart environment as a service: Three factor cloud based user authentication for telecare medical information system. *J. Med. Syst.* 38(1):1–14, 2013.
48. Stallings, W. *Cryptography and Network Security: Principles and Practices*. 3rd edition: Pearson Education India, 2003.
49. Stinson, D.R., Some Observations on the Theory of Cryptographic Hash Functions. *Des. Codes Crypt.* 38(2):259–277, 2006.
50. Tan, Z., An efficient biometrics-based authentication scheme for telecare medicine information systems. *Przegl. Elctrotech.* 89(5):200–204, 2013.

51. Tan, Z., A User Anonymity Preserving Three-Factor Authentication Scheme for Telecare Medicine Information Systems. *J. Med. Syst.* 38(3):1–9, 2014.
52. Tang, H., and Liu, X., Cryptanalysis of a dynamic ID-based remote user authentication with key agreement scheme. *Int. J. Commun. Syst.* 25(12):1639–1644, 2012.
53. von Oheimb, D., The high-level protocol specification language hlpsl developed in the eu project avispa. In: Proceedings of APPSEM 2005 Workshop, 2005.
54. Wei, J., Hu, X., Liu, W., An Improved Authentication Scheme for Telecare Medicine Information Systems. *J. Med. Syst.* 36(6):3597–3604, 2012.
55. Wu, Z.Y., Lee, Y.-C., Lai, F., Lee, H.-C., Chung, Y.-F., A Secure Authentication Scheme for Telecare Medicine Information Systems. *J. Med. Syst.* 36(3):1529–1535, 2012.
56. Xie, Q., A new authenticated key agreement for session initiation protocol. *Int. J. Commun. Syst.* 25(1):47–54, 2012.
57. Yan, H., Huo, H., Xu, Y., Gidlund, M., Wireless sensor network based E-health system implementation and experimental results. *IEEE Trans. Consum. Electron.* 56(4):2288–2295, 2010.
58. 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.
59. Yang, H., Kim, H., Mtonga, K., An efficient privacy-preserving authentication scheme with adaptive key evolution in remote health monitoring system. *Peer-to-Peer Networking and Applications*, 1–11, 2014. doi:10.1007/s12083-014-0299-6.
60. Zhu, Z., An Efficient Authentication Scheme for Telecare Medicine Information Systems. *J. Med. Syst.* 36(6):3833–3838, 2012.